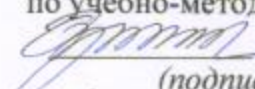


**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ НОВОСИБИРСКОЙ ОБЛАСТИ
«НОВОСИБИРСКИЙ ЭЛЕКТРОМЕХАНИЧЕСКИЙ КОЛЛЕДЖ»
(ГБПОУ НСО «Новосибирский электромеханический колледж»)**

УТВЕРЖДАЮ

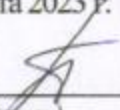
И.о. заместителя директора
по учебно-методической работе

 Е.О. Кравченко
(подпись)
«28» августа 2023 г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
ПРОФЕССИОНАЛЬНОГО МОДУЛЯ
ПМ.02 ОСУЩЕСТВЛЕНИЕ ИНТЕГРАЦИИ
ПРОГРАММНЫХ МОДУЛЕЙ**

Специальность 09.02.07. Информационные системы и программирование

Рассмотрено
на заседании кафедры общеобразовательных дисциплин
Протокол № 1 от «28» августа 2023 г.

Председатель кафедры  Белимова В.Г.
(подпись) (Ф.И.О.)

Фонд оценочных средств учебной дисциплины разработан на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности/профессии: специальность 09.02.07. Информационные системы и программирование

Организация-разработчик: ГБПОУ НСО «Новосибирский электромеханический колледж»

Согласовано:
Методист



(подпись)

Рахметова Т.Ю.

«29» августа

2023 г.

Общие положения

Результатом освоения профессионального модуля является готовность обучающегося к выполнению вида профессиональной деятельности осуществление интеграции программных модулей и составляющих его профессиональных компетенций, а также общие компетенции, формирующиеся в процессе освоения ОПОП в целом.

1. ФОРМЫ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ

Таблица 1

Элементы модуля, профессиональный модуль	Формы промежуточной аттестации
1	2
МДК. 2.1 Технология разработки программного обеспечения	Экзамен
МДК.2.2 Инструментальные средства разработки программного обеспечения	Дифференцированный зачет
МДК.2.3 Математическое моделирование	Дифференцированный зачет
УП 02.02 Практика по получению первичных профессиональных навыков	Защита практики
ПП 02 Практика по профилю специальности	Защита практика
ПМ	Экзамен по профессиональному модулю

2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ МОДУЛЯ, ПОДЛЕЖАЩИЕ ПРОВЕРКИ

2.1. Профессиональные и общие компетенции

В результате контроля и оценки по профессиональному модулю осуществляется комплексная проверка следующих профессиональных и общих компетенций:

Таблица 2

Профессиональные компетенции	Показатели оценки результата
ПК 2.1. Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент	Правильность описания спецификаций программ Обоснованность содержимого спецификации Рациональность метода описания
ПК 2.2. Выполнять интеграцию модулей в программное обеспечение	Правильность сборки модулей
ПК2.3. Выполнять отладку программного модуля с использованием специализированных программных средств	Правильность изложения принципов отладки; Обоснованность процесса отладки Рациональность методики отладки
ПК2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения	Правильность изложение принципов тестирования; Рациональность использования схем тестирования
ПК2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования	Рациональность использования кода

Таблица 3

Общие компетенции	Показатели оценки результата
ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам	– обоснованность постановки цели, выбора и применения методов и способов решения профессиональных задач; - адекватная оценка и самооценка эффективности и качества выполнения профессиональных задач
ОК 2. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности.	- использование различных источников, включая электронные ресурсы, медиаресурсы, Интернет-ресурсы, периодические издания по специальности для решения профессиональных задач
ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях.	- демонстрация ответственности за принятые решения - обоснованность самоанализа и коррекция результатов собственной работы;
ОК 4. Эффективно взаимодействовать и работать в коллективе и команде.	- взаимодействовать с обучающимися, преподавателями и мастерами в ходе обучения, с руководителями учебной и производственной практик; - обоснованность анализа работы членов команды (подчиненных)
ОК5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.	Демонстрировать грамотность устной и письменной речи, - ясность формулирования и изложения мыслей
ОК6. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения.	- соблюдение норм поведения во время учебных занятий и прохождения учебной и производственной практик,
ОК7. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях.	- эффективное выполнение правил ТБ во время учебных занятий, при прохождении учебной и производственной практик; - демонстрация знаний и использование ресурсосберегающих технологий в профессиональной деятельности

ОК 8. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности	- эффективность использовать средств физической культуры для сохранения и укрепления здоровья при выполнении профессиональной деятельности.
ОК 9. Пользоваться профессиональной документацией на государственном и иностранном языке	- эффективность использования в профессиональной деятельности необходимой технической документации, в том числе на английском языке

Таблица 4

Перечень основных показателей оценки результатов, элементов практического опыта, знаний и умений, подлежащих текущему контролю и промежуточной аттестации

Наименование основных показателей оценки результатов (ОПОР)	Наименование элемента практического опыта	Наименование элемента умений	Наименование элемента знаний
Правильность описания спецификаций программ Обоснованность содержимого спецификации Рациональность метода описания Правильность сборки модулей Правильность изложения принципов отладки; Обоснованность процесса отладки Рациональность методики отладки Правильность изложение принципов тестирования; рациональность использования схем тестирования Рациональность использования кода	модели процесса разработки программного обеспечения; основные принципы процесса разработки программного обеспечения; основные подходы к интегрированию программных модулей; основы верификации и аттестации программного обеспечения	использовать выбранную систему контроля версий; использовать методы для получения кода с заданной функциональностью и степенью качества	модели процесса разработки программного обеспечения; основные принципы процесса разработки программного обеспечения; основные подходы к интегрированию программных модулей; основы верификации и аттестации программного обеспечения

Таблица 5

Профессиональные и общие компетенции, которые возможно сгруппировать для проверки	Показатели оценки результата
ПК 2.2. Выполнять интеграцию модулей в программное обеспечение ПК2.3. Выполнять отладку программного модуля с использованием специализированных программных средств ОК 4. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.	Правильность сборки модулей - взаимодействовать с обучающимися, преподавателями и мастерами в ходе обучения, с руководителями учебной и производственной практик; - обоснованность анализа работы членов команды (подчиненных)

Контроль и оценка результатов освоения профессионального модуля по разделам

Код и наименование профессиональных и общих компетенций, формируемых в рамках модуля	Критерии оценки	Методы оценки
Раздел 1. Разработка программного обеспечения		
ПК 2.1 Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент	Оценка «отлично» - разработан и обоснован вариант интеграционного решения с помощью графических средств среды разработки, указано хотя бы одно альтернативное решение; бизнес-процессы учтены в полном объеме; вариант оформлен в полном соответствии с требованиями стандартов; результаты верно сохранены в системе контроля версий. Оценка «хорошо» - разработана и прокомментирована архитектура варианта интеграционного решения с помощью графических средств, учтены основные бизнес-процессы; вариант оформлен в соответствии с требованиями стандартов; результаты сохранены в системе контроля версий. Оценка «удовлетворительно» - разработана и архитектура варианта интеграционного решения с помощью графических средств, учтены основные бизнес-процессы с незначительными упущениями; вариант оформлен в соответствии с требованиями стандартов с некоторыми отклонениями; результат сохранен в системе контроля версий.	Экзамен в форме собеседования: - практическое задание по формированию требований к программным модулям в соответствии с техническим заданием. Защита отчетов по практическим и лабораторным работам. Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики
ПК 2.4 Осуществлять разработку тестовых наборов и тестовых сценариев для программного	Оценка «отлично» - обоснован размер тестового покрытия, разработан тестовый сценарий и тестовые пакеты в соответствии с этим сценарием в соответствии с минимальным размером	Экзамен в форме собеседования: практическое задание по разработке тестовых

обеспечения	тестового покрытия, выполнено тестирование интеграции и ручное тестирование, выполнено тестирование с применением инструментальных средств, выявлены ошибки системных компонент (при наличии), заполнены протоколы тестирования. Оценка «хорошо»- обоснован размер тестового покрытия, разработан тестовый сценарий и тестовые пакеты в соответствии с этим сценарием, выполнено тестирование интеграции и ручное тестирование, выполнено тестирование с применением инструментальных средств, заполнены протоколы тестирования. Оценка «удовлетворительно»- определен размер тестового покрытия, разработан тестовый сценарий и тестовые пакеты, выполнено тестирование интеграции и ручное тестирование, частично выполнено тестирование с применением инструментальных средств, частично заполнены протоколы тестирования.	сценариев и наборов для заданных видов тестирования и выполнение тестирования. Защита отчетов по практическим и лабораторным работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики
ПК 2.5 Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования	Оценка «отлично» - продемонстрировано знание стандартов кодирования более чем одного языка программирования, выявлены все имеющиеся несоответствия стандартам в предложенном коде. Оценка «хорошо» - продемонстрировано знание стандартов кодирования более чем одного языка программирования, выявлены существенные имеющиеся несоответствия стандартам в предложенном коде. Оценка «удовлетворительно» - продемонстрировано знание стандартов кодирования языка программирования, выявлены некоторые несоответствия стандартам в предложенном коде.	Экзамен в форме собеседования: практическое задание по инспектированию программного кода Защита отчетов по практическим и лабораторным работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики
Раздел модуля 2 Средства разработки программного обеспечения		
ПК 2.2 Выполнять интеграцию модулей в программное обеспечение	Оценка «отлично» - в системе контроля версий выбрана верная версия проекта, проанализирована его архитектура, архитектура доработана для интеграции нового модуля; выбраны способы форматирования данных и организована их постобработка, транспортные протоколы и форматы сообщений обновлены (при необходимости);	Экзамен в форме собеседования: практическое задание по обеспечению интеграции заданного модуля в предложенный программный проект

	протестирована интеграция модулей проекта и выполнена отладка проекта с применением инструментальных средств среды; выполнена доработка модуля и дополнительная обработка исключительных ситуаций в том числе с созданием классов-исключений (при необходимости); определены качественные показатели полученного проекта; результат интеграции сохранен в системе контроля версий. Оценка «хорошо» - в системе контроля версий выбрана верная версия проекта, его архитектура доработана для интеграции нового модуля; выбраны способы форматирования данных и организована их постобработка, транспортные протоколы и форматы сообщений обновлены (при необходимости); выполнена отладка проекта с применением инструментальных средств среды; выполнена доработка модуля и дополнительная обработка исключительных ситуаций (при необходимости); определены качественные показатели полученного проекта; результат интеграции сохранен в системе контроля версий. Оценка «удовлетворительно» - в системе контроля версий выбрана верная версия проекта, его архитектура доработана для интеграции нового модуля; выбраны способы форматирования данных и организована их постобработка, форматы сообщений обновлены (при необходимости); выполнена отладка проекта с применением инструментальных средств среды; выполнена доработка модуля (при необходимости); результат интеграции сохранен в системе контроля версий.	Защита отчетов по практическим и лабораторным работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики
ПК 2.3 Выполнять отладку программного модуля с использованием специализированных программных средств	Оценка «отлично» - в системе контроля версий выбрана верная версия проекта; протестирована интеграция модулей проекта и выполнена отладка проекта с применением инструментальных средств среды; проанализирована и сохранена отладочная информация; выполнена условная компиляция проекта в среде разработки; определены качественные показатели полученного проекта в	Экзамен в форме собеседования: практическое задание по выполнению отладки программного модуля. Защита отчетов по практическим и

	полном объеме; результаты отладки сохранены в системе контроля версий. Оценка «хорошо» - в системе контроля версий выбрана верная версия проекта; протестирована интеграция модулей проекта и выполнена отладка проекта с применением инструментальных средств среды; выполнена условная компиляция проекта в среде разработки; определены качественные показатели полученного проекта в достаточном объеме; результаты отладки сохранены в системе контроля версий. Оценка «удовлетворительно» - в системе контроля версий выбрана верная версия проекта; выполнена отладка проекта с применением инструментальных средств среды; выполнена условная компиляция проекта в среде разработки; определены качественные показатели полученного проекта в достаточном объеме; результаты отладки сохранены в системе контроля версий.	лабораторным работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики
ПК 2.5 Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования	Оценка «отлично» - продемонстрировано знание стандартов кодирования более чем одного языка программирования, выявлены все имеющиеся несоответствия стандартам в предложенном коде. Оценка «хорошо» - продемонстрировано знание стандартов кодирования более чем одного языка программирования, выявлены существенные имеющиеся несоответствия стандартам в предложенном коде. Оценка «удовлетворительно» - продемонстрировано знание стандартов кодирования языка программирования, выявлены некоторые несоответствия стандартам в предложенном коде.	Экзамен в форме собеседования: практическое задание по инспектированию программного кода Защита отчетов по практическим и лабораторным работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики
Раздел модуля 3 Моделирование в программных системах		
ПК 2.4 Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения	Оценка «отлично» - обоснован размер тестового покрытия, разработан тестовый сценарий и тестовые пакеты в соответствии с этим сценарием в соответствии с минимальным размером тестового покрытия, выполнено тестирование интеграции и ручное тестирование, выполнено тестирование с применением инструментальных средств, выявлены ошибки системных компонент	Экзамен/зачет в форме собеседования: практическое задание по разработке тестовых сценариев и наборов для заданных видов тестирования и выполнение тестирования.

	(при наличии), заполнены протоколы тестирования. Оценка «хорошо»- обоснован размер тестового покрытия, разработан тестовый сценарий и тестовые пакеты в соответствии с этим сценарием, выполнено тестирование интеграции и ручное тестирование, выполнено тестирование с применением инструментальных средств, заполнены протоколы тестирования. Оценка «удовлетворительно»- определен размер тестового покрытия, разработан тестовый сценарий и тестовые пакеты, выполнено тестирование интеграции и ручное тестирование, частично выполнено тестирование с применением инструментальных средств, частично заполнены протоколы тестирования.	Защита отчетов по практическим и лабораторным работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики
ПК 2.5 Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.	Оценка «отлично» - продемонстрировано знание стандартов кодирования более чем одного языка программирования, выявлены все имеющиеся несоответствия стандартам в предложенном коде. Оценка «хорошо» - продемонстрировано знание стандартов кодирования более чем одного языка программирования, выявлены существенные имеющиеся несоответствия стандартам в предложенном коде. Оценка «удовлетворительно» - продемонстрировано знание стандартов кодирования языка программирования, выявлены некоторые несоответствия стандартам в предложенном коде.	Экзамен/зачет в форме собеседования: практическое задание по инспектированию программного кода Защита отчетов по практическим и лабораторным работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики
ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.	– обоснованность постановки цели, выбора и применения методов и способов решения профессиональных задач; - адекватная оценка и самооценка эффективности и качества выполнения профессиональных задач	Экспертное наблюдение за выполнением работ
ОП 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.	- использование различных источников, включая электронные ресурсы, медиаресурсы, Интернет-ресурсы, периодические издания по специальности для решения профессиональных задач	

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие.	- демонстрация ответственности за принятые решения - обоснованность самоанализа и коррекция результатов собственной работы;	
ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.	- взаимодействовать с обучающимися, преподавателями и мастерами в ходе обучения, с руководителями учебной и производственной практик; - обоснованность анализа работы членов команды (подчиненных)	
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.	Демонстрировать грамотность устной и письменной речи, - ясность формулирования и изложения мыслей	
ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей.	- соблюдение норм поведения во время учебных занятий и прохождения учебной и производственной практик,	
ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях.	- эффективное выполнение правил ТБ во время учебных занятий, при прохождении учебной и производственной практик; - демонстрация знаний и использование ресурсосберегающих технологий в профессиональной деятельности	
ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности.	- эффективность использовать средств физической культуры для сохранения и укрепления здоровья при выполнении профессиональной деятельности.	

ОК 09. Использовать информационные технологии в профессиональной деятельности.	- эффективность использования информационно-коммуникационных технологий в профессиональной деятельности согласно формируемым умениям и получаемому практическому опыту;	
ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языках.	- эффективность использования в профессиональной деятельности необходимой технической документации, в том числе на английском языке.	

2.4. Требования к курсовому проекту

Общие требования, которым должен удовлетворять курсовой проект:

- высокий теоретический уровень проекта;
- глубина и комплексность исследования, полнота освещения излагаемых вопросов;
- четкость построения, логическая последовательность изложения материала;
- убедительность аргументации, полнота, краткость и точность формулировок;
- тщательность, грамотность оформления текстовой и графической части работы;
- конкретность изложения, доказательность выводов.

Выбор темы курсовой работы осуществляется свободно, желательно исходя из нужд предприятия, так же имеется список актуальных тем, который разрабатывается преподавателями, рассматривается предметной цикловой комиссией и доводится до сведения студентов.

Выбранные студентами темы курсовых работ регистрируются в соответствующей документации и изменение тем возможно только через процедуру решения ПЦК «Информационных дисциплин и математики»

Научным руководителем курсовой работы студента назначается ведущий преподаватель ПМ03 МДК03.2 «Осуществление интеграции программных модулей».

В курсовом проекте студент может показать умение изучать и обобщать литературные источники; способность доказательно выявить основную проблему и пути ее разрешения; владение современной вычислительной техникой; умение делать выводы и оценивать эффективность предлагаемых решений.

Пояснительная записка к курсовому проекту содержит текстовой, табличный, графический материал, расположенный по тексту и (или) вынесенный в приложения.

Пояснительная записка должна иметь следующий состав:

- титульный лист;
- содержание;
- введение;
- основную часть;
- заключение;
- перечень условных обозначений, символов, терминов (при необходимости);
- список использованных источников;
- приложения.

Титульный лист выполняется на специальных формах, в соответствии с требованиями описанных в методических рекомендациях к написанию курсовых работ.

Содержание должно включать все заголовки, имеющиеся в пояснительной записке, в той же последовательности.

Введение, основная часть и заключение составляют содержание курсового проекта.

В список использованных источников включают все источники, располагая их в порядке появления ссылок в тексте или по алфавиту.

В приложениях к пояснительной записке должны помещаться материалы вспомогательного характера, которые при включении их в основную часть загромождают ее. К таким материалам могут быть отнесены таблицы справочного и вспомогательного характера, таблицы исследований и статистических (отчетных) данных, копии и формы документов, иллюстрации вспомогательного характера, спецификации к листам графической части, алгоритмы, программы, распечатки расчетов на ЭВМ и т.д.

Общий объем пояснительной записки рекомендуется, как правило, в пределах 40 – 70 страниц.

Курсовой проект должен быть оформлен в гибкую пластиковую папку для бумаг (скоросшиватель).

Темы курсовых работ:

1. Разработка программного обеспечения «Автосалон»
2. Разработка программного обеспечения «Автошкола»
3. Разработка программного обеспечения «Агентство по трудоустройству»
4. Разработка программного обеспечения «АЗС»
5. Разработка программного обеспечения «Анализ успеваемости учащихся школы»
6. Разработка программного обеспечения «Аптека»
7. Разработка программного обеспечения «Библиотека»
8. Разработка программного обеспечения «Военный комиссариат»
9. Разработка программного обеспечения «Дежурство в колледже»
10. Разработка программного обеспечения «Кадровый отдел»
11. Разработка программного обеспечения «Касса автовокзала»
12. Разработка программного обеспечения «Комплектование и учет музейных фондов»
13. Разработка программного обеспечения «Компьютерный салон»
14. Разработка программного обеспечения «Ломбард»
15. Разработка программного обеспечения «Музей»
16. Разработка программного обеспечения «Оплата жилищно-коммунальных услуг»
17. Разработка программного обеспечения «Рекламное агентство»
18. Разработка программного обеспечения «Ремонт бытовой техники»
19. Разработка программного обеспечения «Салон красоты»
20. Разработка программного обеспечения «Спортивный комплекс»
21. Разработка программного обеспечения «Учебно-методическое пособие по физике»
22. Разработка программного обеспечения «Учет расчетов с поставщиками»
23. Разработка программного обеспечения «Учет строительных материалов»
24. Разработка программного обеспечения «Шиномонтажная мастерская»
25. Разработка программного обеспечения «Электронная касса»
26. Разработка программного обеспечения «Электронная регистратура»
27. Разработка программного обеспечения «Электронное портфолио группы»
28. Разработка программного обеспечения «Интернет провайдер»
29. Разработка программного обеспечения «Учет компьютеров предприятия»

Показатели оценки работы (проекта)

Таблица 9

Коды и наименования проверяемых компетенций или их сочетаний	Показатели оценки результата	Оценка (да / нет)
ПК 2.1. Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент	Правильность описания спецификаций программ Обоснованность содержимого спецификации Рациональность метода описания	да
ПК 2.2. Выполнять интеграцию модулей в программное обеспечение	Правильность сборки модулей	да
ПК 2.3. Выполнять отладку программного модуля с использованием специализированных программных средств	Правильность изложения принципов отладки; Обоснованность процесса отладки Рациональность методики отладки	нет
ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения	Правильность изложение принципов тестирования; Рациональность использования схем тестирования	нет
ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования	Рациональность использования кода	да

3. ОЦЕНКА ОСВОЕНИЯ ТЕОРЕТИЧЕСКОГО КУРСА ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

3.1. Типовые задания для оценки освоения

МДК. 2.1 Технология разработки программного обеспечения:

Раздел 1. Разработка программного обеспечения

Тема 2.1.1 Основные понятия и стандартизация требований к программному обеспечению **Задание 1**

Проверяемые результаты обучения: ПК2.1 ПК2.5

Вид учебной работы: аудиторная

Задание. Ответить на устные вопросы:

1. Понятия требований, классификация, уровни требований. Методологии и стандарты, регламентирующие работу с требованиями.
2. Современные принципы и методы разработки программных приложений
3. Методы организации работы в команде разработчиков. Системы контроля версий
4. Основные подходы к интегрированию программных модулей
5. Стандарты кодирования

Критерии оценки: Полнота ответа на поставленные вопросы

«5»- Правильный уверенный ответ

«4» - Уверенный ответ, допущена одна неточность

«3» - Ответ близкий к правильному, с несколькими неточностями

«2» - Неправильный ответ, отказывается отвечать

Задание 2

Практическая работа Анализ предметной области

Проверяемые результаты обучения: ПК2.1 ПК2.2 ПК2.3 ПК2.4 ПК2.5

Вид учебной работы: аудиторная работа (практическая работа Анализ предметной области)

Задание:

1. Представить описание программного продукта: наименование, цели и задачи системы, выполняемые функции, составные модули.
2. Описать работу модулей, их взаимосвязь. Также указать характеристику внешних сущностей программного продукта.
3. Представить список информационных объектов программной системы.
4. Охарактеризовать категории пользователей системы.

ВИДЫ ПРЕДМЕТНЫХ ОБЛАСТЕЙ

1. Страховая компания
2. Гостиница
3. Ломбард
4. Реализация готовой продукции
5. Ведение заказов
6. Бюро по трудоустройству
7. Нотариальная контора
8. Фирма по продаже запчастей
9. Курсы повышения квалификации
10. Туристическая фирма
11. Грузовые перевозки
12. Ювелирная мастерская
13. Парикмахерская
14. Химчистка
15. Сдача в аренду торговых площадей

Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

Задание 3

Практическая работа Разработка и оформление технического задания

Проверяемые результаты обучения: ПК2.1 ПК2.2 ПК2.3 ПК2.4 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

В соответствии с вариантом задания (работа №1) выполнить разработку технического задания, которое должно включать:

1. Введение
 - 1.1. Наименование программы
 - 1.2. Назначение и область применения
2. Требования к программе
 - 2.1. Требования к функциональным характеристикам
 - 2.2. Требования к надежности
 - 2.2.1. Требования к обеспечению надежного функционирования программы

- 2.2.2. Время восстановления после отказа
- 2.2.3. Отказы из-за некорректных действий пользователей системы
- 3. Условия эксплуатации
 - 3.1. Климатические условия эксплуатации
 - 3.2. Требования к квалификации и численности персонала
 - 3.3. Требования к составу и параметрам технических средств
 - 3.4. Требования к информационной и программной совместимости
 - 3.4.1. Требования к информационным структурам и методам решения
 - 3.4.2. Требования к исходным кодам и языкам программирования
 - 3.4.3. Требования к программным средствам, используемым программой
 - 3.4.4. Требования к защите информации и программ
 - 3.5. Специальные требования
- 4. Требования к программной документации
 - 4.1. Предварительный состав программной документации
- 5. Технико-экономические показатели
 - 5.1. Экономические преимущества разработки
- 6. Стадии и этапы разработки
 - 6.1. Стадии разработки
 - 6.2. Этапы разработки
 - 6.3. Содержание работ по этапам
- 7. Порядок контроля и приемки
 - 7.1. Виды испытаний
 - 7.2. Общие требования к приемке работы

Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

Задание 4

Практическая работа Построение архитектуры программного средства

Проверяемые результаты обучения: ПК2.1 ПК2.2 ПК2.3 ПК2.4 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

Разработать проект архитектуры программного средства в соответствии с ГОСТ Р ИСО/МЭК 12207.

С целью реализации начальных этапов разработки ПС в соответствии с техническим заданием:

- 1.выполнить подготовительную работу;
- 2.провести анализ требований к ПС;
- 3.выполнить проектирование архитектуры ПС на высоком уровне: построить архитектуру программы в виде модульной схемы.

Определить функционал и состав модулей.

При возникновении потребностей в заказе, приобретении, разработке, эксплуатации и сопровождении программ перед всеми сторонами, вовлеченными в жизненный цикл программного средства (ПС), возникает целый ряд вопросов, связанных с определением и детальным структурированием жизненного цикла (ЖЦ) ПС, с организационными и техническими правами и обязанностями сторон, с управлением ЖЦ и контролем за его реализацией. Одним из действенных инструментов для решения данных вопросов является

использование унифицированных подходов, закрепленных в современных международных и российских стандартах.

Понятия «жизненный цикл системы» или «жизненный цикл программного средства» часто появляются в статьях и звучат в разговорах разработчиков, по крайней мере, руководителей проектов и подразделений.

Всем понятно, что относятся они к тому, что и в какой последовательности должно делаться при создании и эксплуатации систем. Но прежде, чем две организации или два специалиста договорятся о том, что конкретно входит или не входит в ЖЦ, проходит значительное время. А позже вполне может обнаружиться, что эти двое (две «стороны») все-таки по-разному понимают, какие работы будут входить в ЖЦ, а какие - нет, какие проверки будут планироваться, когда и т. д. Естественно, общие принципы организации работ описаны давно, но что делать сторонам в конкретном проекте — это каждый раз приходится решать заново.

В стандартах, регламентирующих жизненный цикл программных средств, обобщаются опыт и результаты исследований множества специалистов и рекомендуются наиболее эффективные современные методы и процессы создания и развития комплексов программ. В результате таких обобщений оттачиваются технологические процессы и приемы разработки, а также методическая база для их автоматизации.

ЖЦ ПС в стандартах представляет собой набор этапов, частных работ и операций в последовательности их выполнения и взаимосвязи, регламентирующих ведение работ от подготовки технического задания до завершения испытаний ряда версий и окончания эксплуатации ПС или информационной системы (ИС).

Стандарты включают правила описания исходной информации, способов и методов выполнения операций, устанавливают правила контроля технологических процессов, требования к оформлению их результатов, а также регламентируют содержание технологических и эксплуатационных документов на комплексы программ. Они определяют организационную структуру коллектива, обеспечивают распределение и планирование заданий, а также, контроль за ходом создания ПС.

Для того чтобы привести порядок и понимание, общие для любых сторон, участвующих в ЖЦ систем и ПС, давно разрабатывались стандарты различных уровней утверждения - национальные и международные.

В России основы построения и использования профилей стандартов ЖЦ ПС заложены принятием в качестве базового стандарта ГОСТ Р ИСО/МЭК 12207. Данный документ введен в действие с 1 июля 2000 г., тесно взаимоувязан с рядом стандартов, принятых ранее, и с некоторыми стандартами, разрабатываемыми в данное время на основе прямого применения стандартов ИСО.

Актуальность стандарта ГОСТ Р ИСО/МЭК 12207 для современных условий настолько высока, что принятие в ISO его исходного, международного варианта вскоре вызвало самую положительную оценку российских экспертов. Был дан ряд рекомендаций по его использованию в реальных условиях.

В данном стандарте программное обеспечение (ПО) или программный продукт определяется как набор компьютерных программ, процедур и связанной с ними документации и данных.

Процесс определяется как совокупность взаимосвязанных действий, преобразующих некоторые входные данные в выходные. Каждый процесс характеризуется определенными задачами и методами их решения, исходными данными, полученными от других процессов, и результатами.

В соответствии с ГОСТ Р ИСО/МЭК 12207 все процессы ЖЦ ПО разделены на три группы:

1) Основные процессы:

- приобретение;
- поставка;
- разработка;
- эксплуатация;
- сопровождение.

2) Вспомогательные процессы:

- документирование;
- управление конфигурацией;
- обеспечение качества;
- верификация;
- аттестация;
- совместная оценка;
- аудит;
- разрешение проблем.

3) Организационные процессы:

- управление;
- усовершенствование;
- создание инфраструктуры;
- обучение.

Процесс разработки предусматривает действия и задачи, выполняемые разработчиком, и включает следующие действия:

А) Подготовительная работа, которая начинается с выбора модели ЖЦ ПО, соответствующей масштабу, значимости и сложности проекта. Действия и задачи процесса должны соответствовать выбранной модели. Разработчик должен выбрать, адаптировать к условиям проекта и использовать согласованные с заказчиком стандарты, методы и средства разработки, а также составить план выполнения работ.

Б) Анализ требований к системе подразумевает определение ее функциональных возможностей, пользовательских требований, требований к надежности и безопасности, требований к внешним интерфейсам и т.д. Требования к системе оцениваются исходя из критериев реализуемости и возможности проверки при тестировании.

Анализ требований к ПО предполагает определение следующих характеристик для каждого компонента ПО:

- функциональных возможностей, включая характеристики производительности и среды функционирования компонента;
- внешних интерфейсов;
- спецификаций надежности и безопасности;
- эргономических требований;
- требований к используемым данным;
- требований к установке и приемке;
- требований к пользовательской документации;
- требований к эксплуатации и сопровождению.

Требования к ПО оцениваются исходя из критериев соответствия требованиям к системе, реализуемости и возможности проверки при тестировании.

В) Проектирование архитектуры системы на высоком уровне заключается в определении компонентов ее оборудования, ПО и операций, выполняемых эксплуатирующим систему персоналом. Архитектура системы должна соответствовать требованиям, предъявляемым к системе, а также принятым проектным стандартам и методам.

Проектирование архитектуры ПО включает следующие задачи:

- трансформацию требований к ПО в архитектуру, определяющую на высоком уровне структуру ПО и состав ее компонентов;
- разработку и документирование программных интерфейсов ПО и баз данных;
- разработку предварительной версии пользовательской документации;
- разработку и документирование предварительных требований к тестам и планам интеграции ПО.

Архитектура компонентов ПО должна соответствовать требованиям, предъявляемым к ним, а также принятым проектным стандартам и методам.

Г) Детальное проектирование ПО включает следующие задачи:

- описание компонентов и интерфейсов между ними на более низком уровне, достаточном для их последующего самостоятельного кодирования и тестирования;
- разработку и документирование детального проекта базы данных;
- обновление (при необходимости) пользовательской документации;

Д) Кодирование и тестирование ПО охватывает задачи:

- разработку и документирование каждого компонента ПО и базы данных, а также совокупности тестовых процедур и данных для их тестирования;
- тестирование каждого компонента ПО и базы данных на соответствие предъявляемых к ним требованиям. Результаты тестирования компонентов должны быть документированы;
- обновление (при необходимости) пользовательской документации;
- обновление плана интеграции ПО.

Е) Интеграция ПО предусматривает сборку разработанных компонентов ПО в соответствии с планом интеграции и тестирование агрегированных компонентов. Для каждого из агрегированных компонентов разрабатываются наборы тестов и тестовые процедуры, предназначенные для проверки каждого из квалификационных требований при последующем квалификационном тестировании.

Ж) Квалификационное тестирование - это набор критериев и условий, которые необходимо выполнить, чтобы квалифицировать программный продукт как соответствующий своим спецификациям и готовый к использованию в условиях эксплуатации.

Квалификационное тестирование ПО проводится разработчиком, в присутствии заказчика (по возможности), для демонстрации того, что ПО удовлетворяет своим спецификациям и готово к использованию в условиях эксплуатации. Квалификационное тестирование выполняется для каждого компонента ПО по всем разделам требований при широком варьировании тестов.

З) Установка ПО осуществляется разработчиком в соответствии с планом в той среде и на том оборудовании, которые предусмотрены договором. В процессе установки проверяется работоспособность ПО и баз данных.

И) Приемка ПО предусматривает оценку результатов квалификационного тестирования ПО и системы и документирование результатов оценки, которые проводятся заказчиком с помощью разработчика.

Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

Задание 5

Практическая работа Изучение работы в системе контроля версий

Проверяемые результаты обучения: ПК2.1 ПК2.2 ПК2.3 ПК2.4 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

1. Изучить материал
2. Выполнить действия по установке и настройке системы

Управление версиями-это искусство работы с изменениями информации. Долгое время оно было жизненно важным инструментом программистов, которым необходимо произвести небольшие изменения в программе, или же сделать — откат|| изменений, возвращаясь к предыдущей версии. Однако полезность систем управления версиями выходит далеко за пределы мира разработчиков программного обеспечения. Управление версиями требуется повсюду, где можно встретить людей, использующих компьютер для работы с постоянно изменяющейся информациейSoftware Configuration Managementили

Конфигурационное управление подразумевает под собой комплекс методов, направленных на то, чтобы систематизировать изменения, вносимые разработчиками в программный продукт в процессе его разработки и сопровождения, сохранить целостность системы после изменений, предотвратить нежелательные и непредсказуемые эффекты, а также сделать процесс внесения изменений более формальным. Изначально управление конфигурацией применялось не в программировании, но в связи с высокой динамичностью сферы разработки ПО, в ней она особенно полезна. К процедурам можно отнести создание резервных копий, контроль исходного кода, требований проекта, документации т. д.

Степень формальности выполнения данных процедур зависит от размеров проекта, и при правильной ее оценке данная концепция может быть очень полезна.

Конфигурационное управление требует выполнения множества трудоемких рутинных операций. На практике, в большинстве случаев, для конфигурационного управления применяются специальные системы контроля версий исходного кода программ. В качестве примера такой системы рассмотрим самую распространенную на сегодняшний день — Subversion.Subversion —это бесплатная система управления версиями с открытым исходным кодом.

Subversion позволяет управлять файлами и каталогами, а так же сделанными в них изменениями во времени. Это позволяет восстановить более ранние версии данных, даёт возможность изучить

историю всех изменений. Благодаря этому многие считают систему управления версиями своего

рода «машиной времени»

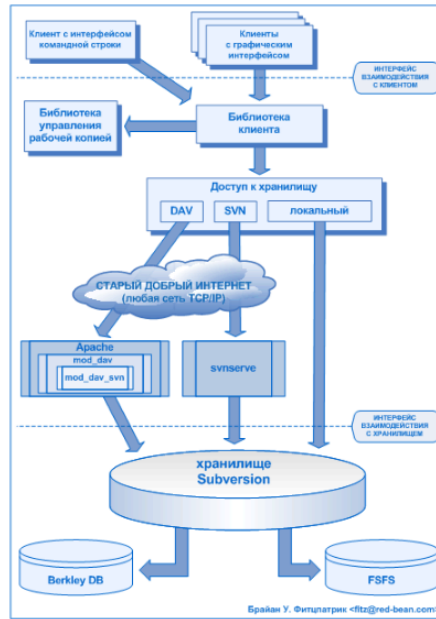


Схема общей архитектуры Subversion. В нижней части схемы изображено хранилище Subversion, в котором хранится информация с версиями. В верхней части схемы показана программа-клиент Subversion, которая управляет локальными отражениями различных фрагментов этих данных (также называемыми «рабочими копиями»). Между этими сторонами проложены различные маршруты, проходящие через разные слои доступа к хранилищу. Некоторые из этих маршрутов используют компьютерные сети и сетевые сервера, чтобы достичь хранилища, в то время как другие маршруты в сети не нуждаются и ведут к хранилищу напрямую.

Subversion может работать через сеть, что позволяет использовать её на разных компьютерах. В какой-то степени, возможность большого количества людей не зависимо от их местоположения совместно работать над единым комплектом данных поощряет сотрудничество. Когда нет того ответственного звена цепи, того контролирующего элемента, который утверждает все изменения, работа становится более эффективной. При этом не нужно опасаться, что отказ от контролирующего элемента повлияет на качество, ведь благодаря сохранению истории изменений, даже если при изменении данных будут допущены ошибки, всегда можно сделать откат изменений к прежнему состоянию.

Некоторые системы управления версиями выступают также в качестве систем управления конфигурацией программного обеспечения. Такие системы специально созданы для управления деревьями исходного кода и имеют множество особенностей, непосредственно относящихся к разработке программ: они понимают языки программирования и предоставляют инструменты для сборки программ. Subversion не является такой системой, она представляет собой систему общего назначения, которую можно использовать для управления любым набором файлов. Для Вас это будут исходники Ваших программ, а для кого-то другого это будет список продуктов или сведённое цифровое видео.

TortoiseSVN

TortoiseSVN—это бесплатный Windows-клиент с открытым исходным кодом для системы управления версиями Apache™ Subversion®. То есть TortoiseSVN управляет файлами и директориями во времени. Файлы хранятся в центральном хранилище.

Хранилище больше похоже на обычный файловый сервер, кроме того, он запоминает каждое изменение, когда-либо сделанное в ваших файлах и директориях. Это позволяет вам восстановить старые версии ваших файлов и проверить историю изменений —как, когда и кто изменял ваши данные. Вот почему многие думают о Subversion, и вообще о системах управления версиями, как о своего рода «машине времени».

Некоторые системы контроля версий являются также и системами управления конфигурацией программ (software configuration management -SCM). Такие системы специально созданы для управления деревьями исходного кода, и имеют множество возможностей, специфичных для разработки программ, таких как непосредственное понимание языков программирования, или предоставление инструментов для сборки программ. Однако Subversion не является такой системой, она является системой общего назначения, которая может быть использована для управления любым набором файлов, включая и исходные коды программ.

1.2. Возможности TortoiseSVN

Что делает TortoiseSVN таким хорошим клиентом Subversion? Вот краткий список возможностей: Интеграция с оболочкой TortoiseSVN интегрируется непосредственно в оболочку Windows (т.е. в Проводник). Это значит, что вы можете работать с уже знакомыми инструментами, и вам не надо переключаться на другое приложение каждый раз, когда вам необходимы функции для управления версиями!

И вам даже не обязательно использовать именно Проводник. Контекстные меню TortoiseSVN работают во многих других файловых менеджерах, и в диалогах для открытия файлов, используемых в большинстве стандартных Windows-приложений. Однако вы должны учитывать, что TortoiseSVN изначально разработан как расширение для Проводника Windows, и

возможно, в других приложениях интеграция будет не полной, например, могут не отображаться пометки на значках.

Пометки на значках.

Статус каждого версированного файла и папки отображается при помощи маленькой пометки поверх основного значка. Таким образом, вы сразу можете видеть состояние вашей рабочей копии.

Графический интерфейс пользователя

При просмотре списка изменений файла или папки вы можете кликнуть на ревизию, чтобы увидеть комментарии для этой фиксации. Также доступен список измененных файлов - всего лишь сделайте двойной клик на файле, чтобы увидеть, какие конкретно изменений были внесены.

Диалог фиксации - это список, в котором перечислены все файлы и папки, которые будут включены в фиксацию. У каждого элемента списка имеется флажок, чтобы вы могли выбрать именно то, что вы хотите включить в фиксацию. Неверсированные файлы также могут быть представлены в этом списке, чтобы вы не забыли добавить в фиксацию новый файл или папку

Простой доступ к командам Subversion

Все команды Subversion доступны из контекстного меню Проводника. TortoiseSVN добавляет туда собственное подменю. Поскольку TortoiseSVN является клиентом Subversion, мы хотели бы показать и некоторые из возможностей самой Subversion:

Версирование папок

CVS отслеживает только историю отдельных файлов, тогда как Subversion реализует «виртуальную» версию файловую систему, которая отслеживает изменения в целых деревьях папок во времени. Файлы и папки являются версированными. В результате, есть команды переместить и копировать, реально выполняемые на стороне клиента и работающие непосредственно с файлами и папками.

Атомарные фиксации

Фиксация сохраняется в хранилище либо полностью, либо не сохраняется вообще. Это позволяет разработчикам фиксировать изменения, собранные в логически связанные части.

Версированные метаданные

Каждый файл и папка имеет прикрепленный невидимый набор «свойств». Вы можете создавать и сохранять произвольные пары ключ/значение для собственных нужд. Свойства тоже версируются во времени, как и содержимое файла.

Возможность выбора сетевого уровня

В Subversion есть абстрагируемое понятие доступа к хранилищу, которое упрощает реализацию новых сетевых механизмов.

«Усовершенствованный» сетевой сервер Subversion является модулем для веб-сервера Apache, который использует для взаимодействия диалект HTTP под названием WebDAV/DeltaV. Это даёт Subversion большие преимущества в стабильности и совместимости, и предоставляет различные ключевые возможности без дополнительных затрат: проверка личности (аутентификация), проверка прав доступа (авторизация), сжатие потока данных при передаче, просмотр хранилища. Также доступна меньшая, автономная версия сервера Subversion, взаимодействующая по собственному протоколу, который легко может быть туннелирован через ssh

Единый способ обработки данных

Subversion получает различия между файлами при помощи бинарного разностного алгоритма, который работает одинаково как с текстовыми (читаемыми человеком), так и с

бинарными (не читаемыми человеком) файлами. Оба типа файлов содержатся в хранилище в сжатом виде, а различия передаются по сети в обоих направлениях.

Эффективные ветки и метки

Стоимость создания веток и меток не обязательно должна быть пропорциональна размеру проекта. Subversion создаёт ветки и метки, просто копируя проект с использованием механизма, похожего на жёсткие ссылки в файловых системах. Благодаря этому, операции по созданию веток и меток происходят за одинаковое, очень малое время и занимают очень мало места в хранилище.

1.3. Установка

TortoiseSVN поставляется в виде простого в использовании установочного файла (на странице загрузки <http://tortoisesvn.net/downloads.html> необходимо выбрать дистрибутив, соответствующий разрядности системы). Сделайте на нем двойной клик и следуйте инструкциям -остальное он сделает за вас. Также на странице загрузки необходимо скачать языковой пакет и установить после установки основной программы. Не забудьте перезагрузить компьютер после установки

1.4. Основная концепция

Перед тем, как мы погрузимся в работу с настоящими файлами, важно получить представление о том, как работает Subversion и какие термины используются.

Хранилище

Subversion использует центральную базу данных, которая содержит все ваши версированные файлы с их полной историей. Эта база данных называется хранилищем. Хранилище обычно находится на файловом сервере, на котором установлен Subversion, по запросу поставляющий данные клиентам Subversion (например, TortoiseSVN). Если вы делаете резервное копирование, то копируйте ваше хранилище, так как это оригинал всех ваших данных.

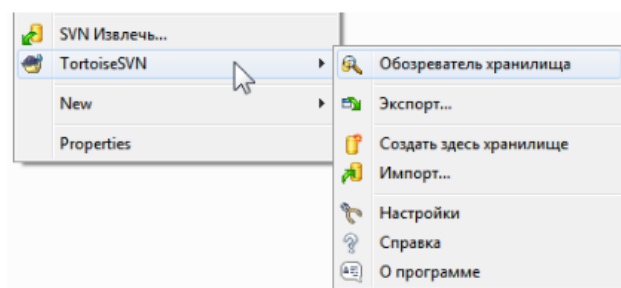
Рабочая копия

Это именно то место, где вы работаете. Каждый разработчик имеет собственную рабочую копию, иногда называемую песочницей, на своем локальном компьютере. Вы можете получить из хранилища последнюю версию файлов, поработать над ней локально, никак не взаимодействуя с кем-либо еще, а когда вы будете уверены в изменениях вы можете зафиксировать эти файлы обратно в хранилище.

Рабочая копия не содержит историю проекта, но содержит копию всех файлов, которые были в хранилище до того, как вы начали делать изменения. Это обозначает, что можно легко узнать, какие конкретно изменения вы сделали.

Вам также нужно знать, где найти TortoiseSVN, потому что в меню «Пуск» его нет. Это так потому, что TortoiseSVN -расширение проводника Windows, поэтому для начала нужно запустить проводник. Сделайте правый клик на папке в проводнике и вы увидите новые пункты в контекстном меню, такие как эти:

Рисунок 1.1. Меню TortoiseSVN для неверсированных папок



1.5. Создание хранилища

Для настоящего проекта вам понадобится хранилище, созданное в безопасном месте, и сервер Subversion, чтобы управлять им. Для обучения мы будем использовать функцию локального хранилища Subversion, которая разрешает прямой доступ к хранилищу, созданного на вашем жестком диске, и не требует наличия сервера.

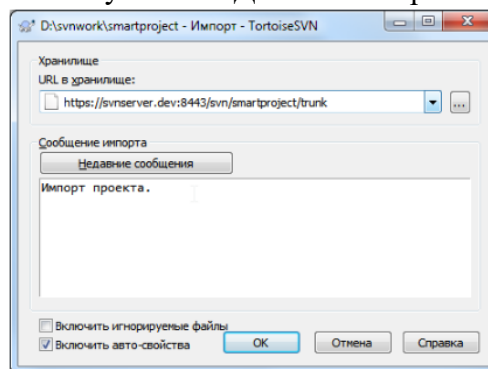
Сначала создайте новую пустую директорию на вашем ПК. Она может быть где угодно, но в этом руководстве мы собираемся назвать её C:\svn_repos. Теперь сделайте правый клик на новой папке и в контекстном меню выберите TortoiseSVN→Создать здесь хранилище.... Хранилище, созданное внутри папки, готово к использованию. Также мы создадим внутреннюю структуру папок нажав кнопку Создать структуру каталогов.

1.6. Импорт проекта

Сейчас у нас есть хранилище, но оно совершенно пустое в данный момент. Давайте предположим, что у меня есть набор файлов в C:\Projects\Widget1, который я хотел бы добавить. Перейдите к папке Widget1 в Проводнике и сделайте правый клик на ней.

Теперь выберите пункт TortoiseSVN→Импорт..., который вызовет диалог.

Рисунок 1.2. Диалог импорта



К хранилищу Subversion обращаются по URL-адресу, который позволяет нам указать хранилище где угодно в Интернете. В данном случае нам нужно указать на наше локальное хранилище, которое имеет URL-адрес `file:///c:/svn_repos/trunk`, и к которому мы добавляем имя нашего проекта Widget1. Обратите внимание, что после `file:` есть 3 слэша и везде используются прямые слэши.. Другая важная функция данного диалога - это окно. Сообщение импорта, в которое вы можете добавить сообщение о том, что вы делаете. Когда вам понадобится просмотреть историю проекта, эти сообщения будут ценным подспорьем для просмотра какие изменения и когда были сделаны. В нашем случае мы напишем что-нибудь простое как «Импорт проекта Виджет1». Нажмите OK, чтобы добавить папку в ваше хранилище.

1.7. Извлечение рабочей копии

Сейчас у нас есть проект в нашем хранилище, и нам надо создать рабочую копию для повседневной работы. Заметьте, что импортирование папки не превращает автоматически эту папку в рабочую копию. Для создания свежей рабочей копии в Subversion используется термин Извлечь. Мы собираемся извлечь папку Widget1 из нашего хранилища в папку для разработки называемую C:\Projects\Widget1-Dev. Создайте эту папку, затем сделайте правый клик на ней и выберите пункт TortoiseSVN→Извлечь.... Введите URL-адрес для извлечения, в данном случае

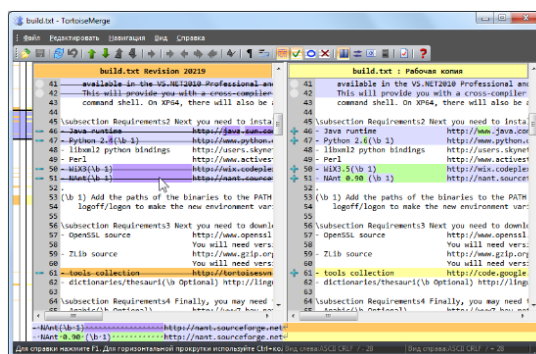
file:///c:/svn_repos/trunk/Widget1, и кликните на ОК. Наша папка для разработки заполнится файлами из хранилища.

Вы заметите, что внешний вид этой папки отличается от обычной папки. У каждого файла появился зелёный флажок в левом углу. Это значки статуса TortoiseSVN, которые присутствуют только в рабочей копии. Зелёный статус означает, что файл не отличается от версии файла, находящегося в хранилище.

1.8. Внесение изменений

Можно приступить к работе. В папке Виджет1-Девмы начинаем изменять файлы - предположим, мы вносим изменения в файлы Виджет1.сиПрочтиМеня.txt. Обратите внимание, что значки на этих файлах теперь стали красными и показывают, что изменения были сделаны локально. Но какие были изменения? Нажмите правой кнопкой на одном из изменённых файлов и выберите команду TortoiseSVN→Различия. Запустится инструмент TortoiseSVN для сравнения файлов и покажет какие точно строки в файлах были изменены.

Рисунок 1.3. Просмотрщик изменений в файлах



Ок, нас устраивают изменения, поэтому давайте обновим хранилище. Это действие называется Фиксироватьизменения. Нажмите правой кнопкой на папке Виджет1-Дев и выберите команду TortoiseSVN→Фиксировать. Появится диалог фиксации со списком изменённых файлов и напротив каждого будет галочка. Вы можете выбрать лишь несколько файлов из списка для фиксации, но в нашем случае мы будем фиксировать изменения в обоих файлах. Введите сообщение с описанием сделанных изменений и нажмите ОК. Появится диалог с прогрессом процесса фиксации файлов в хранилище и мы закончили фиксацию.

1.9. Добавление новых файлов

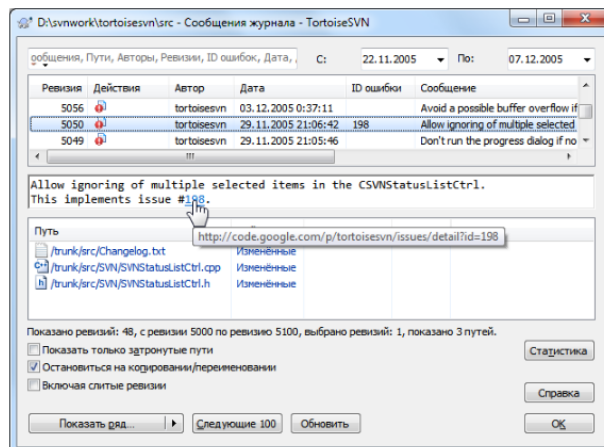
Во время работы над проектом, вам понадобится добавлять новые файлы -предположим мы добавили новые функции в файле Экстра.си добавили справку в существующем файлеСоздать файл. Нажмите правой кнопкой на папке и выберите команду TortoiseSVN→Добавить. Диалог добавления показывает все неверсированные файлы и вы можете выбрать те файлы, которые вы хотите добавить. Другой способ добавления файлов -это нажать правой кнопкой на самом файле и выбрать команду TortoiseSVN→Добавить.

Теперь, если вы откроете папку для фиксации, новый файл будет отображаться как Добавлен и существующий файл как Изменён. Обратите внимание, что вы можете дважды нажать на изменённый файл, чтобы просмотреть какие именно изменения были сделаны.

1.10. Просмотр истории проекта

Одной из самых полезных функций TortoiseSVN является диалоговое окно журнала. Оно показывает список всех фиксаций изменений в файле или папке, а также все те детальные сообщения, которые вы вводили при фиксации изменений;

Рисунок 1.4. Диалоговое окно журнала



Верхняя панель показывает список всех фиксированных ревизий вместе с началом сообщения фиксации. Если вы выберете одну из этих ревизий, то средняя панель отобразит полное сообщение журнала для той ревизии и нижняя панель покажет список измененных файлов и папок.

У каждой из этих панелей есть контекстное меню, которое предоставляет много других способов использования информации. В нижней панели вы можете дважды нажать на файл, чтобы просмотреть какие именно изменения были внесены в той ревизии. Прочтите «Диалоговое окно журнала ревизий», чтобы узнать больше.

1.11. Отмена изменений

Одной общей функцией всех систем управления ревизиями является функция, которая позволяет вам отменить изменения, которые вы внесли ранее. Как вы и догадались, в TortoiseSVN это легко сделать

Если вы хотите избавиться от изменений, которые вы еще не успели фиксировать и восстановить нужный файл в том виде, в котором он был перед началом изменений, то выберите команду TortoiseSVN→Убрать изменения. Это действие отменит ваши изменения (в Корзину) и вернет фиксированную версию файла, с которой вы начинали.

Если же вы хотите убрать лишь некоторых изменения, то вы можете использовать инструмент TortoiseMerge для просмотра изменений и выборочного удаления измененных строк.

Если вы хотите отменить действия определенной ревизии, то начните с диалогового окна журнала и найдите проблемную ревизию. Выберите команду.

Контекстное меню→Отменить изменения из этой ревизии те изменения будут отменены

Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

Задание 6

Практическая работа Стадия эскизный проект

Проверяемые результаты обучения: ПК2.1 ПК2.2 ПК2.3 ПК2.4 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

В соответствии с вариантом задания (работа 1) выполнить разработку документов на стадии эскизный проект, которое должно включать:

- выбор метода решения и языка программирования;
- спецификации процессов;
- все полученные диаграммы;
- словарь терминов.

Эскизное проектирование – разработка предварительных проектных решений по системе и её частям. Итоговым документом выполнения работ на данной стадии проектирования является эскизный проект, который содержит принципиальные конструкторские и схемные решения объекта разработки, а также данные, определяющие его назначение и основные параметры (при проектировании программного обеспечения системы эскизный проект должен содержать полную спецификацию разрабатываемых программ).

Эскизный проект – это комплект проектных документов на АИС, разрабатываемых на этапе эскизного проектирования, утвержденный в установленном порядке, содержащий предварительные общесистемные решения по выбранному на этапе концептуального проектирования варианту АИС и отдельным видам ее обеспечения и достаточный для разработки технического проекта.

Выполнение эскизного проектирования не является строго обязательной стадией. Если основные проектные решения определены ранее или достаточно очевидны для конкретной АИС и объекта автоматизации, то эта стадия может быть исключена из общей последовательности работ.

Основой для эскизного проектирования являются документы, разработанные на предпроектной стадии: отчет о предпроектном обследовании, аван-проект, ТЗ, ТЭО.

В ходе эскизного проектирования принимаются преимущественно общесистемные проектные решения, что же касается обеспечивающих подсистем, то на данном этапе, принимаемые проектные решения касаются преимущественно технического обеспечения и частично программного.

Этапы	Состав решаемых задач	Состав основных разрабатываемых документов	Нормативная база проектирования
Эскизное проектирование	1. Разработка предварительных проектных решений по системе и ее частям 1.1. Определение функций АИС в целом и ее подсистем, их целей и эффекта 1.2. Определение состава комплексов задач и отдельных задач. 1.3. Разработка концепции информационной базы, ее укрупненной структуры	Эскизный проект, включая: - Ведомость эскизного проекта; - Пояснительная записка к эскизному проекту; - Схема организационной структуры; - Схема функциональной структуры	РД 50-34.698-90 Методические указания. ИТ. Комплекс стандартов и руководящих документов на АС. АС. Требования к содержанию документов.

	1.4. Определение функций системы управления базой данных, функций и параметров основных программных средств. 1.5. Определение состава вычислительной системы. 2. Разработка документации на АИС и ее части		
--	--	--	--

Содержание работ на этапе эскизного проектирования:

1. Предварительная разработка структуры входных и выходных данных.
2. Уточнение методов решения задачи.
3. Разработка общего описания алгоритма решения задачи
4. Разработка технико-экономического обоснования.
5. Разработка пояснительной записки.
6. Согласование и утверждение эскизного проекта.

Важнейшим документом, в котором отражаются предварительные проектные решения является пояснительная записка к эскизному проекту, содержание ее регламентируется руководящим документом 50-34.698-90. Пояснительная записка должна содержать 4 раздела в том числе:

- 1) общие положения;
- 2) описание процесса деятельности;
- 3) основные технические решения;
- 4) мероприятия по подготовке объекта автоматизации к вводу системы в действие.

Разделы 1 и 2 по своему наполнению являются адекватными соответствующим разделам ТЗ.

В 3-м разделе приводят:

- 1) решения по структуре системы, подсистем, средствам и способам связи для информационного обмена между компонентами системы, подсистем;
- 2) решения по взаимосвязям АС со смежными системами, обеспечению ее совместимости;
- 3) решения по режимам функционирования, диагностированию работы системы;
- 4) решения по численности, квалификации и функциям персонала АС, режимам его работы, порядку взаимодействия;
- 5) сведения об обеспечении заданных в техническом задании (ТЗ) потребительских характеристик системы (подсистем), определяющих ее качество;
- 6) состав функций, комплексов задач (задач), реализуемых системой (подсистемой);
- 7) решения по комплексу технических средств, его размещению на объекте;
- 8) решения по составу информации, объему, способам ее организации, видам машинных носителей, входным и выходным документам и сообщениям, последовательности обработки информации и другим компонентам;
- 9) решения по составу программных средств, языкам деятельности, алгоритмам процедур и операций и методам их реализации.

В 4-м разделе приводят:

- 1) мероприятия по приведению информации к виду, пригодному для обработки на ЭВМ;
- 2) мероприятия по обучению и проверке квалификации персонала;
- 3) мероприятия по созданию необходимых подразделений и рабочих мест;
- 4) мероприятия по изменению объекта автоматизации;
- 5) другие мероприятия, исходящие из специфических особенностей создаваемых АС.

К числу других документов, разрабатываемых на этапе эскизного проектирования и содержащих общесистемные проектные решения относятся:

- схема организационной структуры;
- схема функциональной структуры;
- ведомость эскизного проекта;
- пояснительная записка.

Вместе с тем на этапе эскизного проектирования создаются документы, содержащие локальные проектные решения, относящиеся к техническому обеспечению.

Результаты проделанной работы на этапе эскизного проектирования оформляются в виде комплекта проектных документов, которые предваряются пояснительной запиской к эскизному проекту.

Состав комплекта проектных документов фиксируется в ведомости эскизного проекта.

Все проектные решения отражаются в данном комплекте проектной документации, согласовываются и утверждаются в установленном порядке. Критерием достаточности содержания проектных документов, разрабатываемых на стадии эскизного проектирования, является возможность выполнения работ на этапе технического проектирования.

На основе технического задания и эскизного проекта разрабатывается технический проект АИС.

Теоретическая часть. Разработка спецификаций

Разработка программного обеспечения начинается с анализа требований к нему. В результате анализа получают спецификации разрабатываемого программного обеспечения, строят общую модель его взаимодействия с пользователем или другими программами и конкретизируют его основные функции.

При структурном подходе к программированию на этапе анализа и определения спецификаций разрабатывают три типа моделей: модели функций, модели данных и модели потоков данных. Поскольку разные модели описывают проектируемое программное обеспечение с разных сторон, рекомендуется использовать сразу несколько моделей, разрабатываемых в виде диаграмм, и пояснять их текстовыми описаниями, словарями и т. п.

Структурный анализ предполагает использование следующих видов моделей:

- диаграмм потоков данных (DFD — Data Flow Diagrams), описывающих взаимодействие источников и потребителей информации через процессы, которые должны быть реализованы в системе;
- диаграмм «сущность—связь» (ERD — Entity-Relationship Diagrams), описывающих базы данных разрабатываемой системы;
- диаграмм переходов состояний (STD — State Transition Diagrams), характеризующих поведение системы во времени;
- функциональных диаграмм (методика SADT);
- спецификаций процессов;
- словаря терминов.

Спецификации процессов

Спецификации процессов обычно представляют в виде краткого текстового описания, схем алгоритмов, псевдокодов, Flow-форм или диаграмм Насси — Шнейдермана (см. разд. 3.5.1).

Словарь терминов

Словарь терминов представляет собой краткое описание основных понятий, используемых при составлении спецификаций. Он должен включать определение основных понятий предметной области, описание структур элементов данных, их типов и форматов, а также всех сокращений и условных обозначений (см. разд. 3.5.2).

Диаграммы переходов состояний

С помощью *диаграмм переходов состояний* можно моделировать последующее функционирование системы на основе ее предыдущего и текущего функционирования.

Моделируемая система в любой заданный момент времени находится точно в одном из конечного множества состояний. С течением времени она может изменить свое состояние, при этом переходы между состояниями должны быть точно определены (см. разд. 3.5.3).

Функциональные диаграммы

Функциональные диаграммы отражают взаимосвязи функций разрабатываемого программного обеспечения.

Они создаются на ранних этапах проектирования систем, для того чтобы помочь проектировщику выявить основные функции и составные части проектируемой системы и, по возможности, обнаружить и устранить существенные ошибки. Для создания функциональных диаграмм предлагается использовать методологию SADT (см. разд. 3.5.4).

Диаграммы потоков данных

Для описания потоков информации в системе применяются *диаграммы потоков данных* (DFD — Data flow diagrams). DFD позволяет описать требуемое поведение системы в виде совокупности процессов, взаимодействующих посредством связывающих их потоков данных. DFD показывает, как каждый из процессов преобразует свои входные потоки данных в выходные потоки данных и как процессы взаимодействуют между собой (см. разд. 3.5.5).

Диаграммы «сущность—связь»

Диаграмма сущность—связь — инструмент разработки моделей данных, обеспечивающий стандартный способ определения данных и отношений между ними. Она включает *сущности* и взаимосвязи, отражающие основные бизнес-правила предметной области. Такая диаграмма не слишком детализирована, в нее включаются основные *сущности* и *связи* между ними, которые удовлетворяют требованиям, предъявляемым к ИС

Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

Задание 7

Практическая работа Стадия технический проект

Проверяемые результаты обучения: ПК2.1 ПК2.2 ПК2.3 ПК2.4 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

1. На основе технического задания из работы № 1 и спецификаций из практической работы № 2 разработать уточненные алгоритмы программ, составляющих заданный программный модуль. Использовать метод пошаговой детализации
2. На основе уточненных и доработанных алгоритмов разработать структурную схему программного продукта
3. Разработать функциональную схему программного продукта
4. Представить структурную схему в виде структурных карт Консантайна
5. Представить структурную схему в виде структурных карт Джексона
6. Оформить результаты, используя MS Office или MS Visio в виде технического проекта.

ВИДЫ ПРЕДМЕТНЫХ ОБЛАСТЕЙ

1. Страховая компания
2. Гостиница
3. Ломбард
4. Реализация готовой продукции
5. Ведение заказов
6. Бюро по трудоустройству
7. Нотариальная контора

8. Фирма по продаже запчастей
9. Курсы повышения квалификации
10. Туристическая фирма
11. Грузовые перевозки
12. Ювелирная мастерская
13. Парикмахерская
14. Химчистка
15. Сдача в аренду торговых площадей

Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

Задание 8

Практическая работа Создание приложения

Проверяемые результаты обучения: ПК2.1 ПК2.2 ПК2.3 ПК2.4 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

По результатам № 1—4 написать код программ для решения поставленной задачи на языке программирования, выбранном на этапе эскизного проектирования.

2. Отладить программный модуль.

3. Получить результаты работы.

4. Оформить документацию к разработанному программному обеспечению (Листинги программ, интерфейса пользователя).

ВИДЫ ПРЕДМЕТНЫХ ОБЛАСТЕЙ

1. Страховая компания
2. Гостиница
3. Ломбард
4. Реализация готовой продукции
5. Ведение заказов
6. Бюро по трудоустройству
7. Нотариальная контора
8. Фирма по продаже запчастей
9. Курсы повышения квалификации
10. Туристическая фирма
11. Грузовые перевозки
12. Ювелирная мастерская
13. Парикмахерская
14. Химчистка
15. Сдача в аренду торговых площадей

Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

Задание 9

Практическая работа Создание инсталляционного пакета

Проверяемые результаты обучения: ПК2.1 ПК2.2 ПК2.3 ПК2.4 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

Создать инсталляционный пакет для программы, созданной на предыдущем занятии.

ВИДЫ ПРЕДМЕТНЫХ ОБЛАСТЕЙ

1. Страховая компания
2. Гостиница
3. Ломбард
4. Реализация готовой продукции
5. Ведение заказов
6. Бюро по трудоустройству
7. Нотариальная контора
8. Фирма по продаже запчастей
9. Курсы повышения квалификации
10. Туристическая фирма
11. Грузовые перевозки
12. Ювелирная мастерская
13. Парикмахерская
14. Химчистка
15. Сдача в аренду торговых площадей

Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

Задание 10

Практическая работа Создание справки для приложения

Проверяемые результаты обучения: ПК2.1 ПК2.2 ПК2.3 ПК2.4 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

Создать справку для ранее созданной программы (в форматах doc, html)

Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

Тема 2.1.2. Описание и анализ требований. Диаграммы IDEF

Задание 1

Проверяемые результаты обучения: ПК2.1 ПК2.5

Вид учебной работы: аудиторная

Задание. Ответить на устные вопросы:

1. Описание и оформление требований (спецификация). Анализ требований и стратегии выбора решения
2. Выбор архитектуры ПО

3. Структура и формат данных
4. Модульное программирование
5. Анализ требований и определение спецификаций при структурном подходе
6. Анализ требований и определение спецификаций при объектном подходе
7. Диаграммы потоков данных. Диаграммы сущность-связь.
8. Описание требований: унифицированный язык моделирования - краткий словарь. Диаграммы UML.

Критерии оценки: Полнота ответа на поставленные вопросы

«5»- Правильный уверенный ответ

«4» - Уверенный ответ, допущена одна неточность

«3» - Ответ близкий к правильному, с несколькими неточностями

«2» - Неправильный ответ, отказывается отвечать

Задание 2

Лабораторная работа Построение моделей программных систем с использованием структурного подхода.

Проверяемые результаты обучения: ПК2.1 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

В соответствии с вариантом задания выполнить разработку спецификации:

Процессов

Диаграммы переходов состояний

Варианты заданий

Общее задание:

Вывести элементы массива разными способами: строкой, столбцом, по диагонали

Дополнительные задания:

$$\sum_{i=1}^{10} \prod_{j=1}^5 \frac{1}{ij}$$

$$2. \prod_{i=1}^{100} \prod_{j=i}^{100} \frac{i-j+1}{i+j}$$

$$3. \sum_{i=1}^{100} \sum_{j=1}^i \frac{1}{2j+1}$$

$$4. \sum_{R=1}^n \frac{1}{(R^2)!}$$

$$5. \prod_{R=1}^n R^R$$

$$6. \sum_{R=1}^n (-1)^R (R^R + 1)!$$

$$7. \sum_{i=1}^n \frac{(2i)! + |x|}{(i^2)!}$$

$$8. \sum_{R=1}^n R^R x^{2R}$$

$$9. \prod_{R=1}^n \prod_{m=R}^n \frac{X+r}{m}$$

$$10. \sum_{R=1}^n (-1)^r \frac{x^r}{(r+1)!}$$

$$11. \sum_{R=1}^n \frac{x^r}{r!}$$

$$12. \sum_{R=1}^n \frac{1}{r!} + \sqrt{|x|}$$

$$13. \sum_{R=1}^n \frac{(-1)^r}{(2r+1)r}$$

$$14. \prod_{r=1}^n \left(1 + \frac{\sin(rx)}{r!}\right)$$

Написать программу решения задачи
Оформить отчет.

Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

Задание 3

Лабораторная работа Построение моделей программных систем с использованием объектно-ориентированного подхода.

Проверяемые результаты обучения: ПК2.1 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

В соответствии с вариантом задания выполнить разработку спецификации:

Функциональную диаграмму

Диаграмму потоков данных

Диаграмму сущность связь

Варианты заданий

Общее задание:

Вывести элементы массива разными способами: строкой, столбцом, по диагонали

Дополнительные задания:

$$\sum_{i=1}^{10} \prod_{j=1}^5 \frac{1}{ij}$$

$$2. \prod_{i=1}^{100} \prod_{j=i}^{100} \frac{i-j+1}{i+j}$$

$$3. \sum_{i=1}^{100} \sum_{j=1}^i \frac{1}{2j+1}$$

$$4. \sum_{R=1}^n \frac{1}{(R^2)!}$$

$$5. \prod_{R=1}^n R^R$$

$$6. \sum_{R=1}^n (-1)^r (r^r + 1)!$$

$$7. \sum_{i=1}^n \frac{(2i)! + |x|}{(i^2)!}$$

$$8. \sum_{R=1}^n r^r x^{2r}$$

$$9. \prod_{R=1}^n \prod_{m=R}^n \frac{X+r}{m}$$

$$10. \sum_{R=1}^n (-1)^r \frac{x^r}{(r+1)!}$$

$$11. \sum_{R=1}^n \frac{x^r}{r!}$$

$$12. \sum_{R=1}^n \frac{1}{r!} + \sqrt{|x|}$$

$$13. \sum_{R=1}^n \frac{(-1)^r}{(2r+1)r}$$

$$14. \prod_{r=1}^n \left(1 + \frac{\sin(rx)}{r!}\right)$$

Написать программу решения задачи
Оформить отчет.

Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

Задание 4

Лабораторная работа Создание диаграммы Вариантов использования

Проверяемые результаты обучения: ПК2.1 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

Составить диаграмму вариантов использования вымышленной компании. Компания занимается в основном сборкой и продажей настольных компьютеров и ноутбуков. Компания не производит компоненты самостоятельно, а только собирает и тестирует компьютеры.

Основные процедуры в компании таковы:

- продавцы принимают заказы клиентов;
- операторы группируют заказы по типам компьютеров;
- операторы собирают и тестируют компьютеры;
- операторы упаковывают компьютеры согласно заказам;
- кладовщик отгружает клиентам заказы.

Компания использует купленную бухгалтерскую информационную систему, которая позволяет оформить заказ, счет и отследить платежи по счетам.

Запустите *BPwin*. (Кнопка «*Start/BPwin*»). Если появляется диалог «*ModelMart Connection Manager*», нажмите на кнопку *Cancel*.

Щелкните по кнопке  Появляется диалог *I would like to*. Внесите имя модели "Деятельность компании" и выберите Type - *IDEF0*. Нажмите *OK*.

Появляется меню *Properties for New Models*. Во вкладке *General* вводится фамилия и инициалы автора, остальные вкладки используются для определения настроек проекта. Автоматически создается контекстная диаграмма.

Обратите внимание на кнопку  на панели инструментов. Эта кнопка включает и выключает инструмент просмотра и навигации – Model Explorer (появляется слева). Model Explorer имеет три вкладки: *Activities*, *Diagrams* и *Objects*. Во вкладке *Activities* щелчок правой кнопкой по объекту позволяет редактировать его свойства. Если вам непонятно, как выполнить то или иное действие, вы можете вызвать помощь - клавиша F1 или меню *Help*.

Измените шрифт для правильного отображения русских букв. Зайдите в меню *Model/Default Fonts* и выберите *Parent Diagram Text*. Поставьте галочку *change all occurrences* и нажмите *OK*. Если ничего не изменилось, повторите эту операцию с другим подменю, например: *Parent Diagram Title Text*.

Для изменений свойств модели используется меню *Model Properties*. По умолчанию тип модели: *Time Frame: AS-IS*. Во вкладке *Purpose* внесите цель – «*Purpose: Моделировать текущие (AS-IS) бизнес-процессы компании*» и точку зрения – «*Viewpoint: Директор*».

Во вкладке *Definition* внесите определение «Это учебная модель, описывающая деятельность компании» и цель *Scope*: «Общее управление бизнесом компании: исследование рынка, закупка компонентов, сборка, тестирование и продажа продуктов».

Перейдите на контекстную диаграмму и правой кнопкой мыши щелкните по работе. В контекстном меню выберите *Name*. Во вкладке *Name* внесите имя «Деятельность компании».

Во вкладке *Definition* внесите определение «Текущие бизнес-процессы компании».

С помощью кнопки  внесите текст в поле диаграммы - точку зрения и цель (рис. 2.1.6).

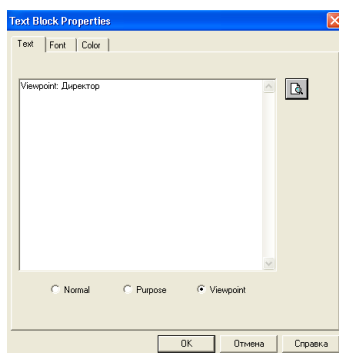


Рис.2.1.6 Внесение текста в поле диаграммы с помощью редактора *Text Block Editor*

Создайте стрелки на контекстной диаграмме с помощью меню *Model/Arrow Editor* согласно табл. 2.1.2.

Таблица 2.1.2 Стрелки контекстной диаграммы

<i>Arrow Name</i>	<i>Arrow Type</i>	<i>Arrow Definition</i>
Бухгалтерская система	Механизм	Оформление счетов, оплата счетов, работа с заказами
Звонки клиентов	Вход	Запросы информации, заказы, техподдержка и т.д.
Правила и процедуры	Управление	Правила продаж, инструкции по сборке, процедуры тестирования, критерии производительности и т.д.
Проданные продукты	Выход	Настольные и портативные компьютеры

Соедините стрелки с работой «Деятельность компании» согласно рис.2.1.7

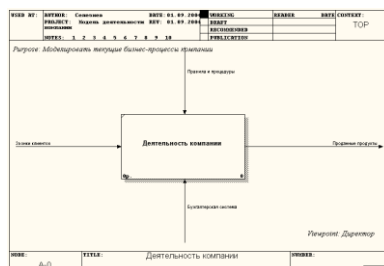


Рис. 2.1.7. Контекстная диаграмма

Создайте отчет по модели. Меню *Tools/Reports/Model Report* (рис. 2.1.8).

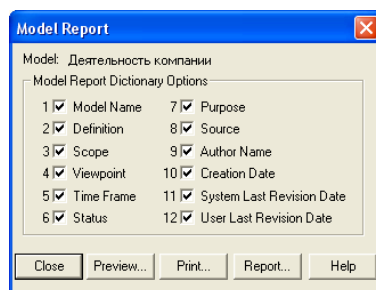


Рис. 2.1.8.(а) Отчет *Model Report*

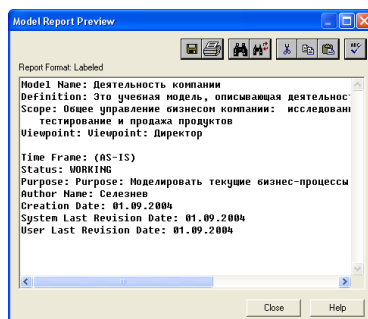


Рис. 2.1.8.(б) Отчет *Model Report*

Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

Задание 5

Лабораторная работа Создание диаграммы последовательности

Проверяемые результаты обучения: ПК2.1 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

В соответствии с вариантом задания выполнить разработку: Диаграммы последовательности в BPWIN

Варианты заданий:

Предметная область «Компания по сборке компьютеров»

Компания занимается в основном сборкой и продажей настольных компьютеров и ноутбуков. Компания не производит компоненты самостоятельно, а только собирает и тестирует компьютеры.

Основные процедуры в компании таковы:

- продавцы принимают заказы клиентов;
- операторы группируют заказы по типам компьютеров;
- операторы собирают и тестируют компьютеры;
- операторы упаковывают компьютеры согласно заказам;
- кладовщик отгружает клиентам заказы.

Компания использует купленную бухгалтерскую информационную систему, которая позволяет оформить заказ, счет и отследить платежи по счетам.

Предметная область «Маршруты городских автобусов»

Фирма, занимающаяся пассажирскими перевозками, имеет в своем распоряжении парк автобусов и штат водителей. Наиболее оптимальным распределением автобусов и водителей по маршрутам занимается диспетчерская. Для этого она собирает информацию об автобусах, водителях и маршрутах.

К основным характеристикам автобуса можно отнести количество мест, комфортабельность, объем двигателя, расход топлива. Некоторые из этих показателей учитываются при распределении автобусов по маршрутам. Например, немногочисленные маршруты могут обслуживаться автобусами с небольшим количеством мест. Так же автобусы должны иметь уникальные номера в учетном списке (это могут быть номерные знаки).

Информация о водителях имеется в отделе кадров. Диспетчерской необходима информация о возрасте, стаже водителя, а так же его фамилия, имя, отчество. Например, водитель с высоким стажем будет отправлен на более многочисленный и сложный маршрут.

Информация о маршрутах включает в себя номера маршрутов, их протяженность в км, популярность (число пассажиров в час), время обслуживания. Например, протяженность и время обслуживания плюс информация о расходе топлива автобуса нужны, чтобы рассчитывать количество топлива на маршрут. Можно так же, когда уже конкретно распределены автобусы и водители по маршрутам, учитывать информацию о дневной выручке на маршруте.

Предметная область «Комплектующие компьютера»

С каждым днем появляются все новые и новые модели комплектующих на компьютер. Улучшаются показатели, функции уже существующих. Появилось множество фирм, сервисных центров, торгующих компьютерами и комплектующими. Многие из них комплектуют компьютер по желанию клиента. В этом случае как раз пригодится систематизированный список комплектующих (прайс-лист). Он должен содержать все необходимые данные: вид, наименование, модель, параметры, цену (например, вид: материнская плата, наименование: MB,

модель: VX Pro+). Обязательно должна быть указана совместимость/несовместимость одного типа комплектующего с другими. Очень часто из-за несовместимости происходят поломки, сбои, устройства сгорают, приходят в негодность.

Каждая единица устройства имеет свой серийный номер, однозначно его идентифицирующий. Этот номер помимо учета устройств, необходим при предоставлении гарантийных талонов клиентам. Фирма по гарантийному талону будет обслуживать лишь то устройство, номер которого указан в талоне.

Предметная область «Оборудование, собираемое мебельной фабрикой «Лада»

«Лада» производит следующие типы оборудования: аптечное, торговое и офисное. В каждом типе оборудования имеются свои категории оборудования (столы, шкафы, стеллажи и т.д.). Единица оборудования имеет название, габариты, цвет, материал, из которого собрана, цену. На фабрике каждому виду и каждой категории оборудования присваиваются шифры (шифр вида и шифр категории). Эти шифры нужны для однозначной идентификации оборудования (например, шифр ША2-шкаф аптечный с № 2 по списку).

Произведенная мебель реализуется отделом сбыта. Выше перечисленная информация, характеризующая оборудование, может быть занесена в прайс - листы и предоставляться покупателям.

Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

Задание 6

Лабораторная работа Создание кооперативной диаграммы

Проверяемые результаты обучения: ПК2.1 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

В соответствии с вариантом задания выполнить разработку: Кооперативной диаграммы

Варианты заданий:

Предметная область «Компания по сборке компьютеров»

Компания занимается в основном сборкой и продажей настольных компьютеров и ноутбуков. Компания не производит компоненты самостоятельно, а только собирает и тестирует компьютеры.

Основные процедуры в компании таковы:

- продавцы принимают заказы клиентов;
- операторы группируют заказы по типам компьютеров;
- операторы собирают и тестируют компьютеры;
- операторы упаковывают компьютеры согласно заказам;
- кладовщик отгружает клиентам заказы.

Компания использует купленную бухгалтерскую информационную систему, которая позволяет оформить заказ, счет и отследить платежи по счетам.

Предметная область "Маршруты городских автобусов"

Фирма, занимающаяся пассажирскими перевозками, имеет в своем распоряжении парк автобусов и штат водителей. Наиболее оптимальным распределением автобусов и водителей по маршрутам занимается диспетчерская. Для этого она собирает информацию об автобусах, водителях и маршрутах. К основным характеристикам автобуса можно отнести количество

мест, комфортабельность, объем двигателя, расход топлива. Некоторые из этих показателей учитываются при распределении автобусов по маршрутам. Например, немногочисленные маршруты могут обслуживаться автобусами с небольшим количеством мест. Так же автобусы должны иметь уникальные номера в учетном списке (это могут быть номерные знаки).

Информация о водителях имеется в отделе кадров. Диспетчерской необходима информация о возрасте, стаже водителя, а также его фамилия, имя, отчество. Например, водитель с высоким стажем будет отправлен на более многочисленный и сложный маршрут.

Информация о маршрутах включает в себя номера маршрутов, их протяженность в км, популярность (число пассажиров в час), время обслуживания. Например, протяженность и время обслуживания плюс информация о расходе топлива автобуса нужны, чтобы рассчитывать количество топлива на маршрут. Можно так же, когда уже конкретно распределены автобусы и водители по маршрутам, учитывать информацию о дневной выручке на маршруте.

Предметная область "Комплектующие компьютера"

С каждым днем появляются все новые и новые модели комплектующих на компьютер. Улучшаются показатели, функции уже существующих. Появилось множество фирм, сервисных центров, торгующих компьютерами и комплектующими. Многие из них комплектуют компьютер по желанию клиента. В этом случае как раз пригодится систематизированный список комплектующих (прайс-лист). Он должен содержать все необходимые данные: вид, наименование, модель, параметры, цену (например, вид: материнская плата, наименование: MB, модель: VX Pro+). Обязательно должна быть указана совместимость/несовместимость одного типа комплектующего с другими. Очень часто из-за несовместимости происходят поломки, сбои, устройства сгорают, приходят в негодность.

Каждая единица устройства имеет свой серийный номер, однозначно его идентифицирующий. Этот номер помимо учета устройств, необходим при предоставлении гарантийных талонов клиентам. Фирма по гарантийному талону будет обслуживать лишь то устройство, номер которого указан в талоне.

Предметная область «Оборудование, собираемое мебельной фабрикой «Лада»

«Лада» производит следующие типы оборудования: аптечное, торговое и офисное. В каждом типе оборудования имеются свои категории оборудования (столы, шкафы, стеллажи и т.д.). Единица оборудования имеет название, габариты, цвет, материал, из которого собрана, цену. На фабрике каждому виду и каждой категории оборудования присваиваются шифры (шифр вида и шифр категории). Эти шифры нужны для однозначной идентификации оборудования (например, шифр ША2-шкаф аптечный с № 2 по списку).

Произведенная мебель реализуется отделом сбыта. Выше перечисленная информация, характеризующая оборудование, может быть занесена в прайс - листы и предоставляться покупателям.

Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

Задание 7

Лабораторная работа Создание диаграммы классов

Проверяемые результаты обучения: ПК2.1 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

В соответствии с вариантом задания выполнить разработку: Диаграммы классов

Варианты заданий:

Предметная область «Компания по сборке компьютеров»

Компания занимается в основном сборкой и продажей настольных компьютеров и ноутбуков. Компания не производит компоненты самостоятельно, а только собирает и тестирует компьютеры.

Основные процедуры в компании таковы:

- продавцы принимают заказы клиентов;
- операторы группируют заказы по типам компьютеров;
- операторы собирают и тестируют компьютеры;
- операторы упаковывают компьютеры согласно заказам;
- кладовщик отгружает клиентам заказы.

Компания использует купленную бухгалтерскую информационную систему, которая позволяет оформить заказ, счет и отследить платежи по счетам.

Предметная область «Маршруты городских автобусов»

Фирма, занимающаяся пассажирскими перевозками, имеет в своем распоряжении парк автобусов и штат водителей. Наиболее оптимальным распределением автобусов и водителей по маршрутам занимается диспетчерская. Для этого она собирает информацию об автобусах, водителях и маршрутах.

К основным характеристикам автобуса можно отнести количество мест, комфортабельность, объем двигателя, расход топлива. Некоторые из этих показателей учитываются при распределении автобусов по маршрутам. Например, немногочисленные маршруты могут обслуживаться автобусами с небольшим количеством мест. Так же автобусы должны иметь уникальные номера в учетном списке (это могут быть номерные знаки).

Информация о водителях имеется в отделе кадров. Диспетчерской необходима информация о возрасте, стаже водителя, а так же его фамилия, имя, отчество. Например, водитель с высоким стажем будет отправлен на более многочисленный и сложный маршрут.

Информация о маршрутах включает в себя номера маршрутов, их протяженность в км, популярность (число пассажиров в час), время обслуживания. Например, протяженность и время обслуживания плюс информация о расходе топлива автобуса нужны, чтобы рассчитывать количество топлива на маршрут. Можно так же, когда уже конкретно распределены автобусы и водители по маршрутам, учитывать информацию о дневной выручке на маршруте.

Предметная область «Комплектующие компьютера»

С каждым днем появляются все новые и новые модели комплектующих на компьютер.

Улучшаются показатели, функции уже существующих.

Появилось множество фирм, сервисных центров, торгующих компьютерами и комплектующими. Многие из них комплектуют компьютер по желанию клиента. В этом случае как раз пригодится систематизированный список комплектующих (прайс-лист). Он должен содержать все необходимые данные: вид, наименование, модель, параметры, цену (например, вид: материнская плата, наименование: MB, модель: VX Pro+). Обязательно должна быть указана совместимость/несовместимость одного типа комплектующего с другими. Очень часто из-за несовместимости происходят поломки, сбои, устройства сгорают, приходят в негодность.

Каждая единица устройства имеет свой серийный номер, однозначно его идентифицирующий. Этот номер помимо учета устройств, необходим при предоставлении гарантийных талонов клиентам. Фирма по гарантийному талону будет обслуживать лишь то устройство, номер которого указан в талоне.

Предметная область «Оборудование, собираемое мебельной фабрикой «Лада»

"Лада" производит следующие типы оборудования: аптечное, торговое и офисное. В каждом типе оборудования имеются свои категории оборудования (столы, шкафы, стеллажи и т.д.). Единица оборудования имеет название, габариты, цвет, материал, из которого собрана, цену. На фабрике каждому виду и каждой категории оборудования присваиваются шифры (шифр вида и шифр категории). Эти шифры нужны для однозначной идентификации оборудования (например, шифр ША2-шкаф аптечный с № 2 по списку).

Произведенная мебель реализуется отделом сбыта. Выше перечисленная информация, характеризующая оборудование, может быть занесена в прайс - листы и предоставляться покупателям.

Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

Задание 8

Лабораторная работа Создание контекстной диаграммы в среде *BPwin*.

Проверяемые результаты обучения: ПК2.1 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

Создать контекстную диаграмму в среде *BPwin*

Варианты заданий:

Предметная область «Компания по сборке компьютеров»

Компания занимается в основном сборкой и продажей настольных компьютеров и ноутбуков. Компания не производит компоненты самостоятельно, а только собирает и тестирует компьютеры.

Основные процедуры в компании таковы:

- продавцы принимают заказы клиентов;
- операторы группируют заказы по типам компьютеров;
- операторы собирают и тестируют компьютеры;
- операторы упаковывают компьютеры согласно заказам;
- кладовщик отгружает клиентам заказы.

Компания использует купленную бухгалтерскую информационную систему, которая позволяет оформить заказ, счет и отследить платежи по счетам.

Предметная область «Маршруты городских автобусов»

Фирма, занимающаяся пассажирскими перевозками, имеет в своем распоряжении парк автобусов и штат водителей. Наиболее оптимальным распределением автобусов и водителей по маршрутам занимается диспетчерская. Для этого она собирает информацию об автобусах, водителях и маршрутах.

К основным характеристикам автобуса можно отнести количество мест, комфортабельность, объем двигателя, расход топлива. Некоторые из этих показателей учитываются при распределении автобусов по маршрутам. Например, немногочисленные маршруты могут обслуживаться автобусами с небольшим количеством мест. Так же автобусы должны иметь уникальные номера в учетном списке (это могут быть номерные знаки).

Информация о водителях имеется в отделе кадров. Диспетчерской необходима информация о возрасте, стаже водителя, а так же его фамилия, имя, отчество. Например, водитель с высоким стажем будет отправлен на более многочисленный и сложный маршрут.

Информация о маршрутах включает в себя номера маршрутов, их протяженность в км, популярность (число пассажиров в час), время обслуживания. Например, протяженность и время обслуживания плюс информация о расходе топлива автобуса нужны, чтобы рассчитывать количество топлива на маршрут. Можно так же, когда уже конкретно распределены автобусы и водители по маршрутам, учитывать информацию о дневной выручке на маршруте.

Предметная область «Комплектующие компьютера»

С каждым днем появляются все новые и новые модели комплектующих на компьютер. Улучшаются показатели, функции уже существующих.

Появилось множество фирм, сервисных центров, торгующих компьютерами и комплектующими. Многие из них комплектуют компьютер по желанию клиента. В этом случае как раз пригодится систематизированный список комплектующих (прайс-лист). Он должен содержать все необходимые данные: вид, наименование, модель, параметры, цену (например, вид: материнская плата, наименование: MB, модель: VX Pro+). Обязательно должна быть указана совместимость/несовместимость одного типа комплектующего с другими. Очень часто из-за несовместимости происходят поломки, сбои, устройства сгорают, приходят в негодность.

Каждая единица устройства имеет свой серийный номер, однозначно его идентифицирующий. Этот номер помимо учета устройств, необходим при предоставлении гарантийных талонов клиентам. Фирма по гарантийному талону будет обслуживать лишь то устройство, номер которого указан в талоне.

Предметная область «Оборудование, собираемое мебельной фабрикой «Лада»

«Лада» производит следующие типы оборудования: аптечное, торговое и офисное. В каждом типе оборудования имеются свои категории оборудования (столы, шкафы, стеллажи и т.д.). Единица оборудования имеет название, габариты, цвет, материал, из которого собрана, цену. На фабрике каждому виду и каждой категории оборудования присваиваются шифры (шифр вида и шифр категории). Эти шифры нужны для однозначной идентификации оборудования (например, шифр ША2-шкаф аптечный с № 2 по списку).

Произведенная мебель реализуется отделом сбыта. Выше перечисленная информация, характеризующая оборудование, может быть занесена в прайс - листы и предоставляться покупателям.

Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

Задание 9

Лабораторная работа Создание диаграммы декомпозиции

Проверяемые результаты обучения: ПК2.1 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

Создать диаграмму декомпозиции

Варианты

Вариант 1

Система должна описывать порядок подготовки к экзамену, предполагающий получение отличной оценки.

Вариант 2

Система должна описывать порядок выполнения практической работы по дисциплине «Проектирование ИС».

Вариант 3

Система должна описывать порядок получения водительских прав.

Вариант 4

Система должна описывать порядок организации городского спортивного соревнования.

Вариант 5

Система должна описывать порядок организации общеинститутского студенческого мероприятия.

Вариант 6

Система составления учебного графика дисциплин, изучаемых на факультете

Вариант 7

Система должна описывать порядок поставок товара в систему розничных киосков.

Вариант 8

Система должна описывать порядок обработки заказов в службе быта.

Вариант 9

Система должна описывать работу одного из участков автосалона.

Вариант 10

Система должна описывать работу приемного покоя в больнице.

Вариант 11

Система должна описывать порядок приема заявки на поставку продукции на хлебокомбинате.

Вариант 12

Система должна описывать процесс поставки сезонных товаров в оптовой фирме.

Вариант 13

Система должна описывать процесс работы торгового отдела.

Вариант 15

Система учета в видеопрокате.

Вариант 16

Система учета проката на лыжной базе

Критерии оценки:

«5»- правильное решение, допущено одно исправление, может обосновать решение

«4» - правильное решение, допущено два-три исправления, может обосновать решение

«3» - правильное решение, не может обосновать решение

«2» - задания не решены, не может обосновать решение

Задание 10

Лабораторная работа Создание диаграммы компонентов

Проверяемые результаты обучения: ПК2.1 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

Создать диаграмму компонентов

Диаграммы компонентов (component diagram) предназначены для распределения классов и объектов по компонентам при физическом проектировании системы. На них изображены компоненты программного обеспечения и связи между ними. При этом на такой диаграмме выделяют два типа компонентов: исполняемые компоненты и библиотеки кода. Часто данный тип диаграмм называют диаграммами модулей.

При проектировании больших систем может оказаться, что система должна быть разложена на несколько сотен или даже тысяч компонентов, и этот тип диаграмм позволяет не потеряться в обилии модулей и их связей.

Каждый класс модели (или подсистема) преобразуется в компонент исходного кода. После создания они сразу добавляются к диаграмме компонентов.

У системы может быть несколько диаграмм, компонентов в зависимости от числа подсистем или исполняемых файлов. Каждая подсистема является пакетом компонентов. В общем случае пакеты - это совокупности компонентов.

Диаграммы компонентов применяются теми участниками проекта, кто отвечает за компиляцию системы. Из нее видно, в каком порядке надо компилировать компоненты, а также какие исполняемые компоненты будут созданы системой. На такой диаграмме показано соответствие классов реализованным компонентам. Она нужна там, где начинается генерация кода.

На рисунке 7.1 показаны компоненты пакета «Form». Они содержат классы пакета «Form» логического представления системы.

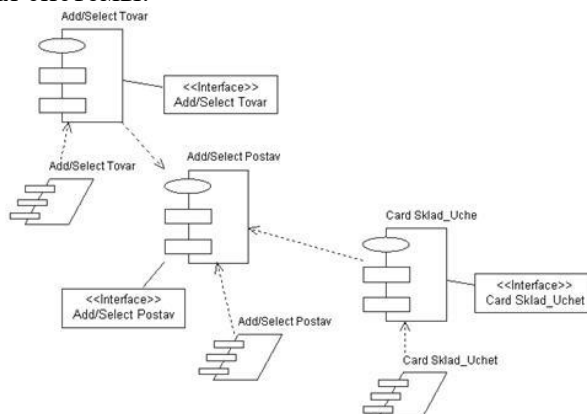


Рисунок 7.1 - Диаграмма компонентов пакета «Form»

На рисунке 7.2 показаны компоненты пакета «DataBase». Они содержат классы пакета «DataBase» логического представления системы.

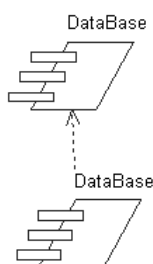


Рисунок 7.2 - Диаграмма компонентов пакета «DataBase»

Вариант 1

Система должна описывать порядок подготовки к экзамену, предполагающий получение отличной оценки.

Вариант 2

Система должна описывать порядок выполнения практической работы по дисциплине «Проектирование ИС».

Вариант 3

Система должна описывать порядок получения водительских прав.

Вариант 4

Система должна описывать порядок организации городского спортивного соревнования.

Вариант 5

Система должна описывать порядок организации общеинститутского студенческого мероприятия.

Вариант 6

Система составления учебного графика дисциплин, изучаемых на факультете

Вариант 7

Система должна описывать порядок поставок товара в систему розничных киосков.

Вариант 8

Система должна описывать порядок обработки заказов в службе быта.

Вариант 9

Система должна описывать работу одного из участков автосалона.

Вариант 10

Система должна описывать работу приемного покоя в больнице.

Вариант 11

Система должна описывать порядок приема заявки на поставку продукции на хлебокомбинате.

Вариант 12

Система должна описывать процесс поставки сезонных товаров в оптовой фирме.

Вариант 13

Система должна описывать процесс работы торгового отдела.

Вариант 15

Система учета в видеопрокате.

Вариант 16

Система учета проката на лыжной базе

Критерии оценки:

«5»- правильное решение, допущено одно исправление, может обосновать решение

«4» - правильное решение, допущено два-три исправления, может обосновать решение

«3» - правильное решение, не может обосновать решение

«2» - задания не решены, не может обосновать решение

Задание 11

Лабораторная работа Построение диаграммы потоков данных

Проверяемые результаты обучения: ПК2.1 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

Диаграммы потоков данных (*DFD*) используются для описания документооборота и обработки информации. Подобно *IDEF0*, *DFD* представляет модельную систему как сеть связанных между собой работ. Их можно использовать как дополнение к модели *IDEF0* для более наглядного отображения текущих операций документооборота в корпоративных системах обработки информации. *DFD* описывает:

- функции обработки информации (работы);
- документы (стрелки, *arrow*), объекты, сотрудников или отделы, которые участвуют в обработке информации;
- внешние ссылки (*external references*), которые обеспечивают интерфейс с внешними объектами, находящимися за границами моделируемой системы;
- таблицы для хранения документов (хранилище данных, *data store*).

В *Вrwin* для построения диаграмм потоков данных используется нотация Гейна-Сарсона.

Для того чтобы дополнить модель *IDEF0* диаграммой *DFD*, нужно в процессе декомпозиции в диалоге *Activity Box Count* кликнуть по радио-кнопке *DFD*. В палитре инструментов на новой диаграмме *DFD* появляются новые кнопки:

- добавить в диаграмму внешнюю ссылку (*External Reference*). Внешняя ссылка является источником или приемником данных извне модели;
- добавить в диаграмму хранилище данных (*Data store*). Хранилище данных позволяет описать данные, которые необходимо сохранить в памяти прежде, чем использовать в работах;

- ссылка на другую страницу. В отличие от *IDEF0* инструмент *offpage reference* позволяет направить стрелку на любую диаграмму (а не только на верхний уровень).

В отличие от стрелок *IDEF0*, которые представляют собой жесткие взаимосвязи, стрелки *DFD* показывают, как объекты (включая данные) двигаются от одной работы к другой. Это представление потоков совместно с хранилищами данных и внешними сущностями делает модели *DFD* более похожими на физические характеристики системы - движение объектов (*data flow*), хранение объектов (*data stores*), поставка и распространение объектов (*external entities*).

DFD рассматривает систему как совокупность предметов. Контекстная диаграмма часто включает работы и внешние ссылки. Работы обычно именуются по названию системы, например "Система обработки информации". Включение внешних ссылок в контекстную диаграмму не отменяет требования методологии четко определить цель, область и единую точку зрения на моделируемую систему.

Работы. В *DFD* работы представляют собой функции системы, преобразующие входы в выходы. Хотя работы изображаются прямоугольниками со скругленными углами, смысл их совпадает со смыслом работ *IDEF0* и *IDEF3*. Так же как работы *IDEF3*, они имеют входы и выходы, но не поддерживают управления и механизмы, как *IDEF0*.

Внешние сущности. Внешние сущности изображают входы в систему и/или выходы из системы. Внешние сущности изображаются в виде прямоугольника с тенью и обычно располагаются по краям диаграммы. Одна внешняя сущность может быть использована многократно на одной или нескольких диаграммах. Обычно такой прием используют, чтобы не рисовать слишком длинных и запутанных стрелок.

Стрелки (Потоки данных). Стрелки описывают движение объектов из одной части системы в другую. Поскольку в *DFD* каждая сторона работы не имеет четкого назначения, стрелки могут подходить к любой грани и выходить из любой грани прямоугольника работы. В *DFD* также применяются двунаправленные стрелки для описания диалогов типа «команда-ответ» между работами, между работой и внешней сущностью и между внешними сущностями.

Хранилище данных. В отличие от стрелок, описывающих объекты в движении, хранилища данных изображают объекты в покое. В материальных системах хранилища данных изображаются там, где объекты ожидают обработки, например в очереди. В системах обработки информации хранилища данных являются механизмом, который позволяет сохранить данные для последующих процессов.

Слияние и разветвление стрелок. В *DFD* стрелки могут сливаться и разветвляться, что позволяет описать декомпозицию стрелок. Каждый новый сегмент сливающейся или разветвляющейся стрелки может иметь собственное имя.

Построение диаграмм *DFD*. Диаграммы *DFD* могут быть построены с использованием традиционного структурного анализа, подобно тому, как строятся диаграммы *IDEF0*. Сначала строится физическая модель, отображающая текущее состояние дел. Затем эта модель преобразуется в логическую модель, которая отображает требования к существующей системе. После этого строится модель, отображающая требования к будущей системе. И, наконец, строится физическая модель, на основе которой должна быть построена новая система.

Альтернативным подходом является подход, популярный при создании программного обеспечения, называемый событийным разделением (*event partitioning*), в котором различные

диаграммы *DFD* выстраивают модель системы. Логическая модель строится как совокупность работ и документирования того, что они (эти работы) должны делать. Затем модель окружения (*environment model*) описывает систему как объект, взаимодействующий с событиями из внешних сущностей. Модель окружения обычно содержит описание цели системы, одну контекстную диаграмму и список событий. Контекстная диаграмма содержит один прямоугольник работы, изображающий систему в целом, и внешние сущности, с которыми система взаимодействует. Наконец, модель поведения (*behavior model*) показывает, как система обрабатывает события. Эта модель состоит из одной диаграммы, в которой каждый прямоугольник изображает каждое событие из модели окружения. Хранилища могут быть добавлены для моделирования данных, которые необходимо запоминать между событиями. Потоки добавляются для связи с другими элементами, и диаграмма проверяется с точки зрения соответствия модели окружения.

Полученные диаграммы могут быть преобразованы с целью более наглядного представления системы, в частности работы на диаграммах могут быть декомпозированы.

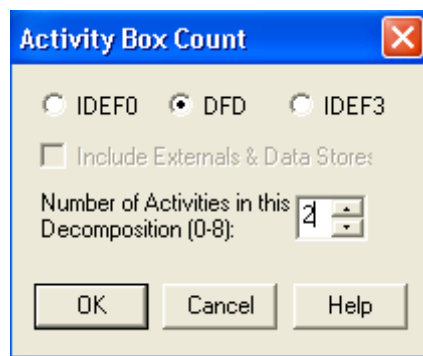
Нумерация объектов. В *DFD* номер каждой работы может включать префикс, номер родительской работы (*A*) и номер объекта. Номер объекта – это уникальный номер работы на диаграмме. Например, работа может иметь номер *A.12.4*. Уникальный номер имеют хранилища данных и внешние сущности независимо от их расположения на диаграмме. Каждое хранилище данных имеет префикс *D* и уникальный номер, например *D5*. Каждая внешняя сущность имеет префикс *E* и уникальный номер, например *E5*.

Порядок выполнения работы

При оформлении заказа важно проверить, существует ли такой клиент в базе данных и, если не существует, внести его в базу данных и затем оформить заказ. Оформление заказа начинается со звонка клиента. В процессе оформления заказа база данных клиентов может просматриваться и редактироваться. Заказ должен включать как информацию о клиенте, так и информацию о заказанных продуктах. Оформление заказа подразумевает чтение и запись информации о прочих заказах.

В процессе декомпозиции согласно правилам *DFD* необходимо преобразовать граничные стрелки во внутренние, начинающиеся и заканчивающиеся на внешних ссылках.

1. Декомпозируйте работу "Оформление заказов" на диаграмме *A2*. В диалоге *Activity Box Count* выберите количество работ 2 и нотацию *DFD* (рис.2.9.1).



2.
Рис. 2.9.1. Выбор нотации *DFD* в диалоге *Activity Box Count*

3. Щелкните по *OK* и внесите в новую диаграмму *DFD A22* имена работ:
 - Проверка и внесение клиента;
 - Внесение заказа.
4. Используя кнопку  на палитре инструментов, внесите хранилища данных:
 - Список клиентов
 - Список продуктов

- Список заказов

5. Удалите граничные стрелки с диаграммы *DFD A22*.
6. Используя кнопку  на палитре инструментов, внесите внешнюю ссылку:
 - Звонки клиентов.
7. Создайте внутренние ссылки согласно рис.2.9.2. При переименовании стрелок используйте словарь.

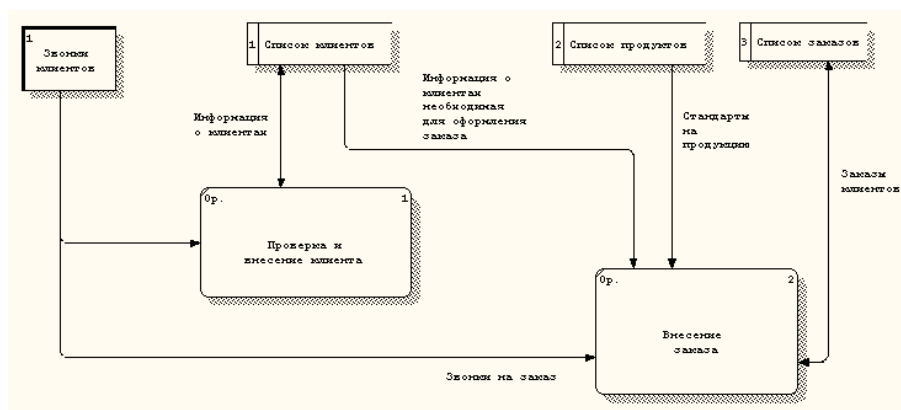


Рис.2.9.2. Диаграмма A22 «Оформление заказов»

8. Обратите внимание, что стрелки «**Информация о клиентах**» и «**Заказы клиентов**» двунаправленные. Для того чтобы сделать стрелку двунаправленной, щелкните правой кнопкой по стрелке, выберите в контекстном меню пункт *Style* и во вкладке *Style* выберите опцию *Bidirectional*.
9. На родительской диаграмме A2 туннелируйте (*Change to Tunnel*) стрелки, подходящие и исходящие из работы «Оформление заказов» (рис.2.9.3).

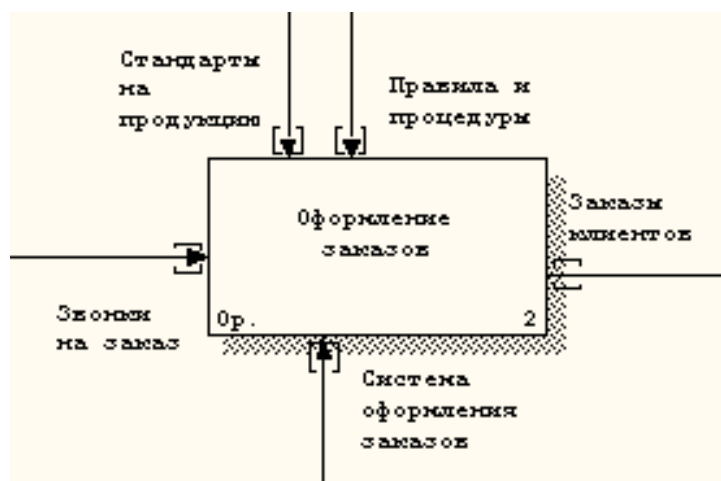


Рис.2.9.3. Работа "Оформление заказов" на диаграмме A2

Некоторые стрелки с диаграмм IDEF0 и DFD (не только с родительских) могут показываться на диаграмме DFD. Для отображения таких стрелок используется инструмент Off-Page Reference.

10. Декомпозируйте работу «**Исследование рынка**» на диаграмме A2 на диаграмму *DFD*. Удалите граничные стрелки. Создайте следующие работы:
- Разработка прогнозов продаж;
 - Разработка маркетинговых материалов;
 - Привлечение новых клиентов.
11. Используя кнопку  на палитре инструментов, внесите хранилища данных:
- Список клиентов;
 - Список продуктов;
 - Список заказов.
12. Добавьте две внешние ссылки:
- Маркетинговые материалы;
 - Прогноз продаж
13. Свяжите объекты диаграммы *DFD* стрелками, как показано на рис.2.9.4.

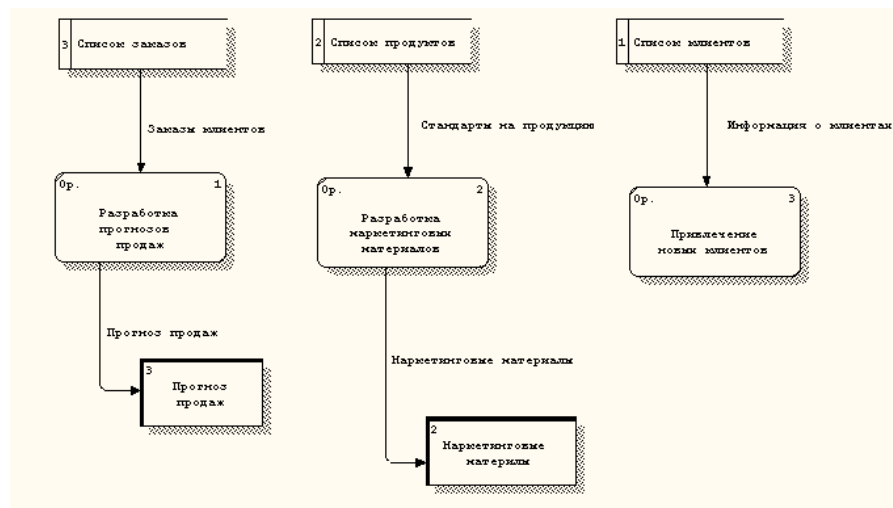


Рис. 2.9.4. Диаграмма A23 “Исследование рынка”

14. На родительской диаграмме A2 туннелируйте (Change to Tunnel) стрелки, подходящие и исходящие из работы «**Исследование рынка**».
15. В случае внесения новых клиентов в работе «**Проверка и внесение клиента**» на диаграмме A22 «**Оформление заказов**» информация должна направляться к работе «**Привлечение новых клиентов**» диаграммы A23 «**Исследование рынка**». Для этого необходимо использовать инструмент *Off-Page Reference*. На диаграмме A22 «**Оформление заказов**» создайте новую граничную стрелку, исходящую от работы «**Проверка и внесение клиента**», и назовите ее «**Информация о новом клиенте**» (рис.2.9.5).

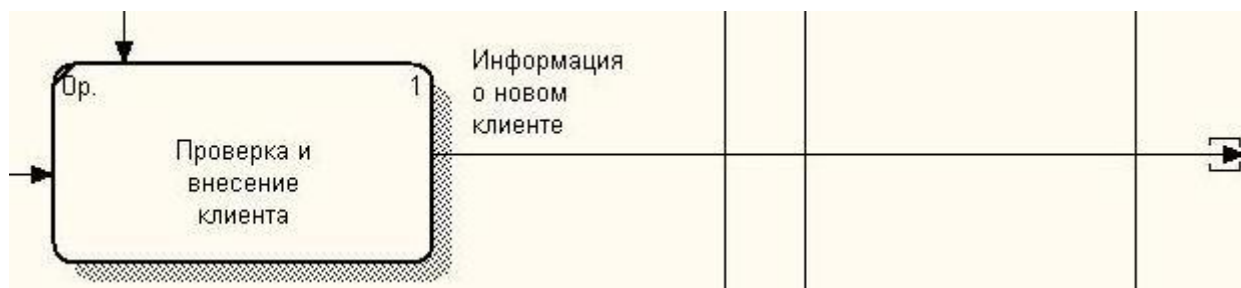


Рис.2.9.5 Стрелка “Информация о новом клиенте” на диаграмме A22

16. Правой кнопкой щелкните по наконечнику стрелки и выберите в меню *Off-Page Reference*. В появившемся диалоге *Off-Page Arrow Reference* (рис.2.9.6) выберите в качестве диаграммы A23D «Исследование рынка». В качестве *Destination border* выберите *Input*. Выберите «Ok and Remain in current diagram».

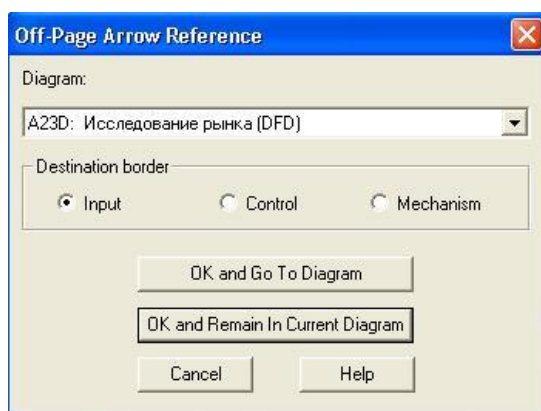


Рис.2.9.6. Диалог *Off-page arrow reference*

17. Перейдите в меню *Model/Model Properties*, далее - во вкладку *Display*. Установите опцию *Off-Page Reference label - Node number*.
18. Перейдите на диаграмму A23D «Исследование рынка» и направьте стрелку «Информация о новом клиенте» на вход работы «Привлечение новых клиентов». Результат представлен на рис.2.9.7.



Рис.2.9.7. Межстраничная ссылка на диаграмме A23

Варианты заданий:

Предметная область «Компания по сборке компьютеров»

Компания занимается в основном сборкой и продажей настольных компьютеров и ноутбуков. Компания не производит компоненты самостоятельно, а только собирает и тестирует компьютеры.

Основные процедуры в компании таковы:

- продавцы принимают заказы клиентов;
- операторы группируют заказы по типам компьютеров;
- операторы собирают и тестируют компьютеры;
- операторы упаковывают компьютеры согласно заказам;
- кладовщик отгружает клиентам заказы.

Компания использует купленную бухгалтерскую информационную систему, которая позволяет оформить заказ, счет и отследить платежи по счетам.

Предметная область «Маршруты городских автобусов»

Фирма, занимающаяся пассажирскими перевозками, имеет в своем распоряжении парк автобусов и штат водителей. Наиболее оптимальным распределением автобусов и водителей по маршрутам занимается диспетчерская. Для этого она собирает информацию об автобусах, водителях и маршрутах.

К основным характеристикам автобуса можно отнести количество мест, комфортабельность, объем двигателя, расход топлива. Некоторые из этих показателей учитываются при распределении автобусов по маршрутам. Например, немногочисленные маршруты могут обслуживаться автобусами с небольшим количеством мест. Так же автобусы должны иметь уникальные номера в учетном списке (это могут быть номерные знаки).

Информация о водителях имеется в отделе кадров. Диспетчерской необходима информация о возрасте, стаже водителя, а так же его фамилия, имя, отчество. Например, водитель с высоким стажем будет отправлен на более многочисленный и сложный маршрут.

Информация о маршрутах включает в себя номера маршрутов, их протяженность в км, популярность (число пассажиров в час), время обслуживания. Например, протяженность и время обслуживания плюс информация о расходе топлива автобуса нужны, чтобы рассчитывать количество топлива на маршрут. Можно так же, когда уже конкретно распределены автобусы и водители по маршрутам, учитывать информацию о дневной выручке на маршруте.

Предметная область «Комплектующие компьютера»

С каждым днем появляются все новые и новые модели комплектующих на компьютер. Улучшаются показатели, функции уже существующих. Появилось множество фирм, сервисных центров, торгующих компьютерами и комплектующими. Многие из них комплектуют компьютер по желанию клиента. В этом случае как раз пригодится систематизированный список комплектующих (прайс-лист). Он должен содержать все необходимые данные: вид, наименование, модель, параметры, цену (например, вид: материнская плата, наименование: MB, модель: VX Pro+). Обязательно должна быть указана совместимость/несовместимость одного типа комплектующего с другими. Очень часто из-за несовместимости происходят поломки, сбои, устройства сгорают, приходят в негодность.

Каждая единица устройства имеет свой серийный номер, однозначно его идентифицирующий. Этот номер помимо учета устройств, необходим при предоставлении гарантийных талонов клиентам. Фирма по гарантийному талону будет обслуживать лишь то устройство, номер которого указан в талоне.

Предметная область «Оборудование, собираемое мебельной фабрикой «Лада»

«Лада» производит следующие типы оборудования: аптечное, торговое и офисное. В каждом типе оборудования имеются свои категории оборудования (столы, шкафы, стеллажи и т.д.). Единица оборудования имеет название, габариты, цвет, материал, из которого собрана, цену. На фабрике каждому виду и каждой категории оборудования присваиваются шифры (шифр

вида и шифр категории). Эти шифры нужны для однозначной идентификации оборудования (например, шифр ША2-шкаф аптечный с № 2 по списку).

Произведенная мебель реализуется отделом сбыта. Выше перечисленная информация, характеризующая оборудование, может быть занесена в прайс - листы и предоставляться покупателям.

Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

Тема 2.1.3. Оценка качества программных средств

Проверяемые результаты обучения: ПК2.1 ПК2.5

Вид учебной работы: аудиторная

Задание. Ответить на устные вопросы:

1. Цели и задачи и виды тестирования. Стандарты качества программной документации. Меры и метрики.
2. Тестовое покрытие.
3. Тестовый сценарий, тестовый пакет.
4. Анализ спецификаций.
5. Верификация программного обеспечения.
6. Аттестация программного обеспечения.

Критерии оценки: Полнота ответа на поставленные вопросы

- «5»- Правильный уверенный ответ
- «4» - Уверенный ответ, допущена одна неточность
- «3» - Ответ близкий к правильному, с несколькими неточностями
- «2» - Неправильный ответ, отказывается отвечать

Задание 2

Лабораторная работа Разработка тестового сценария и Оценка необходимого количества тестов

Проверяемые результаты обучения: ПК2.1 ПК2.2 ПК2.3 ПК2.4 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

1. Разработать тестовый сценарий
2. Определить количество тестов
3. Провести тестирование ранее созданной программы

Характеристики хорошего теста:

- Существует обоснованная вероятность выявления тестом ошибки
- Набор тестов не должен быть избыточным
- Тест должен быть наилучшим в своей категории
- Он не должен быть слишком простым или слишком сложным

Методология разработки тестовых случаев на основе сценариев использования. Случай использования состоит из некоторого множества сценариев: нормальный случай, расширения и исключительные ситуации. Для разработки тест кейсов на основе одного случая использования разрабатываются несколько сценариев, соответственно: Сценарий использования представляет собой оптимистический сценарий, который выбирается чаще других. В раздел Альтернативные пути могут быть включены несколько сценариев, которые отличаются от сценария использования в различных аспектах, однако остаются полноценными

путями исполнения. В раздел Исключительные пути попадают те сценарии, которые приводят к возникновению ошибок. Каждый сценарий предусматривает действия, предпринимаемые действующим субъектом, и требует от системы отклика, который соответствует основной части тестового случая. Тестовый случай состоит из некоторого набора предусловий, стимулирующего воздействия (входные данные) и ожидаемого отклика.

При разработке нужно определить, сколько необходимо использовать тестовых случаев из каждого случая использования, после чего построить эти случаи. Первым шагом на пути определения количества тестовых случаев, приходящихся на один случай и Профиль использования системы — это упорядочивание индивидуальных случаев использования, в основу которого положено некоторое сочетание значений частоты использования и критичности для отдельных случаев использования., является построение профилей использования.

Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

Задание 3

Лабораторная работа Разработка тестовых пакетов

Проверяемые результаты обучения: ПК2.1 ПК2.2 ПК2.3 ПК2.4 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

Составить набор тестов для проверки программы, с помощью которой решается предлагаемая задача.

Задача: написать программу, в которой с клавиатуры вводятся три целых положительных числа, которые являются сторонами треугольника. На экран выводится сообщение, является ли треугольник неравносторонним, равнобедренным, равносторонним. (Треугольник должен быть правильным, т.е. сумма длин любых двух сторон должна быть больше длины третьей стороны.)

Программа должна быть составлена с применением пользовательских процедур и функций.

Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

Задание 4

Лабораторная работа Оценка программных средств с помощью метрик

Проверяемые результаты обучения: ПК2.1 ПК2.2 ПК2.3 ПК2.4 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

1.Используя Холстеда выполнить оценку разработанной программы

Основные измеряемые параметры:

- 1) Число уникальных операторов η_1
- 2) Число уникальных операндов η_2
- 3) Общее число операторов N_1
- 4) Общее число операндов N_2

Дополнительные параметры:1) Частота вхождений i -го оператора f_{1i} 2) Частота вхождений j -го операнда f_{2j} Словарь программы: $\eta = \eta_1 + \eta_2$ Опытная длина реализации: $N = N_1 + N_2$ **2. Заполнить таблицу**

Оператор	Номер (i)	f_{1i}	Операнд	Номер (j)	f_{2j}

function GCD(a,b:integer):integer;

Label L1;

Var G,R:integer;

Begin

if (a=0) then begin

GCD:=b; return;

end;

if (b=0) then begin

GCD:=a; return;

end;

L1:

G:=a/b;

R:=a-b*G;

if (R=0) then begin

GCD:=b; return;

end;

a:=b; b:=R; GOTO L1;

end;

Оператор	Номер (i)	f_{1i}	Операнд	Номер (j)	f_{2j}
:=	1	7	GCD	1	3
=	2	3	G	2	2
			R	3	3
if then	3	3	a	4	5
/	4	1	b	5	7
*	5	1	0	6	3
-	6	1			
() begin end	7	8			
goto	8	1			
function GCD	9	1			
return	10	3			
;	11	14			

 $\eta_1 = 11, \eta_2 = 6, \eta = 17$ $N_1 = 43, N_2 = 23, N = 66$

Критерии оценки:

«5»- правильное решение, допущено одно исправление, может обосновать решение

«4» - правильное решение, допущено два-три исправления, может обосновать решение

«3» - правильное решение, не может обосновать решение

«2» - задания не решены, не может обосновать решение

Задание 5

Лабораторная работа Инспекция программного кода на предмет соответствия стандартам кодирования

Проверяемые результаты обучения: ПК2.1 ПК2.2 ПК2.3 ПК2.4 ПК2.5

Вид учебной работы: аудиторная работа

Задание

1.Переписать программный код (варианты программ согласно варианту), используя общепринятые соглашения и рекомендации по именованию и форматированию переменных, операторов, выражений.

2.Ответить на вопросы письменно:

- Для чего нужны стандарты кодирования.
- Какие существуют соглашения по именованию переменных, функций.
- В каком формате следует записывать константы в тексте программы.
- Какие стили расстановки скобок существуют? В чем их различия?

Требования:

- добавить комментарии в программный код;
- проверить правильность именования переменных, констант, методов;
- проверить правильность объявления переменных и констант;
- записать исправленный текст программы в двух стилях (согласно таблице, заданной выше).

Стандарты и руководства по кодированию

Различные модули, указанные в проектной документации, кодируются на этапе кодирования в соответствии со спецификацией модуля. Основной целью этапа кодирования является кодирование из проектной документации, подготовленной после этапа проектирования, на языке высокого уровня, а затем модульное тестирование этого кода.

Хорошие организации по разработке программного обеспечения хотят, чтобы их программисты придерживались определенного и стандартного стиля кодирования, называемого стандартами кодирования. Обычно они создают свои собственные стандарты и руководства по кодированию в зависимости от того, что лучше всего подходит для их организации, и на основе типов программного обеспечения, которое они разрабатывают. Для программистов очень важно поддерживать стандарты кодирования, в противном случае код будет отклонен во время проверки кода.

Цель наличия стандартов кодирования:

- Стандарт кодирования дает единообразный вид кодам, написанным разными инженерами.
- Это улучшает удобочитаемость и удобство сопровождения кода, а также снижает сложность.
- Это помогает в повторном использовании кода и помогает легко обнаружить ошибку.
- Это продвигает здоровые методы программирования и увеличивает эффективность программистов.

Некоторые из стандартов кодирования приведены ниже:

1. **Ограниченное использование глобалов:**

Эти правила говорят о том, какие типы данных могут быть объявлены глобальными, а какие нет.

2. **Стандартные заголовки для разных модулей:**

3. Для лучшего понимания и обслуживания кода заголовков различных модулей должен соответствовать стандартному формату и информации. Формат заголовка должен содержать ниже вещи, которые используются в различных компаниях:

- Наименование модуля
- Дата создания модуля
- Автор модуля
- История изменений
- Краткое описание модуля о том, что делает модуль
- В модуле поддерживаются различные функции, а также входные и выходные параметры
- Глобальные переменные, доступные или измененные модулем

4. **Соглашения об именах для локальных переменных, глобальных переменных, констант и функций:**

Некоторые из соглашений об именах приведены ниже:

- Содержательное и понятное название переменных помогает любому понять причину его использования.
- Локальные переменные должны быть названы с использованием букв верблюда, начинающихся с маленькой буквы (например, **localData**), тогда как имена глобальных переменных должны начинаться с заглавной буквы (например, **GlobalData**). Имена констант должны быть сформированы только заглавными буквами (например, **CONSDATA**).
- Лучше избегать использования цифр в именах переменных.
- Названия функции должны быть написаны в верблюжьем регистре, начиная с маленьких букв.
- Название функции должно четко и кратко описывать причину ее использования.

5. **Отступ:**

Правильный отступ очень важен для улучшения читабельности кода. Чтобы сделать код читабельным, программисты должны правильно использовать пробелы. Некоторые из интервалов даны ниже:

- После запятой между аргументами функции должен быть пробел.
- Каждый вложенный блок должен иметь правильные отступы и интервалы.
- Надлежащий отступ должен быть в начале и в конце каждого кадра в программе.
- Все фигурные скобки должны начинаться с новой строки, а код, следующий за окончанием фигурных скобок, также начинается с новой строки.

6. **Возвращаемые значения ошибок и соглашения об обработке исключений:**

Все функции, которые сталкиваются с ошибкой, должны возвращать 0 или 1 для упрощения отладки.

С другой стороны, руководящие принципы кодирования дают некоторые общие рекомендации относительно стиля кодирования, которому необходимо следовать для улучшения понятности и читабельности кода. Некоторые из руководств по кодированию приведены ниже:

7. **Избегайте использования стиля кодирования, который слишком сложен для понимания:**

Код должен быть легко понятным. Сложный код делает обслуживание и отладку трудной и дорогой.

8. **Избегайте использования идентификатора для нескольких целей:**

Каждой переменной должно быть дано описательное и осмысленное имя, указывающее причину ее использования. Это невозможно, если идентификатор используется для нескольких целей, что может привести к путанице у читателя. Кроме того, это приводит к более трудностям во время будущих улучшений.

9. **Код должен быть хорошо документирован:**

Код должен быть правильно прокомментирован для понимания. Комментарии относительно утверждений повышают понятность кода.

10. **Длина функций не должна быть очень большой:**

Длинные функции очень трудно понять. Вот почему функции должны быть достаточно маленькими, чтобы выполнять небольшую работу, а длинные функции должны быть разбиты на маленькие для выполнения небольших задач.

11. **Старайтесь не использовать оператор GOTO:**

Оператор GOTO делает программу неструктурированной, что снижает ее понятность и затрудняет отладку.

Критерии оценки:

«5»- правильное решение, допущено одно исправление, может обосновать решение

«4» - правильное решение, допущено два-три исправления, может обосновать решение

«3» - правильное решение, не может обосновать решение

«2» - задания не решены, не может обосновать решение

Раздел 2. Средства разработки программного обеспечения

МДК.2.2 Инструментальные средства разработки программного обеспечения:

Тема 2.2.1 Современные технологии и инструменты интеграции.

Задание 1

Проверяемые результаты обучения: ПК2.1 ПК2.5

Вид учебной работы: аудиторная

Задание. Ответить на устные вопросы:

1. Понятие репозитория проекта, структура проекта.
2. Виды, цели и уровни интеграции программных модулей. Автоматизация бизнес-процессов
3. Выбор источников и приемников данных, сопоставление объектов данных.
4. Транспортные протоколы. Стандарты форматирования сообщений
5. Организация работы команды в системе контроля версий.

Критерии оценки: Полнота ответа на поставленные вопросы

«5»- Правильный уверенный ответ

«4» - Уверенный ответ, допущена одна неточность

«3» - Ответ близкий к правильному, с несколькими неточностями

«2» - Неправильный ответ, отказывается отвечать

Задание 2

Лабораторная работа «Разработка структуры проекта»

Проверяемые результаты обучения: ПК2.1 ПК2.2 ПК2.3 ПК2.4 ПК2.5

Вид учебной работы: аудиторная работа (Лабораторная работа «Разработка структуры проекта»)

Задание:

1. New Project: Создание проекта.

1. Пункт меню *File -> New -> Project* (Файл -> Создать -> Проект).
2. В открывшемся окне в разделе *Project Types* (Установленные) найти *Visual C++*.
3. В разделе *Templates* (Шаблоны) выбрать *Win32 Console Application* (Консольное приложение Win32).
4. Указать имя нового «решения» (*Solution Name*) и имя проекта (*Name*).
5. Сохранить проект на локальном диске в легко доступном месте (например, *D:\Projects*).
6. Можно снять галочку *Create folder for solution* (Создать каталог для решения), чтобы не множить каталоги без необходимости.
7. В следующем диалоговом окне щелкнуть *Далее* и затем выбрать *Application Settings* (Дополнительные параметры) -> *Empty Project* (Пустой проект). Убрать галочку с пункта Проверки жизненного цикла...

2. Add files: Добавление файлов.

Правый клик на проекте в окне *Solution Explorer* (Обозреватель решений) — пункт *Add* (Добавить) — пункт *Create New Item* (Создать элемент) или *Add existing Item* (Существующий элемент). Для этого и любого другого проекта понадобится как минимум один файл с точкой входа в программу — функцией *main*.

3. Exclude files: Исключение файлов.

Сначала удобно создать один проект и добавлять в него по одному файлу с решением каждой отдельной задачи. Однако при этом возникает проблема: в нескольких файлах проекта присутствует функция *main*, что противоречит правилам C++. Для решения данной проблемы, необходимо «исключать» файлы из проекта (правый клик на файле, *Exclude from project* (Исключить)). При этом файлы исчезнут из окна проекта, однако сохранятся в каталоге проекта на диске.

1. Создайте новый проект с названием *MyFirstProject* (алгоритм создания проекта см. выше).
2. В окне *Обозреватель решений* выполните правый клик на папке *Файлы исходного кода* и выберите пункт *Добавить -> Создать элемент -> Файл C++ (cpp)*. В поле *Имя файла* введите *main.cpp*. Щелкните *Добавить*
3. В первой строке файла добавим директиву *include* для подключения внешних файлов-библиотек:
4. `#include <iostream>`
5. Далее подключим глобальные пространства имен:
`using std::cout;`
`using std::cin;`
`using std::endl;`
6. Добавьте главную функцию `main () {}` и вставьте в нее инструкцию вывода в консоль слов «Привет мир!»
`int main () {`
`cout<< "Hello World!";`

операторы вывода:
`cout << "Hello World!";`

или вывод значения переменной:
`cout << n;`
7. чтобы консольное окно не исчезало сразу, добавим паузу:
`int main () {`
`cout<< "Hello World!";`
`system("pause");`
`}`

Часть 2:

Выполним то же самое, используя функцию.

8. Перейдите в окно *Обозреватель решений* и выполните правый клик на папке *Заголовочные файлы*. Выберите пункт *Добавить -> Создать элемент -> Файл Заголовка*. В поле *Имя* введите *hello.h*.

9. Перейдите во вкладку с файлом *main.cpp* и подключите к файлу созданный заголовочный файл:

```
#include "hello.h"
#include <iostream>
```

10. Перейдите во вкладку с файлом *hello.h* и создайте там функцию с названием *SayHello()*. Добавьте объявление функции:

```
void SayHello(); // объявление функции
```

```
void SayHello(){ // определение функции
    cout<< "Hello World!";
}
```

Ключевое слово *void* означает, что функция не возвращает никакого значения. Следовательно, такая функция может быть вызвана оператором вызова функции *SayHello()*. Т.е. просто по имени функции.

11. Добавьте в заголовочный файл директиву *<iostream>* и пространства имен *std*, удалив его из файла *main.cpp*

```
#include <iostream>
using std::cout;
using std::cin;
using std::endl;
```

```
void SayHello(); // объявление функции
```

```
void SayHello() { // определение функции
    cout<< "Hello World!";
}
```

12. Перейдите во вкладку с файлом *main.cpp* и удалите из функции *main()* вывод слов. Добавьте в нее вызов функции *SayHello()*:

```
...
int main () {
    SayHello();
    system("pause");
    return 0;
}
```

Поскольку перед названием функции стоит *int*, что означает, что функция должна вернуть целочисленное значение, то необходим оператор возврата результата выполнения функции, который располагается в самом конце функции:

```
return 0; // оператор возврата результата выполнения функции
```

13. Откомпилируйте и запустите проект. Посмотрите результат.

При работе в программе *codeBlocks* необходимо самостоятельно создать файл *hello.h* в папке с проектом. Все остальные действия делаются по аналогии

- Для ввода данных применяется стандартный потоковый объект *cin*, который использует операцию `>>`. Справа от этого знака находится переменная, принимающая вводимую информацию.
- Для вывода данных применяется стандартный потоковый объект *cout*, который использует операцию `<<`. Справа от этого знака находится литерал или переменная, значение которой выводится.
- Использование объектов *cin* и *cout* возможно, потому что в программе подключен файл *iostream*

Критерии оценки:

- «5» – правильное решение, допущено одно исправление, может обосновать решение
- «4» – правильное решение, допущено два-три исправления, может обосновать решение
- «3» – правильное решение, не может обосновать решение
- «2» – задания не решены, не может обосновать решение

Задание 3

Лабораторная работа «Разработка модульной структуры проекта (диаграммы модулей)»

Проверяемые результаты обучения: ПК2.1 ПК2.2 ПК2.3 ПК2.4 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

Составить модульную структуру ранее созданной программы

ВИДЫ ПРЕДМЕТНЫХ ОБЛАСТЕЙ

1. Страховая компания
2. Гостиница
3. Ломбард
4. Реализация готовой продукции
5. Ведение заказов
6. Бюро по трудоустройству
7. Нотариальная контора
8. Фирма по продаже запчастей
9. Курсы повышения квалификации
10. Туристическая фирма
11. Грузовые перевозки
12. Ювелирная мастерская
13. Парикмахерская
14. Химчистка
15. Сдача в аренду торговых площадей

На этапе проектирования необходимо выполнить проектирование модульной структуры программы. Модульная структура представляет собой иерархию процедур и функций (называемых модулями), с помощью которых программа решает поставленную задачу. При этом программа является головным модулем в данной иерархии.

Размер модуля измеряется числом содержащихся в нем операторов или строк. Модуль не должен быть слишком маленьким или слишком большим. Маленькие модули приводят к громоздкой модульной структуре программы и могут не окупать накладных расходов, связанных с их оформлением. Большие модули неудобны для изучения изменений, они могут существенно увеличить суммарное время повторных трансляций программы при отладке программы. Обычно рекомендуются программные модули размером от нескольких десятков до нескольких сотен операторов.

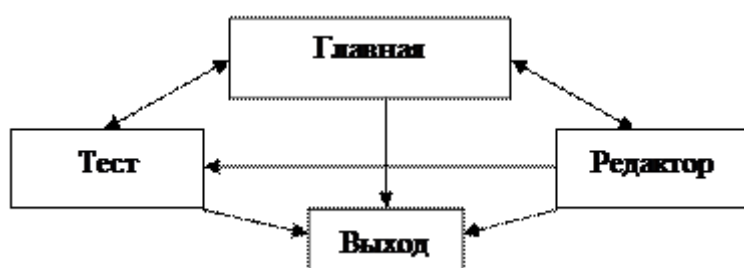
В качестве модульной структуры программы принято использовать древовидную структуру, включая деревья со сросшимися ветвями. В узлах такого дерева размещаются программные модули, а направленные дуги (стрелки) показывают статическую подчиненность модулей, т.е. каждая дуга показывает, что в тексте модуля, из которого она исходит, имеется ссылка на модуль, в который она входит. Другими словами, каждый модуль может обращаться к подчиненным ему модулям, т.е. выражается через эти модули. При этом модульная структура программы, в конечном счете, должна включать и совокупность спецификаций модулей, образующих эту программу.

Спецификация программного модуля содержит:

- синтаксическую спецификацию его входов, позволяющую построить на используемом языке программирования синтаксически правильное обращение к нему (к любому его входу);
- функциональную спецификацию модуля (описание семантики функций, выполняемых этим модулем по каждому из его входов).

В процессе разработки программы ее модульная структура может по-разному формироваться и использоваться для определения порядка программирования и отладки модулей, указанных в этой структуре.

Модульная структура программы представлена на (схема 1)



Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

Задание 4

Лабораторная работа «Разработка перечня артефактов и протоколов проекта»

Проверяемые результаты обучения: ПК2.1 ПК2.2 ПК2.3 ПК2.4 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

Для ранее разработанного проекта составить перечень артефактов

Артефакты — это некоторые продукты проекта, порождаемые или используемые в нем при работе над окончательным продуктом.

Ниже приведен набор артефактов (моделей, документов и т.п.) для каждого из потоков.

Business modeling

Артефакты-модели — используется Rational Rose:

- модель бизнес-процессов — определение бизнес-требований к разрабатываемой системе;
- модель структуры предприятия — артефакт для разработки функциональной модели системы;
- модели документов, бизнес-сущностей, модели сценариев бизнес-функций, модели состояний бизнес-сущностей — для проектирования пользовательского интерфейса, БД системы; представляют собой описание статического и динамического состояний системы с различных точек зрения;

- модели бизнес-правил — артефакт используется для моделирования правил в ПО.
- Артефакты-документы — используются RequisitePro, SoDA, текстовые процессоры,

Microsoft Project:

- оценка организации заказчика, структура бизнеса;
- словарь терминов предметной области;
- набор бизнес-правил;
- коммерческое предложение;
- спецификации бизнес-функций;
- план работ на этапе бизнес-моделирования;
- рекомендации по проведению бизнес-моделирования;
- запросы на изменение.

Requirements

Артефакты-модели — используется Rational Rose:

- модель функции системы;
- модель сценариев функций системы;
- модель интерфейсов пользователя;
- модель сценариев работы пользователя системы;
- модель выходных форм;
- модель правил системы.

Артефакты-документы — используются RequisitePro, SoDA, текстовые процессоры, MS Project:

- план управления требованиями;
- словарь терминов системы;
- спецификация на программную систему;
- спецификация на функции системы;
- правила системы;
- запросы заинтересованных лиц;
- план работ на этапе определения требований к системе;
- рекомендации по моделированию на этапе определения требований;
- запросы на изменение.

Analysis and design

Артефакты-модели — используется Rational Rose:

- логическая модель данных;
- физическая модель данных;
- модель спецификаций компонентов системы;
- сценарии взаимодействия классов, реализующих компоненты системы.

Артефакты-документы — используются RequisitePro, SoDA, текстовые процессоры, MS Project:

- архитектура программного обеспечения;
- спецификации программных компонентов;
- рекомендации на этапе анализа и проектирования;
- план работ на этапе анализа и проектирования;
- запросы на изменение.

Implementation

Артефакты-модели — используется Rational Rose:

- компонентная модель приложения.

Артефакты-код — используются Rational Rose, средства программирования, текстовые процессоры:

- элементы генерации кода, полученные в Rational Rose;
- собственно код приложения;
- документация.

Артефакты-документы — используются RequisitePro, SoDA, текстовые процессоры, MS Project:

- план сборки приложения;
- план работ на этапе реализации.

Test

Артефакты-модели — используется Rational Rose:

- модель тестовых примеров;
- функциональная модель тестовой программы;
- модель спецификации компонентов тестовой программы;
- сценарии взаимодействия классов, реализующих взаимодействие компонентов тестовой программы.

Артефакты-документы — используются SoDA, текстовые процессоры, MS Project:

- описание тестовых примеров;
- план тестирования;
- план работ на этапе тестирования;
- запросы на изменение.

Реализация тестирования — Quantify, Purify, PureCoverage, Robot, SiteLoad, SiteCheck.

Deployment

Артефакты-модели — используется Rational Rose:

- модель размещения — описание размещения компонентов по узлам обработки.

Артефакты-документы — используются SoDA, текстовые процессоры, MS Project:

- обучающие материалы;
- документы по инсталляции;
- описание версий системы;
- план внедрения.

Критерии оценки:

«5»- правильное решение, допущено одно исправление, может обосновать решение

«4» - правильное решение, допущено два-три исправления, может обосновать решение

«3» - правильное решение, не может обосновать решение

«2» - задания не решены, не может обосновать решение

Задание 5

Лабораторная работа «Настройка работы системы контроля версий (типов импортируемых файлов, путей, фильтров и др. параметров импорта в репозиторий)»

Проверяемые результаты обучения: ПК2.1 ПК2.2 ПК2.3 ПК2.4 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

Произвести настройку работы системы контроля версий.

Разработка — это почти всегда командная игра. Пора учиться работать в команде. Даже если пока что в твоей команде только монитор, системник (или старенький ноутбук) и острое желание стать программистом, всё равно пора учиться.

Программисту проще стать своим среди своих, ведь за него говорят не дежурные улыбки, перекуры и чаепития с печеньками, а чёткие строки кода, элегантные функции и безупречная работа готовых приложений.

Чтобы эффективно работать в команде, мало знать синтаксис языка, ключевые библиотеки и уметь обращаться с базами данных. Необходимо уметь работать в удобной для команды системе контроля версий.

О системах контроля версий их преимуществах и недостатках можно почитать [здесь](#). В этой статье мы перейдём от теории к практике и расскажем, как работать с git'ом.

Шаг 1. Выбираем git-хостинг

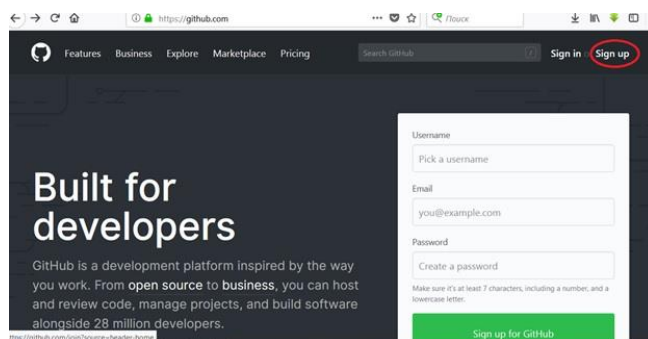
Git-хостинг на разных условиях предлагают десятки компаний.

Самые известные из них: [Github](#), [Sourceforge](#), [Google Code](#), [GitLab](#), [Codebase](#). Выбери удобный интерфейс и регистрируйся на понравившемся хостинге.

В этой статье мы рассмотрим работу с git-хостингом на примере [Github](#)'а.

Шаг 2. Регистрация

Процедура регистрации на Гитхабе простая и интуитивно понятная даже для тех, чей уровень английского далёк от Upper Intermediate.

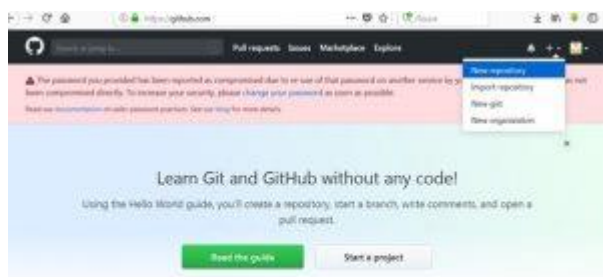


Логин, пароль, почта → подтверждение, и связь с мировым сообществом программистов налажена.

Шаг 3. Создание репозитория

Ты можешь создать любое количество репозиториев, у каждого из которых будет issue tracking, wiki, условия для проведения code review и многое другое. Политика разработчиков Github предполагает бесплатное использование хостинга для всех open-source проектов.

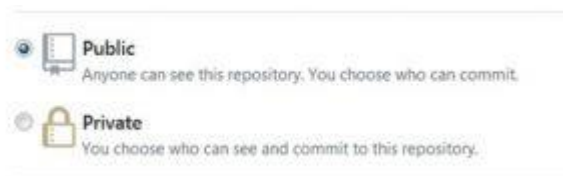
Чтобы создать новый репозиторий нажмём кнопку + в верхней части экрана и выберем **New repository**



Создание репозитория на Гитхабе

Многие разработчики рано или поздно сталкиваются с необходимостью создания приватного репозитория, код из которого доступен только их команде. Для этих случаев на Github'е есть определённый тарифный план.

Но пока острой необходимости в создании приватного репозитория у нас нет, создадим обычный.



Жмём волшебную кнопку **Create** внизу экрана, и репозиторий готов.

Шаг 4. Работа с репозиторием

Работа с репозиторием может вестись из командной строки, напрямую из среды разработки или из графического интерфейса (git — клиент приложения).

Работа с графическим интерфейсом позволяет лучше понимать процессы, происходящие в локальном и удалённом репозитории. Поэтому я рекомендую начать работу с git с использованием графического интерфейса.

Шаг 5. Выбираем Гит-клиент

Потом, когда суть процессов изменения и обновления (восстановления) информации в репозитории станет для Вас очевидна, можно работать и через командную строку. В этом принципе работы есть немало своих преимуществ. Например, все новые опции Гитхаба реализуются сначала для использования в командной строке, и только потом адаптируются под графические интерфейсы.

Но вернёмся к git-клиентам.

Самыми популярными гит- клиентами на данный момент являются:

SmartGit

Удобное приложение гармонично сочетает все необходимые функции и доступную интуитивно понятную систему управления. SmartGit – один из самых удобных графических интерфейсов для профессиональной разработки. Некоммерческая разработка и разработка open-source проектов не требуют платной лицензии.

GitHub Desktop

«Родной» графический интерфейс Гитхаба. GitHub Desktop работает под Windows и Mac и практически полностью копирует функционал основного сайта. Работает под той же учётной записью.

Правда, не всегда оперативно справляется с большими программами.

Зато отлично подходит для начала работы с git.

GitKraken

Поддерживает GitHub, Bitbucket и Gitlab.

Кракен очень любят программисты — фрилансеры, которым периодически приходится менять команды, а значит, и условия командной разработки. Возможность работы с разными git-хостингами через привычное приложение со знакомым интерфейсом в таких случаях играет важную роль.

SourceTree

SourceTree позволяет работать с Bitbucket и GitHub. В приложении довольно простой интерфейс, подходящий, как для опытных программистов, так и для новичков.

Шаг 6. Работа со SmartGit

В этой статье мы рассмотрим работу с SmartGit.

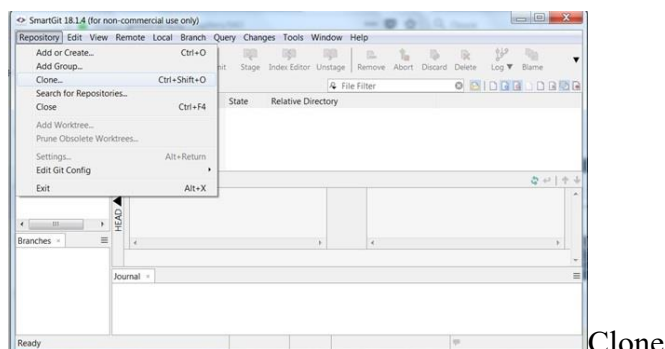
Скачать SmartGit можно, например, [отсюда](#):
Устанавливаем и запускаем.

Основные операции для работы с git

Clone

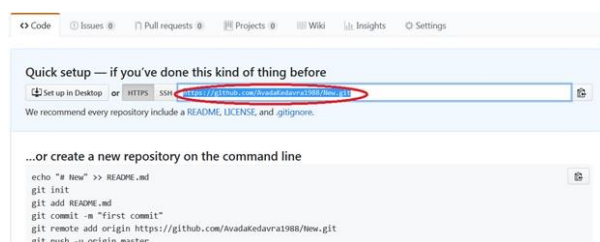
Первое, чему стоит научиться — это снимать копию проекта из удалённого репозитория в локальный.

Делается это довольно просто:



Clone

Затем копируем ссылку репозитория, созданного на Гитхабе (шаг 2)



Ссылка на репозиторий

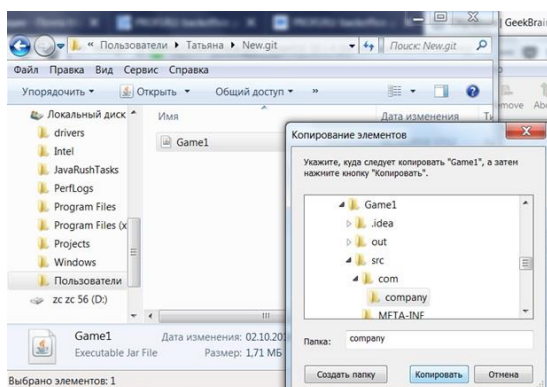
Вставляем адрес удалённого репозитория в нужную ячейку в открывшемся окне, выбираем расположение нового локального репозитория у нас на компьютере, и получаем готовый локальный репозиторий.

К слову, аналогичным образом можно клонировать чужой открытый репозиторий и поближе познакомиться с чужим кодом.

Commit

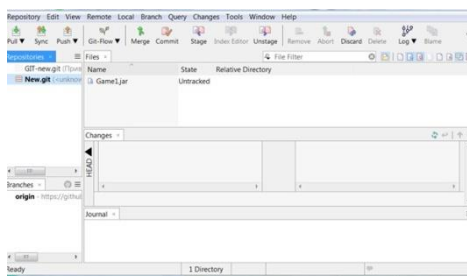
Репозиторий готов – пора приступать к работе.

Написанный код мы помещаем в локальный репозиторий — папку `.git` (путь к которой мы указали в операции `clone`).



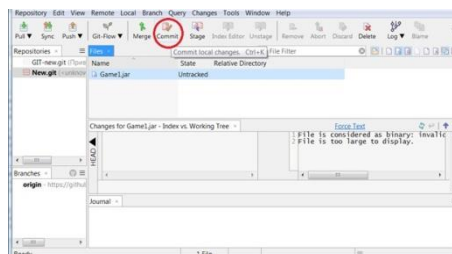
Добавление файла в локальный репозиторий

Если всё прошло успешно, в окошке SmartGit'a появится скопированный файл.



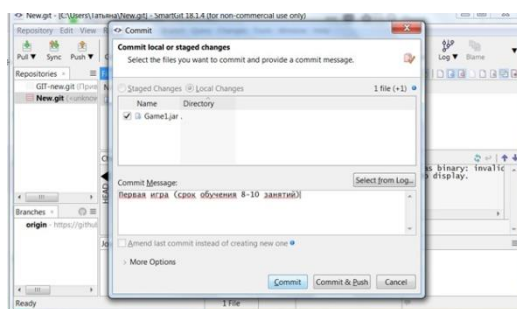
Новый файл в SmartGit

Для того чтобы зафиксировать изменения в локальном репозитории, нажимаем кнопку **Commit**.



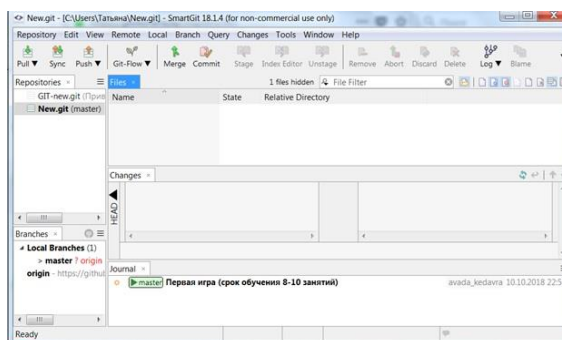
Commit

В открывшемся окне пишем пояснительный комментарий к сохраняемому файлу и снова нажимаем кнопку **Commit**



Пояснения к Commit'у

Файл сохранён, а изменения внесены в журнал.

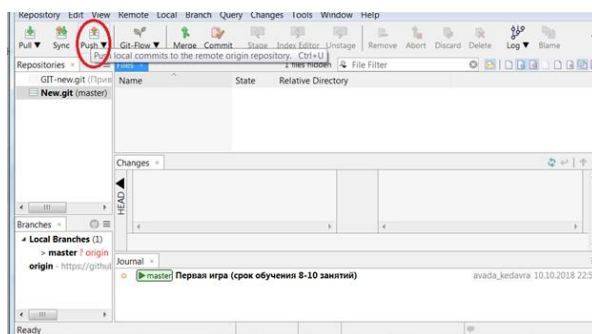


Файл отправлен в локальный репозиторий

Push

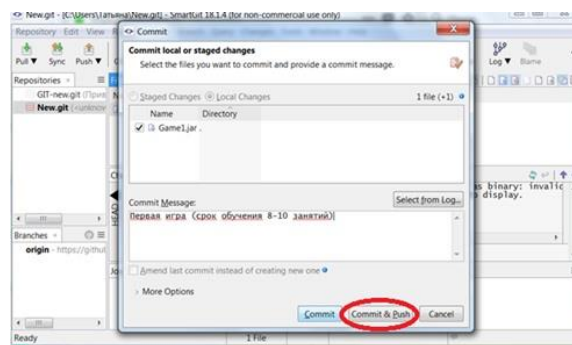
Теперь заглянем на [Github.com](https://github.com) в наш удалённый репозиторий. Там до сих пор нет ни одного файла. Нужно срочно менять ситуацию.

Чтобы перенести изменения, внесённые в локальный репозиторий, в удалённый репозиторий, необходимо нажать кнопку **Push**.



Push

К слову, отправить изменения в удалённый репозиторий, нам предлагают ещё в точке **Commit'а**

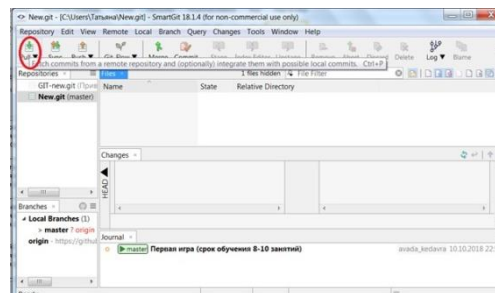


Commit & Push

Pull

Возникает резонный вопрос: как получают изменения остальные участники разработки, если они клонировали проект в самом начале?

Для этого существует команда **Pull**, передающая в локальный репозиторий все изменения, происходящие в удалённом.

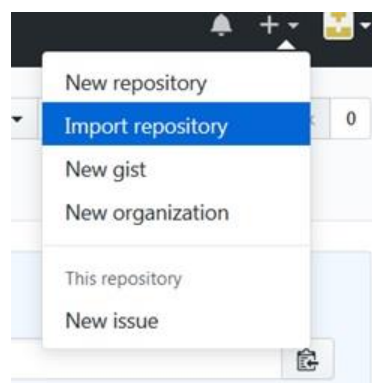


Pull

К слову, для командной разработки на Гитхабе есть ещё несколько важных опций.

Перенос информации из сторонних репозиториев на Гитхаб

Когда нужно собрать разрозненные кусочки кода в один проект, используйте кнопку **Import repository** и работайте с файлами в удобном репозитории Гитхаба.



Импортировать репозиторий

Кнопка **New gist** на этой панели предназначена для мгновенного обмена информацией.

А кнопка **New organization** открывает массу возможностей для командной разработки.

Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

Лабораторная работа «Разработка и интеграция модулей проекта (командная работа)»

Задание 6

Лабораторная работа «Разработка и интеграция модулей проекта (командная работа)»

Проверяемые результаты обучения: ПК2.1 ПК2.2 ПК2.3 ПК2.4 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

Разбиться на команды и выполнить разработку приложения

Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

Задание 7

Лабораторная работа «Отладка отдельных модулей программного проекта»

Проверяемые результаты обучения: ПК2.1 ПК2.2 ПК2.3 ПК2.4 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

1. Описать процесс отладки модуля и заполнить журнал ошибок

Ошибка	Описание

2. Описать применяемые средства отладки

Отладка программы в любом случае предполагает обдумывание и логическое осмысление всей имеющейся информации об ошибке. Большинство ошибок можно обнаружить по косвенным признакам посредством тщательного анализа текстов программ и результатов тестирования без получения дополнительной информации. При этом используют различные методы:

- ручного тестирования;
- индукции;
- дедукции;
- обратного прослеживания.

Метод ручного тестирования. Это - самый простой и естественный способ данной группы.

При обнаружении ошибки необходимо выполнить тестируемую программу вручную, используя тестовый набор, при работе с которым была обнаружена ошибка.

Метод очень эффективен, но не применим для больших программ, программ со сложными вычислениями и в тех случаях, когда ошибка связана с неверным представлением программиста о выполнении некоторых операций.

Данный метод часто используют как составную часть других методов отладки.

Метод индукции. Метод основан на тщательном анализе симптомов ошибки, которые могут проявляться как неверные результаты вычислений или как сообщение об ошибке. Если компьютер просто «зависает», то фрагмент проявления ошибки вычисляют, исходя из последних полученных результатов и действий пользователя. Полученную таким образом информацию организуют и тщательно изучают, просматривая соответствующий фрагмент программы. В результате этих действий выдвигают гипотезы об ошибках, каждую из которых проверяют. Если гипотеза верна, то детализируют информацию об ошибке, иначе - выдвигают другую гипотезу.

Последовательность выполнения отладки методом индукции показана на рис. 10.3 в виде схемы алгоритма.

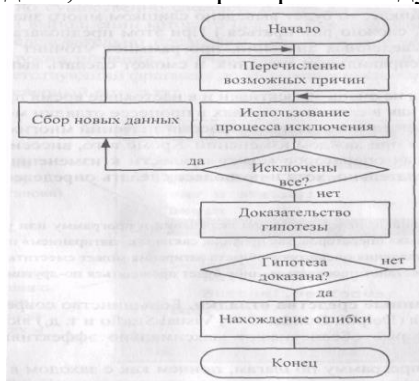
Самый ответственный этап - выявление симптомов ошибки. Организуя данные об ошибке, целесообразно записать все, что известно о ее проявлениях, причем фиксируют, как ситуации, в которых фрагмент с ошибкой выполняется нормально, так и ситуации, в которых ошибка проявляется. Если в результате изучения данных никаких гипотез не появляется, то необходима дополнительная информация об ошибке. Дополнительную информацию можно получить, например, в результате выполнения схожих тестов.

В процессе доказательства пытаются выяснить, все ли проявления ошибки объясняет данная гипотеза, если не все, то либо гипотеза не верна, либо ошибок несколько.



Рис. 10.3. Схема процесса отладки методом индукции

Метод дедукции. По методу дедукции вначале формируют множество причин, которые могли бы вызвать данное проявление ошибки. Затем анализируя причины, исключают те, которые противоречат имеющимся данным. Если все причины исключены, то следует выполнить дополнительное тестирование исследуемого фрагмента. В противном случае наиболее вероятную гипотезу пытаются доказать. Если гипотеза объясняет полученные признаю ошибки, то ошибка найдена, иначе -- проверяют следующую причину (рис. 10.4).



Метод обратного прослеживания. Для небольших программ эффективно применение метода обратного прослеживания. Начинают с точки вида неправильного результата. Для этой точки строится гипотеза о значениях основных переменных, которые могли бы привести к получению имеющегося результата.

Далее, исходя из этой гипотезы, делают предположения о значениях переменных в предыдущей точке. Процесс продолжают, пока не обнаружат причину ошибки.

Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

Задание 8

Лабораторная работа «Организация обработки исключений»

Проверяемые результаты обучения: ПК2.1 ПК2.2 ПК2.3 ПК2.4 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

Для ранее созданной программы применить обработку исключительной ситуации, используя примеры

try

{Операторы, выполнение которых может вызвать ошибку}

except

{операторы, которые должны быть выполнены в случае ошибки}

on on Exception1 do ...;

on Exception2 do ...;

...

else

.end;

Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

Тема 2.2.2 Инструментарий тестирования и анализа качества программных средств

Задание 1

Проверяемые результаты обучения: ПК2.1 ПК2.5

Вид учебной работы: аудиторная

Задание. Ответить на устные вопросы:

1. Отладка программных продуктов. Инструменты отладки. Отладочные классы.
2. Ручное и автоматизированное тестирование. Методы и средства организации тестирования.
3. Инструментарии анализа качества программных продуктов в среде разработки.
4. Обработка исключительных ситуаций. Методы и способы идентификации сбоев и ошибок.
5. Выявление ошибок системных компонентов.

Критерии оценки: Полнота ответа на поставленные вопросы

- «5»- Правильный уверенный ответ
- «4» - Уверенный ответ, допущена одна неточность
- «3» - Ответ близкий к правильному, с несколькими неточностями
- «2» - Неправильный ответ, отказывается отвечать

Задание 2

Лабораторная работа «Применение отладочных классов в проекте»

Проверяемые результаты обучения: ПК2.1 ПК2.2 ПК2.3 ПК2.4 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

Создать отладочные классы.

Классы Trace и Debug. Эти классы доступны в платформе .NET Framework корпорации Майкрософт. Эти классы можно использовать для предоставления сведений о производительности приложения во время разработки или после развертывания в рабочей среде. Эти классы являются только одной частью инструментальных средств, которые доступны в платформе .NET Framework.

Требования

В следующем списке представлены рекомендованные оборудование, программное обеспечение, сетевая инфраструктура и пакеты обновления, которые необходимы:

- Microsoft Windows 2000 или Microsoft Windows XP или Microsoft Windows Server 2003
- Microsoft Visual C#

Также предполагается, что вы знакомы с отладкой программы.

Описание методики

Действия, описанные в разделе Создание образца с помощью класса Debug демонстрируют, как создать консольное приложение, которое использует класс Debug для предоставления сведений о выполнении программы.

При запуске программы, можно использовать методы класса Debug для получения сообщений, которые помогают отслеживать последовательность выполнения программы для обнаружения неисправности или для предоставления сведений о измерении производительности. По умолчанию сообщения, которые выдает класс Debug отображаются в окне вывода из Visual Studio интегрированной среды разработки (IDE).

В примере кода используется метод WriteLine для создания сообщения, заканчивающееся ограничителем строки. При использовании этого метода для создания сообщения, каждое сообщение выводится в отдельной строке в окне вывода.

При использовании метода класса Debug Assert, окно вывода отображает сообщение только в том случае, если заданное условие не примет значение false. Сообщение также появляется в модальном диалоговом окне для пользователя. Диалоговое окно содержит сообщение, имя проекта и номер инструкции Debug.Assert. Диалоговое окно также содержит следующие три кнопки:

- **Прервать:** Приложение не будет выполняться.
- **Повторить:** Приложение входит в режим отладки.
- **Игнорировать:** Приложение выполняется.

Пользователь должен щелкнуть одну из этих кнопок, прежде чем приложение может продолжить работу.

Также существует возможность вывода из класса Debug назначения в окне вывода. Класс Debug имеет коллекцию прослушивателей, включающего объекты прослушивателя с именем.

Каждый объект прослушивателя отладки вывода контролирует и направляет выходные данные в заданном конечном расположении.

Каждый прослушиватель в коллекцию прослушивателей получает все выходные данные, создающий класс `Debug`. Класс `TextWriterTraceListener` для определения объектов прослушивателя. Можно задать целевой объект для класса `TextWriterTraceListener` с помощью его конструктора.

Ниже перечислены некоторые возможные выходные данные целевых объектов.

- В окне консоли с помощью свойства `System.Console.Out`.
- Текстовый файл (.txt) с помощью оператора `System.IO.File.CreateText("FileName.txt")`.

После создания объекта `TextWriterTraceListener`, необходимо добавить объект в коллекцию `Debug.Listeners` для получения выходных данных отладки.

Пример создания класса `Debug`

1. Запустите Visual Studio Express Edition или C#.
2. Создайте новый проект приложения консоли Visual C# с именем `conInfo`. В Visual Studio .NET будет создан класс `Class1`. В Visual Studio 2005 создается файл `Program.cs`.
3. Сверху в `Class1` или `Program.cs` добавьте следующее пространство имен.
`using System.Diagnostics;`
4. Чтобы инициализировать переменные для хранения информации о продукте, добавьте приведенные ниже операторы объявления в метод `Main`:
`string sProdName = "Widget";`
`int iUnitQty = 100;`
`double dUnitCost = 1.03;`
5. Определите сообщение, которое создает класс, как первый входной параметр метода `WriteLine`. Нажмите сочетание клавиш `CTRL + ALT + O`, чтобы убедиться, что окно вывода отображается.
`Debug.WriteLine("Debug Information-Product Starting ");`
6. Для удобства чтения используйте метод `Indent` для смещения вправо последующих сообщений в окне вывода:
`Debug.Indent();`
7. Чтобы отобразить содержимое выбранных переменных, используйте метод `WriteLine` следующим образом:
`Debug.WriteLine("The product name is " + sProdName);`
`Debug.WriteLine("The available units on hand are" + iUnitQty.ToString());`
`Debug.WriteLine("The per unit cost is " + dUnitCost.ToString());`
8. Можно также использовать метод `WriteLine` для отображения пространства имен и имени класса для существующего объекта. Например, следующий код отображает в окне вывода пространство имен `System.Xml.XmlDocument`:
`System.Xml.XmlDocument oxml = new System.Xml.XmlDocument();`
`Debug.WriteLine(oxml);`
9. Для систематизации выходных данных, можно включить категорию, как опциональный второй входной параметр метода `WriteLine`. Если указать категорию, формат выходного сообщения окна будет «Категория: сообщение.» Например, первая строка следующего кода отображает "Поле: название продукта — это мини-приложение" в окне вывода:
`Debug.WriteLine("The product name is " + sProdName,"Field");`
`Debug.WriteLine("The units on hand are" + iUnitQty,"Field");`
`Debug.WriteLine("The per unit cost is" + dUnitCost.ToString(),"Field");`
`Debug.WriteLine("Total Cost is" + (iUnitQty * dUnitCost),"Calc");`
10. Окно вывода может отображать сообщения только в том случае, если указанное условие имеет значение `true`, если использовать метод `WriteLineIf` класса `Debug`. Проверяемое условие является первым входным параметром метода `WriteLineIf`. Вторым

параметром WriteLineIf является сообщение, которое появляется только в том случае, если условие первого параметра примет значение true.

```
Debug.WriteLineIf(iUnitQty > 50, "This message WILL appear");
```

```
Debug.WriteLineIf(iUnitQty < 50, "This message will NOT appear");
```

11. Используйте метод Assert класса Debug для того, чтобы в окне вывода сообщения отображались только в том случае, если заданное условие примет значение false.

```
Debug.Assert(dUnitCost > 1, "Message will NOT appear");
```

```
Debug.Assert(dUnitCost < 1, "Message will appear since dUnitcost < 1 is false");
```

12. Создайте объекты TextWriterTraceListener для окна консоли (tr1) и текстовый файл с именем Output.txt (tr2), а затем добавьте каждый объект в коллекцию Debug Listeners :

```
TextWriterTraceListener tr1 = new TextWriterTraceListener(System.Console.Out);
```

```
Debug.Listeners.Add(tr1);
```

```
TextWriterTraceListener tr2 = new TextWriterTraceListener
```

```
(System.IO.File.CreateText("Output.txt"));
```

```
Debug.Listeners.Add(tr2);
```

13. Для удобства чтения, используйте метод Отменить отступ для удаления отступа для последующих сообщений

Создает класс Debug . При использовании методов Indent и Unindent вместе, читатель может определить вывод как группу.

```
Debug.Unindent();
```

```
Debug.WriteLine("Debug Information-Product Ending");
```

14. Чтобы убедиться в том, что каждый объект Listener получает все выходные данные, вызовите метод Flush для буферов класса Debug :

```
Debug.Flush();
```

Использование класса Trace

Можно также использовать класс Trace для получения сообщений, мониторинга выполнения приложения. Классы Trace и Debug используют большую часть одинаковых методов для получения выходных данных, включая следующие:

- **WriteLine**
- **WriteLineIf**
- **Отступ**
- **Отменить отступ**
- **Утверждение**
- **Очистка**

Классы Debug и Trace можно использовать отдельно или вместе в одном приложении. В проекте конфигурации отладки решения активны трассировки и отладки вывода. Проект создает выходные данные из обоих этих классов для всех объектов прослушивателя . Тем не менее решения конфигурации выпуска проекта только создает выходные данные из класса Trace . Конфигурация решения проекта Release игнорирует любые вызовы методов класса Debug.

```
Trace.WriteLine("Trace Information-Product Starting ");
```

```
Trace.Indent();
```

```
Trace.WriteLine("The product name is "+sProdName);
```

```
Trace.WriteLine("The product name is "+sProdName,"Field" );
```

```
Trace.WriteLineIf(iUnitQty > 50, "This message WILL appear");
```

```
Trace.Assert(dUnitCost > 1, "Message will NOT appear");
```

```
Trace.Unindent();
```

```
Trace.WriteLine("Trace Information-Product Ending");
```

Trace.Flush();

Console.ReadLine();

Убедитесь, что он работает

1. Убедитесь в том, что Debug является текущей конфигурацией решения.
2. Если окно **Обозревателя решений** не отображается, нажмите сочетание клавиш CTRL + ALT + L для отображения этого окна.
3. Щелкните правой кнопкой мыши **conInfo** и выберите команду **Свойства**.
4. В левой области страницы свойств conInfo в группе Папка **конфигурации** убедитесь, что стрелка указывает на В режиме **отладки**.
Примечание. В Visual C# 2005 и Visual C# 2005 Express Edition щелкните **Отладка** на странице **conInfo**.
5. На более высоком уровне **конфигурации** в Поле раскрывающегося списка **конфигурации**, нажмите кнопку **активный (Отладка)** или **отладки** нажмите кнопку **ОК**. В Visual C# 2005 и Visual C# 2005 Express Edition нажмите кнопку **активный (Отладка)** или **Отладка** в поле раскрывающегося списка **конфигурации** в страницу **Отладка** и нажмите кнопку **Сохранить** в меню **файл**.
6. Нажмите сочетание клавиш CTRL + ALT + O для отображения в окне вывода.
7. Нажмите клавишу F5 для запуска кода. При Появится **диалоговое окно**, нажмите кнопку **Игнорировать**.
8. В окне консоли нажмите клавишу ВВОД. Программа должна завершиться, и в окне вывода следует вывести следующего вида
Debug Information-Product Starting
The product name is Widget
The available units on hand are100
The per unit cost is 1.03
System.Xml.XmlDocument
Field: The product name is Widget
Field: The units on hand are100
Field: The per unit cost is1.03
Calc: Total Cost is 103
This message WILL appear
---- DEBUG ASSERTION FAILED ----
---- Assert Short Message ----
Message will appear since dUnitcost < 1 is false
---- Assert Long Message ----

at Class1.Main(String[] args) <%Path%>\class1.cs(34)

The product name is Widget
The available units on hand are100
The per unit cost is 1.03
Debug Information-Product Ending
Trace Information-Product Starting
The product name is Widget
Field: The product name isWidget
This message WILL appear

Trace Information-Product Ending

9. В окне консоли и файл Output.txt должны отображаться следующие выходные данные:

```
The product name is Widget
The available units on hand are 100
The per unit cost is 1.03
Debug Information-Product Ending
Trace Information-Product Starting
The product name is Widget
Field: The product name is Widget
This message WILL appear
Trace Information-Product Ending
```

Примечание. Файл Output.txt находится в том же каталоге исполняемого файла conInfo (conInfo.exe). Как правило это папка \bin, где хранится исходный проект. По умолчанию это C:\Documents and появлетсяия **пользователя для входа**\My Documents\Visual Studio Projects\conInfo\bin. В Visual C# 2005 и Visual C# 2005 Express Edition файл Output.txt находится в следующей папке:

C:\Documents and появлетсяия **пользователя для входа**\My Documents\Visual Studio 2005\Projects\conInfo\conInfo\bin\Debug

Полный пример кода

```
using System;
using System.Diagnostics;

class Class1
{
    [STAThread]
    static void Main(string[] args)
    {
        string sProdName = "Widget";
        int iUnitQty = 100;
        double dUnitCost = 1.03;
        Debug.WriteLine("Debug Information-Product Starting ");
        Debug.Indent();
        Debug.WriteLine("The product name is "+sProdName);
        Debug.WriteLine("The available units on hand are"+iUnitQty.ToString());
        Debug.WriteLine("The per unit cost is "+ dUnitCost.ToString());

        System.Xml.XmlDocument oxml = new System.Xml.XmlDocument();
        Debug.WriteLine(oxml);

        Debug.WriteLine("The product name is "+sProdName,"Field");
        Debug.WriteLine("The units on hand are"+iUnitQty,"Field");
        Debug.WriteLine("The per unit cost is"+dUnitCost.ToString(),"Field");
        Debug.WriteLine("Total Cost is "+(iUnitQty * dUnitCost),"Calc");

        Debug.WriteLineIf(iUnitQty > 50, "This message WILL appear");
        Debug.WriteLineIf(iUnitQty < 50, "This message will NOT appear");

        Debug.Assert(dUnitCost > 1, "Message will NOT appear");
        Debug.Assert(dUnitCost < 1, "Message will appear since dUnitcost < 1 is false");
    }
}
```

```
TextWriterTraceListener tr1 = new TextWriterTraceListener(System.Console.Out);
Debug.Listeners.Add(tr1);
```

```
TextWriterTraceListener tr2 = new
TextWriterTraceListener(System.IO.File.CreateText("Output.txt"));
Debug.Listeners.Add(tr2);
```

```
Debug.WriteLine("The product name is "+sProdName);
Debug.WriteLine("The available units on hand are"+iUnitQty);
Debug.WriteLine("The per unit cost is "+dUnitCost);
Debug.Unindent();
Debug.WriteLine("Debug Information-Product Ending");
Debug.Flush();
```

```
Trace.WriteLine("Trace Information-Product Starting ");
Trace.Indent();
```

```
Trace.WriteLine("The product name is "+sProdName);
Trace.WriteLine("The product name is"+sProdName,"Field" );
Trace.WriteLineIf(iUnitQty > 50, "This message WILL appear");
Trace.Assert(dUnitCost > 1, "Message will NOT appear");
```

```
Trace.Unindent();
Trace.WriteLine("Trace Information-Product Ending");
```

```
Trace.Flush();
```

```
Console.ReadLine();
```

```
}
}
```

Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

Задание 3

Лабораторная работа «Отладка проекта»

Проверяемые результаты обучения: ПК2.1 ПК2.2 ПК2.3 ПК2.4 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

1. Описать процесс отладки модуля и заполнить журнал ошибок

Ошибка	Описание

2. Описать применяемые средства отладки

Отладка программы в любом случае предполагает обдумывание и логическое осмысление всей имеющейся информации об ошибке. Большинство ошибок можно обнаружить по косвенным признакам посредством тщательного анализа текстов программ и результатов

тестирования без получения дополнительной информации. При этом используют различные методы:

- ручного тестирования;
- индукции;
- дедукции;
- обратного прослеживания.

Метод ручного тестирования. Это - самый простой и естественный способ данной группы.

При обнаружении ошибки необходимо выполнить тестируемую программу вручную, используя

тестовый набор, при работе с которым была обнаружена ошибка.

Метод очень эффективен, но не применим для больших программ, программ со сложными вычислениями и в тех случаях, когда ошибка связана с неверным представлением программиста о выполнении некоторых операций.

Данный метод часто используют как составную часть других методов отладки.

Метод индукции. Метод основан на тщательном анализе симптомов ошибки, которые могут проявляться как неверные результаты вычислений или как сообщение об ошибке. Если компьютер

просто «зависает», то фрагмент проявления ошибки вычисляют, исходя из последних полученных

результатов и действий пользователя. Полученную таким образом информацию организуют и тщательно изучают, просматривая соответствующий фрагмент программы. В результате этих действий выдвигают гипотезы об ошибках, каждую из которых проверяют. Если гипотеза верна, то детализируют информацию об ошибке, иначе - выдвигают другую гипотезу.

Последовательность выполнения отладки методом индукции показана на рис. 10.3 в виде схемы алгоритма.

Самый ответственный этап - выявление симптомов ошибки. Организуя данные об ошибке, целесообразно записать все, что известно о ее проявлениях, причем фиксируют, как ситуации, в которых фрагмент с ошибкой выполняется нормально, так и ситуации, в которых ошибка проявляется. Если в результате изучения данных никаких гипотез не появляется, то необходима дополнительная информация об ошибке. Дополнительную информацию можно получить, например, в результате выполнения схожих тестов.

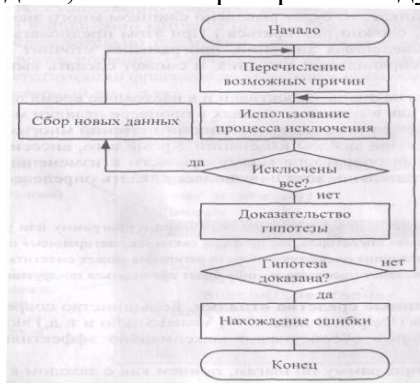
В процессе доказательства пытаются выяснить, все ли проявления ошибки объясняет данная гипотеза, если не все, то либо гипотеза не верна, либо ошибок несколько.



Рис. 10.3. Схема процесса отладки методом индукции

Метод дедукции. По методу дедукции вначале формируют множество причин, которые могли бы вызвать данное проявление ошибки. Затем анализируя причины, исключают те, которые противоречат имеющимся данным. Если все причины исключены, то следует выполнить дополнительное тестирование исследуемого фрагмента. В противном случае наиболее вероятную гипотезу пытаются доказать. Если гипотеза объясняет полученные

признаю ошибки, то ошибка найдена, иначе -- проверяют следующую причину (рис. 10.4).



Метод обратного прослеживания. Для небольших программ эффективно применение метода обратного прослеживания. Начинают с точки вода неправильного результата. Для этой точки строится гипотеза о значениях основных переменных, которые могли бы привести к получению имеющегося результата.

Далее, исходя из этой гипотезы, делают предположения о значениях переменных в предыдущей точке. Процесс продолжают, пока не обнаружат причину ошибки.

Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

Задание 4

Лабораторная работа «Инспекция кода модулей

Проверяемые результаты обучения: ПК2.1 ПК2.2 ПК2.3 ПК2.4 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

- 1.Провести инспектирование кода модулей, используя общепринятые соглашения и рекомендации по именованию и форматированию переменных, операторов, выражений.
- 2.Ответить на вопросы письменно:

- Для чего нужны стандарты кодирования.
- Какие существуют соглашения по именованию переменных, функций.
- В каком формате следует записывать константы в тексте программы.
- Какие стили расстановки скобок существуют? В чем их различия?

Требования:

- добавить комментарии в программный код;
- проверить правильность именования переменных, констант, методов;
- проверить правильность объявления переменных и констант;
- записать исправленный текст программы в двух стилях (согласно таблице, заданной выше).

Стандарты и руководства по кодированию

Различные модули, указанные в проектной документации, кодируются на этапе кодирования в соответствии со спецификацией модуля. Основной целью этапа кодирования

является кодирование из проектной документации, подготовленной после этапа проектирования, на языке высокого уровня, а затем модульное тестирование этого кода.

Хорошие организации по разработке программного обеспечения хотят, чтобы их программисты придерживались определенного и стандартного стиля кодирования, называемого стандартами кодирования. Обычно они создают свои собственные стандарты и руководства по кодированию в зависимости от того, что лучше всего подходит для их организации, и на основе типов программного обеспечения, которое они разрабатывают. Для программистов очень важно поддерживать стандарты кодирования, в противном случае код будет отклонен во время проверки кода.

Цель наличия стандартов кодирования:

- Стандарт кодирования дает единообразный вид кодам, написанным разными инженерами.
- Это улучшает удобочитаемость и удобство сопровождения кода, а также снижает сложность.
- Это помогает в повторном использовании кода и помогает легко обнаружить ошибку.
- Это продвигает здоровые методы программирования и увеличивает эффективность программистов.

Некоторые из стандартов кодирования приведены ниже:

12. Ограниченное использование глобалов:

Эти правила говорят о том, какие типы данных могут быть объявлены глобальными, а какие нет.

13. Стандартные заголовки для разных модулей:

Для лучшего понимания и обслуживания кода заголовков различных модулей должен соответствовать стандартному формату и информации. Формат заголовка должен содержать ниже вещи, которые используются в различных компаниях:

- Наименование модуля
- Дата создания модуля
- Автор модуля
- История изменений
- Краткое описание модуля о том, что делает модуль
- В модуле поддерживаются различные функции, а также входные и выходные параметры
- Глобальные переменные, доступные или измененные модулем

14. Соглашения об именах для локальных переменных, глобальных переменных, констант и функций:

Некоторые из соглашений об именах приведены ниже:

- Содержательное и понятное название переменных помогает любому понять причину его использования.
- Локальные переменные должны быть названы с использованием букв верблюда, начинающихся с маленькой буквы (например, **localData**), тогда как имена глобальных переменных должны начинаться с заглавной буквы (например, **GlobalData**). Имена констант должны быть сформированы только заглавными буквами (например, **CONSDATA**).
- Лучше избегать использования цифр в именах переменных.
- Названия функции должны быть написаны в верблюжьем регистре, начиная с маленьких букв.
- Название функции должно четко и кратко описывать причину ее использования.

15. Отступ:

Правильный отступ очень важен для улучшения читабельности кода. Чтобы сделать код читабельным, программисты должны правильно использовать пробелы. Некоторые из интервалов даны ниже:

- После запятой между аргументами функции должен быть пробел.
- Каждый вложенный блок должен иметь правильные отступы и интервалы.
- Надлежащий отступ должен быть в начале и в конце каждого кадра в программе.
- Все фигурные скобки должны начинаться с новой строки, а код, следующий за окончанием фигурных скобок, также начинается с новой строки.

16. Возвращаемые значения ошибок и соглашения об обработке исключений:

Все функции, которые сталкиваются с ошибкой, должны возвращать 0 или 1 для упрощения отладки.

С другой стороны, руководящие принципы кодирования дают некоторые общие рекомендации относительно стиля кодирования, которому необходимо следовать для улучшения понятности и читабельности кода. Некоторые из руководств по кодированию приведены ниже:

17. Избегайте использования стиля кодирования, который слишком сложен для понимания:

Код должен быть легко понятным. Сложный код делает обслуживание и отладку трудной и дорогой.

18. Избегайте использования идентификатора для нескольких целей:

Каждой переменной должно быть дано описательное и осмысленное имя, указывающее причину ее использования. Это невозможно, если идентификатор используется для нескольких целей, что может привести к путанице у читателя. Кроме того, это приводит к более трудностям во время будущих улучшений.

19. Код должен быть хорошо документирован:

Код должен быть правильно прокомментирован для понимания. Комментарии относительно утверждений повышают понятность кода.

20. Длина функций не должна быть очень большой:

Длинные функции очень трудно понять. Вот почему функции должны быть достаточно маленькими, чтобы выполнять небольшую работу, а длинные функции должны быть разбиты на маленькие для выполнения небольших задач.

21. Старайтесь не использовать оператор GOTO:

Оператор GOTO делает программу неструктурированной, что снижает ее понятность и затрудняет отладку.

Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

проекта»

Задание 5

Лабораторная работа «Тестирование интерфейса пользователя средствами инструментальной среды разработки

Проверяемые результаты обучения: ПК2.1 ПК2.2 ПК2.3 ПК2.4 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

1. Провести анализ требований к пользовательскому интерфейсу;
2. Разработать тест-требований и тест-планов для проверки пользовательского интерфейса;
3. Выполнить тестовые примеры и собрать информацию о выполнении тестов;
4. Определить полноту покрытия пользовательского интерфейса требованиями;
5. Составить отчет о проблемах в случае несовпадения поведения системы и требований либо в случае отсутствия требований на отдельные интерфейсные элементы.

Цели тестирования

Обеспечение качественного взаимодействия с пользователем – Выявление ошибок функциональности – Выявление необработанных исключений при взаимодействии с интерфейсом – Выявление потери или искажения данных, передаваемых через элементы интерфейса – Выявление ошибки в интерфейсе (несоответствие проектной документации, отсутствие элементов интерфейса)

Особенности тестирования

Тест планы в виде сценариев работы пользователя

Сценарии на естественном языке или в виде скриптов

Выполнение тестов в ручном режиме или с помощью эмулятора

Анализ экранных форм и видимых элементов, а не внутренних переменных

Покрытие — участие интерфейсных элементов в тестах

Найденный дефект: несоответствие реального поведения требованиям или проблемы в требованиях

Проблема отчета об ошибке: расплывчатость формулировок

Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

Задание 6

Лабораторная работа «Разработка тестовых модулей проекта для тестирования отдельных модулей»

Проверяемые результаты обучения: ПК2.1 ПК2.2 ПК2.3 ПК2.4 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

Написать программный код для тестирования отдельных модулей

Описать найденные ошибки при тестировании

Модульное тестирование. Этот вид тестирования представляет собой процесс проверки отдельных программных процедур и подпрограмм, входящих в состав программ или программных систем. Модульное тестирование производится непосредственным разработчиком и позволяет проверять все внутренние структуры и потоки данных в каждом модуле. Этот вид тестирования является частью этапа разработки. При модульном тестировании выполняется набор тестов, определяемый разработчиком так, чтобы охват тестированием каждого модуля был не менее 70...75 %.

Элементами модульного тестирования являются:

- синтаксическая проверка — проверка с использованием некоторого инструментального средства для выявления синтаксических ошибок в программном коде;

- проверка соответствия стандартам кодирования — проверка кода на соответствие стандартам кодирования компании;
- технический обзор программного кода.

После успешного завершения модульного тестирования все измененные модули и наборы тестов сохраняются в базе данных проекта.

Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

Задание 7

Лабораторная работа «Выполнение функционального тестирования»

Проверяемые результаты обучения: ПК2.1 ПК2.2 ПК2.3 ПК2.4 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

1. Разработать функцию в соответствии со своим вариантом.
2. Разработать функциональные тесты для написанного кода методом черного ящика. Добиваться 100% прохождения тестов не нужно. Необходимо описать принципы выбора тестов.

К отчету должна быть приложена спецификация на тесты в следующем формате:

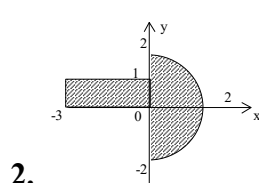
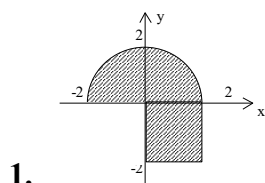
Тестируемая функциональность

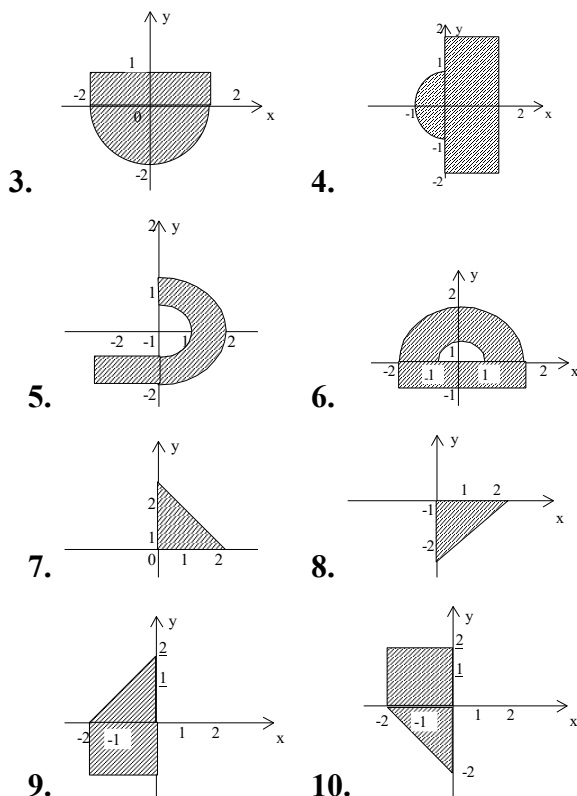
Имя теста	Описание сценария	Входные данные	Выходные данные

Например: Функция add(int,int)

Имя теста	Описание сценария	Входные данные	Выходные данные
addTwoPositive	сложение двух положительных чисел, проверка результата	Первый параметр 1 Второй параметр 2	Результат вызова 3
addTwoNegative	сложение двух отрицательных чисел, проверка результата	Первый параметр -1 Второй параметр -2	Результат вызова -3
addTwoTimes	Сложение двух положительных чисел, затем сложение двух отрицательных чисел, затем сложение результатов	При первом вызове Первый параметр 1 Второй параметр 2 При втором вызове Первый параметр -1 Второй параметр -2	Результат первого вызова 3, Результат второго вызова -3, Результат третьего вызова 0

Даны x , y . Выяснить, принадлежит ли точка (x,y) фигуре, изображенной на рисунке. Ответ выдать в форме: принадлежит, не принадлежит.





Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

Задание 8

Лабораторная работа Тестирование интеграции»

Проверяемые результаты обучения: ПК2.1 ПК2.2 ПК2.3 ПК2.4 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

1. Разработать тест-плана – руководства к действию для тестировщиков;
2. Сформировать тестовых данных и создание тест-кейсов;
3. Реализовать сценарии для запуска тест-кейсов;
4. Выполнить тест-кейсов и исправление ошибок;
5. Повторить цикл тестирования до успешной интеграции

Интеграционное тестирование – вид тестирования, при котором на соответствие требований проверяется интеграция модулей, их взаимодействие между собой, а также интеграция подсистем в одну общую систему. Для интеграционного тестирования используются компоненты, уже проверенные с помощью модульного тестирования, которые группируются в множества.



Существует несколько подходов к интеграционному тестированию:

- Снизу вверх. Сначала собираются и тестируются модули самих нижних уровней, а затем по возрастанию к вершине иерархии. Данный подход требует готовности всех собираемых модулей на всех уровнях системы.
- Сверху вниз. Данный подход предусматривает движение с высокоуровневых модулей, а затем направляется вниз. При этом используются заглушки для тех модулей, которые находятся ниже по уровню, но включение которых в тест еще не произошло.
- Большой взрыв. Все модули всех уровней собираются воедино, а затем тестируется. Данный метод экономит время, но требует тщательной проработки тест кейсов.

Преимущество.

Интеграционное тестирование позволяет имитировать действия пользователей и быстро получать подтверждение, что программный продукт успешно взаимодействует с другими системами. Такой подход гарантирует сразу несколько преимуществ:

1. Предотвращение появления критичных ошибок в опытно-промышленной эксплуатации
2. Снижение влияния человеческого фактора
3. Экономия затрат на исправление дефектов

Главной задачей интеграционного тестирования является поиск ошибок, связанных с взаимодействием модулей системы или нескольких систем. В результате все смежные системы и модули одной системы должны работать согласованно.

Способы проведения интеграционного тестирования подбираются в зависимости от интеграционных решений.

Этапы

Разработайте план тестирования Вашего программного комплекса. При подготовке контрольных тестов воспользуйтесь вышеописанными рекомендациями. Тесты должны отображать:

- a. типичную ситуацию;
- b. ненормальную ситуацию;
- c. граничные значения;
- d. затрагивающие причинно–следственные связи.

Одним из тестов является контрольный пример из раздела «Постановка задачи

Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

Задание 9

Лабораторная работа «Документирование результатов тестирования»

Проверяемые результаты обучения: ПК2.1 ПК2.2 ПК2.3 ПК2.4 ПК2.5

Вид учебной работы: аудиторная работа

Задание:

- 1.Выполнить поиск документов тестирования
- 2.Описать документ в виде

Название документа	Описание

3.Собрать документы по тестированию для своего проекта

4.Продемонстрировать преподавателю.

Критерии оценки:

- «5»- правильное решение, допущено одно исправление, может обосновать решение
- «4» - правильное решение, допущено два-три исправления, может обосновать решение
- «3» - правильное решение, не может обосновать решение
- «2» - задания не решены, не может обосновать решение

РАЗДЕЛ. 3 МОДЕЛИРОВАНИЕ В ПРОГРАММНЫХ СИСТЕМАХ

МДК 2.3 МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

Тема 2.3.1. Основы моделирования. Детерминированные задачи

Проверяемые результаты обучения: ПК 2.1, ПК 2.4, ПК 2.5

Вид учебной работы: аудиторная

Задание 1 Составить экономико-математическую модель задачи

Производитель элементов центрального отопления изготавливает радиаторы четырех моделей. Ограничения на производство обусловлены количеством рабочей силы и количеством стальных листов, из которых изготавливаются радиаторы.

Модель радиатора	A	B	C	D	Фонд
Необходимое количество рабочей силы, человеко-часы	0,5	1,5	2	1,5	100
Необходимое количество стального листа, м2	4	2	6	8	80
Прибыль от продажи одного радиатора, долл.	5	5	12,5	10	

Решите эту задачу с максимизацией прибыли в качестве целевой функции.

Критерии оценки: Правильность составления экономико-математической модели задачи

«5» - модель составлена верно, без ошибок

«4» - модель составлена верно, но присутствуют 1-2 неточности

«3» - модель составлена, но присутствуют 1-2 ошибки

«2» - модель составлена неверно, либо вообще не составлена

Задание: Лабораторные занятия №1 «Построение простейших математических моделей.

Построение простейших статистических моделей»

Решить транспортную задачу. Для откорма скота необходимо 4 вида продуктов А, В, С, D. Фуражная промышленность производит два сорта кормов М и N, которые содержат эти составляющие.

Продукт	Содержание в корме (кг)		Требуемое кол-во продуктов ежедневно одному животному (кг)
	М	N	
A	0,1	0	0,4
B	0	0,1	0,6
C	0,1	0,2	2
D	0,2	0,1	1,7
Цена за 1 кг	10	4	

Определить, какое количество кормов того и другого сорта необходимо скармливать ежедневно каждому животному, чтобы кормление скота обошлось как можно дешевле.

Критерии оценки:

«5» - выполнено верно задание

«4» - выполнено верно задание, но имеются недочеты

«3» - выполнено верно задание, но в работе есть ошибки

«2» - задание выполнено неверно, либо вообще не выполнено

Задание 2. Решить симплекс-методом задачу линейного программирования

$$F(x)=7x_1-2x_2 \rightarrow \max$$

$$5x_1-2x_2 \leq 3$$

$$-3x_1+x_2 \leq 3$$

$$2x_1+x_2 \leq 4$$

$$x_1 \geq 0, x_2 \geq 0$$

Критерии оценки: Правильность составления экономико-математической модели задачи

«5» - модель составлена верно, без ошибок

«4» - модель составлена верно, но присутствуют 1-2 неточности

«3» - модель составлена, но присутствуют 1-2 ошибки

«2» - модель составлена неверно, либо вообще не составлена

Задание: Лабораторные занятия №2 Решение простейших однокритериальных задач

Найти максимум функции $F(X)$ при заданных ограничениях.

$$F(x)=2x_1+3x_2 \rightarrow \max$$

$$x_1 + x_2 \leq 2$$

$$x_1 + x_2 \geq -1$$

$$3x_1 + 2x_2 \leq 10$$

$$x_1 \geq 0, x_2 \geq 0$$

Критерии оценки:

«5» - выполнено верно задание

«4» - выполнено верно задание, но имеются недочеты

«3» - выполнено верно задание, но в работе есть ошибки

«2» - задание выполнено неверно, либо вообще не выполнено

Задание: Лабораторные занятия №3 Решение задач линейного программирования симплекс-методом

Решить транспортную задачу.

$$F(x)=5x_1+3x_2 \rightarrow \max$$

$$3x_1 + 5x_2 \leq 15$$

$$5x_1 + 2x_2 \leq 10$$

$$2x_1 + 2x_2 \leq 6$$

$$x_1 \geq 0, x_2 \geq 0$$

Критерии оценки:

«5» - выполнено верно задание

«4» - выполнено верно задание, но имеются недочеты

«3» - выполнено верно задание, но в работе есть ошибки

«2» - задание выполнено неверно, либо вообще не выполнено

Задание: Практическая работа №1 «Сведение произвольной задачи линейного программирования к основной задаче линейного программирования»

Решить исходную ЗЛП симплекс-методом составить двойственную задачу линейного программирования, установить сопряженные пары прямой и двойственной задач, согласно сопряженным парам переменных из решения прямой задачи получить решение двойственной задачи, сравнить результаты.

$$x_1 + x_2 \rightarrow \max$$

$$4x_1 - x_2 \leq 10$$

$$2x_1 + 5x_2 \leq 12$$

$$x_1, x_2 \geq 0$$

Критерии оценки:

«5» - выполнено верно задание

«4» - выполнено верно задание, но имеются недочеты

«3» - выполнено верно задание, но в работе есть ошибки

«2» - задание выполнено неверно, либо вообще не выполнено

Задание. Решить транспортную задачу.

					$\mathbf{a}_i$
	1	8	2	3	30
$\ C_{ij}\ =$	4	7	5	1	50
	5	3	4	4	20
$\mathbf{b}_j$	15	15	40	30	

Критерии оценки: Правильность составления экономико-математической модели задачи

Задание. Лабораторные занятия №4 «Нахождение начального решения транспортной задачи. Решение транспортной задачи методом потенциалов»

Решить транспортную задачу.

				a_i	
	7	4	3	4	40
$\ C_{ij}\ =$	7	1	5	10	35
	3	13	6	7	15
b_i	35	10	40	15	

Критерии оценки:

- «5» - выполнено верно задание
- «4» - выполнено верно задание, но имеются недочеты
- «3» - выполнено верно задание, но в работе есть ошибки
- «2» - задание выполнено неверно, либо вообще не выполнено

Задание: Лабораторные занятия №5 Задача Коши для уравнения теплопроводности

Найти решение уравнения

$$u_t = a^2 u_{xx}$$

для следующего распределения температуры стержня:

$$u(x,t)|_{t=0} = \varphi(x) = \begin{cases} u_0 & \text{при } x_1 < x < x_2, \\ 0 & \text{при } x < x_1 \text{ или } x > x_2. \end{cases}$$

Критерии оценки:

- «5» - выполнено верно задание
- «4» - выполнено верно задание, но имеются недочеты
- «3» - выполнено верно задание, но в работе есть ошибки
- «2» - задание выполнено неверно, либо вообще не выполнено

Задание. Решить графическим методом задачу линейного программирования. Найти максимум и минимум функции $F(X)$ при заданных ограничениях.

$$F(x) = 6x_1 - 2x_2 \rightarrow \max$$

$$5x_1 - 2x_2 \leq 6$$

$$-3x_1 + x_2 \leq 3$$

$$2x_1 + x_2 \leq 8$$

$$x_1 \geq 0, x_2 \geq 0$$

Критерии оценки: Правильность построения в плоскости области допустимых решений. Правильность нахождения целевой функции $F(X)$ при заданных ограничениях.

«5» - выполнено верно задание

«4» - выполнено верно задание, но имеются недочеты

«3» - выполнено верно задание, но в работе есть ошибки

«2» - задание выполнено неверно, либо вообще не выполнено

Задание. Лабораторные занятия №6 «Применение метода стрельбы для решения линейной краевой задачи»

Методом стрельбы решить следующую первую краевую задачу для ОДУ второго порядка (точность $\varepsilon = 0,01$) с шагом $h = 0,25$:

$$\begin{cases} y' = -xy' + e^x, \\ y(0) = 0, \\ y(1) = 1. \end{cases}$$

Критерии оценки:

«5» - выполнено верно задание

«4» - выполнено верно задание, но имеются недочеты

«3» - выполнено верно задание, но в работе есть ошибки

«2» - задание выполнено неверно, либо вообще не выполнено

Задание. Решить задачу динамического программирования

Распределить 500 тысяч рублей инвестиции между проектами с целью получения максимальной прибыли (суммы распределять частями, кратными 100 тысячам)

Сумма средств (тыс руб)	Проект 1	Проект 2	Проект 3	Проект 4
100	120	125	130	130
200	220	240	235	240
300	350	330	340	330
400	440	420	430	420
500	530	500	510	510

Критерии оценки:

«5» - выполнено верно задание

«4» - выполнено верно задание, но имеются недочеты

«3» - выполнено верно задание, но в работе есть ошибки

«2» - задание выполнено неверно, либо вообще не выполнено

Задание. Лабораторные занятия №7 «Задача о распределении средств между предприятиями. Задача о замене оборудования. Нахождение кратчайших путей в графе. Решение задачи о максимальном потоке»

- 1) Фирма имеет возможность рекламировать свою продукцию, используя местные радио и телесети. Затраты на рекламу в бюджете фирмы ограничены величиной 1000 долл. в месяц. Каждая минута радиорекламы обходится в 5 долл., а каждая минута телерекламы - в 100 долл. Фирма хотела бы использовать радиосеть по крайней мере в 2 раза чаще, чем сеть телевидения. Опыт прошлых лет показал, что объем сбыта, который обеспечивает каждая минута телерекламы, в 25 раз больше сбыта, обеспечиваемого одной минутой радиорекламы.

Определить оптимальное распределение финансовых средств, ежемесячно отпускаемых на рекламу, между радио и телерекламой

- 2) Пусть в эксплуатации находится некоторое оборудование. Покупная цена оборудования S . Известны затраты на эксплуатацию оборудования (уход за ним, ремонт т.д.), производимые в начале $(1, 2, \dots, t, \dots, n)$ периодов. Предположим, что периоды равны году. Обозначим затраты, производимые в t – й период, через C_t . В результате старения балансовая цена оборудования непрерывно падает и зависит от периода списания, обозначим ее St . Требуется определить период списания оборудования, чтобы затраты на единицу времени были минимальны.
- 3) Заменить колесо машины (работу выполняют два человека). Достать из багажника домкрат и инструменты (40 с); снять диск с колеса (30 с); освободить колесо (50 с); поставить домкрат под машину (26 с); поднять машину (20 с); из багажника взять запасное колесо (25 с); снять гайки и колесо (20 с); установить запасное колесо на ось (10 с); завинтить (не сильно) гайки на оси (15 с); опустить машину и собрать домкрат (25 с); поставить домкрат обратно в багажник (10 с); завинтить гайки на оси до конца (12 с); положить плохое колесо и инструменты в багажник (40 с); поставить на место диск колеса (10 с). Построить граф. Найти наикратчайший путь в графе.

Критерии оценки:

- «5» - выполнено верно задание
- «4» - выполнено верно задание, но имеются недочеты
- «3» - выполнено верно задание, но в работе есть ошибки
- «2» - задание выполнено неверно, либо вообще не выполнено

Тема 2.3.2 Задачи в условиях неопределенности

Проверяемые результаты обучения: ПК 2.1, ПК 2.4, ПК 2.5

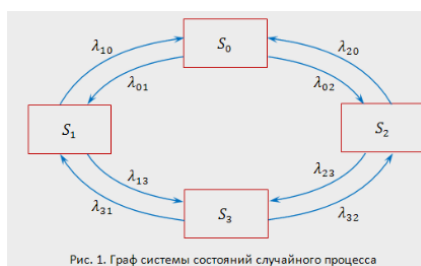
Вид учебной работы: аудиторная

Задание: Технологическая система состоит из одного станка. На станок поступают заявки на изготовление деталей в среднем через 0,5 часа ($\bar{t}_3 = 0,5ч.$). Среднее время изготовления одной детали равно $\bar{t}_{об} = 0,6ч$. Если при поступлении заявки на изготовление детали станок занят, то она (деталь) направляется на другой станок. Найти абсолютную и относительную пропускную способности системы и вероятность отказа по изготовлению детали.

Критерии оценки:

- «5» - выполнено верно задание
- «4» - выполнено верно задание, но имеются недочеты
- «3» - выполнено верно задание, но в работе есть ошибки
- «2» - задание выполнено неверно, либо вообще не выполнено

Задание: По графу состояний составить уравнения Колмогорова



Критерии оценки:

- «5» - выполнено верно задание
- «4» - выполнено верно задание, но имеются недочеты
- «3» - выполнено верно задание, но в работе есть ошибки
- «2» - задание выполнено неверно, либо вообще не выполнено

Задание: Практическая работа №2 «Составление систем уравнений Колмогорова. Нахождение финальных вероятностей. Нахождение характеристик простейших систем массового обслуживания»

Найти предельные вероятности для системы S задачи 1, граф состояний которой приведен на рис. 1, при $l_{01}=1$, $l_{02}=2$, $l_{10}=2$, $l_{13}=2$, $l_{20}=3$, $l_{23}=1$, $l_{31}=3$, $l_{32}=2$.

Критерии оценки:

- «5» - выполнено верно задание
- «4» - выполнено верно задание, но имеются недочеты
- «3» - выполнено верно задание, но в работе есть ошибки
- «2» - задание выполнено неверно, либо вообще не выполнено

Задание: Практическая работа №3 «Решение задач массового обслуживания методами имитационного моделирования»

Интервалы прихода клиентов в парикмахерскую с одним креслом распределены равномерно на интервале $(28+N) \pm (16+N/2)$ мин. Время стрижки также распределено равномерно на интервале $(16+N) \pm (4+N/4)$ мин. Клиенты приходят в парикмахерскую, стригутся в порядке очереди: «первым пришел - первым обслужился». Необходимо построить GPSS-модель парикмахерской, которая должна обеспечить сбор статистических данных об очереди и загрузке парикмахера. Работу парикмахерской промоделировать в течение 8 часов. Определить коэффициент загрузки парикмахера и среднее время ожидания клиентов в очереди.

Критерии оценки:

- «5» - выполнено верно задание
- «4» - выполнено верно задание, но имеются недочеты
- «3» - выполнено верно задание, но в работе есть ошибки
- «2» - задание выполнено неверно, либо вообще не выполнено

Задание: Дан временной ряд $x = (5, 1, 1, 3, 2, 9, 6, 2, 5, 2)$. Вычислите среднее, дисперсию (смещенную), автоковариационную и автокорреляционную матрицы

Критерии оценки:

- «5» - выполнено верно задание
- «4» - выполнено верно задание, но имеются недочеты
- «3» - выполнено верно задание, но в работе есть ошибки
- «2» - задание выполнено неверно, либо вообще не выполнено

Задание: Практическая работа №4 «Построение прогнозов»

По 15 предприятиям отрасли (табл. 1.1) известны: x – объем произведенной продукции (тыс. ед.) и y – затраты на выпуск этой продукции (тыс. ден. ед.). Необходимо:

- 1) определить зависимость y от x ;
- 2) построить корреляционные поля и график уравнения линейной регрессии y на x ;
- 3) сделать вывод о качестве модели и рассчитать прогнозное значение y при прогнозном значении x , составляющем 107% от среднего уровня.

Таблица 1.1

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x	2,7	1,5	8,2	4,5	3,3	5,8	3,0	7,1	1,2	10,4	4,9	5,2	11,5	9,4	6,5
y															

Критерии оценки:

- «5» - выполнено верно задание
- «4» - выполнено верно задание, но имеются недочеты
- «3» - выполнено верно задание, но в работе есть ошибки
- «2» - задание выполнено неверно, либо вообще не выполнено

Задание: Лабораторные занятия №8 «Моделирование прогноза»

Для временного ряда $x = (7, -9, 10, -2, 21, 13, 40, 36, 67, 67)$ оцените параметры полиномиального тренда второго порядка. Постройте точечный и интервальный прогнозы по тренду на 2 шага вперед.

Критерии оценки:

- «5» - выполнено верно задание
- «4» - выполнено верно задание, но имеются недочеты
- «3» - выполнено верно задание, но в работе есть ошибки
- «2» - задание выполнено неверно, либо вообще не выполнено

Задание: Игрок А записывает одно из двух чисел: 1 или 2, игрок В— одно из трех чисел: 1, 2 или 3. Если оба числа одинаковой четности, то А выигрывает и выигрыш равен сумме этих чисел, если четности выбранных игроками чисел не совпадают, то В выигрывает, выигрыш равен сумме этих чисел. Построить платежную матрицу игры, определить нижнюю и верхнюю цены игры и проверить наличие седловой точки.

Критерии оценки:

- «5» - выполнено верно задание
- «4» - выполнено верно задание, но имеются недочеты
- «3» - выполнено верно задание, но в работе есть ошибки
- «2» - задание выполнено неверно, либо вообще не выполнено

Задание: Практическая работа №5 Решение матричной игры методом итераций

Для следующих платежных матриц определить нижнюю и верхнюю цены игры, минимаксные стратегии и оптимальные решения игры, если существует седловая точка.

1

$$\begin{pmatrix} 0,3 & 0,6 & 0,8 \\ 0,9 & 0,4 & 0,2 \\ 0,7 & 0,5 & 0,4 \end{pmatrix}$$

2

$$\begin{pmatrix} 4 & 5 & 3 \\ 6 & 7 & 4 \\ 5 & 2 & 3 \end{pmatrix}$$

3

$$\begin{pmatrix} 8 & 9 & 9 & 4 \\ 6 & 5 & 8 & 7 \\ 3 & 4 & 5 & 6 \end{pmatrix}$$

4

$$\begin{pmatrix} 4 & 5 & 6 & 7 & 9 \\ 3 & 4 & 6 & 5 & 6 \\ 7 & 6 & 10 & 8 & 11 \\ 8 & 5 & 4 & 7 & 3 \end{pmatrix}$$

5

$$\begin{pmatrix} 4 & 9 & 5 & 3 \\ 7 & 8 & 6 & 9 \\ 7 & 4 & 2 & 6 \\ 8 & 3 & 4 & 7 \end{pmatrix}$$

6

$$\begin{pmatrix} 2 & 5 & 3 \\ 6 & 4 & 5 \\ 3 & 7 & 6 \\ 2 & 3 & 4 \end{pmatrix}$$

Критерии оценки:

«5» - выполнено верно задание

«4» - выполнено верно задание, но имеются недочеты

«3» - выполнено верно задание, но в работе есть ошибки

«2» - задание выполнено неверно, либо вообще не выполнено

Задание: Лабораторные занятия №9 «Выбор оптимального решения с помощью дерева решений»

Фермер может выращивать либо кукурузу, либо соевые бобы. Вероятность того, что цены на будущий урожай этих культур повысятся, останутся на том же уровне или понизятся, равна соответственно 0,25, 0,30 и 0,45. Если цены возрастут, урожай кукурузы даст 30 000 долл. чистого дохода, а урожай соевых бобов — 10 000 долл. Если цены останутся неизменными, фермер лишь покроет расходы. Но если цены станут ниже, урожай кукурузы и соевых бобов приведет к потерям в 35 000 и 5 000 долл. соответственно. Постройте дерево решений. Какую культуру следует выращивать фермеру? Каково ожидаемое значение его прибыли?

Критерии оценки:

«5» - выполнено верно задание

«4» - выполнено верно задание, но имеются недочеты

«3» - выполнено верно задание, но в работе есть ошибки

«2» - задание выполнено неверно, либо вообще не выполнено

4. КОНТРОЛЬ ПРИОБРЕТЕНИЯ ПРАКТИЧЕСКОГО ОПЫТА.ОЦЕНКА ПО УЧЕБНОЙ И (ИЛИ) ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ

4.1. Общие положения

Целью оценки по учебной и (или) производственной практике является оценка: 1) профессиональных и общих компетенций; 2) практического опыта и умений.

Оценка по учебной и (или) производственной практике выставляется на основании данных аттестационного листа (характеристики профессиональной деятельности обучающегося/студента на практике) с указанием видов работ, выполненных обучающимся во время практики, их объема, качества выполнения в соответствии с технологией и (или) требованиями организации, в которой проходила практика.

4.2. Виды работ практики и проверяемые результаты обучения по профессиональному модулю

Таблица 11

Иметь практический опыт	Виды и объем работ на учебной практике, требования к их выполнению и/ или условия выполнения	Документ, подтверждающий качество выполнения работ
1	2	3
модели процесса разработки программного обеспечения; основные принципы процесса разработки программного обеспечения; основные подходы к интегрированию программных модулей; основы верификации и аттестации программного обеспечения	Анализ предметной области. Разработка и оформление технического задания. Создание команды разработчиков. Проектирование архитектуры программного средства. Построение диаграмм UML. Работа в системе контроля версий. Интегрирование программных модулей. Разработка и применение тестовых сценариев. Оценка программных средств с помощью метрик. Инспекция программного кода на предмет соответствия стандартам кодирования.	Отчет по практике

Таблица 12

Иметь практический опыт	Виды и объем работ на производственной практике, требования к их выполнению и/ или условия выполнения	Документ, подтверждающий качество выполнения работ
1	2	3
модели процесса разработки программного обеспечения; основные принципы процесса разработки программного обеспечения; основные подходы к интегрированию программных модулей; основы верификации и аттестации программного	Сбор информации о предприятии Выявление не автоматизированного процесса Анализ требований к системе Стадия техническое задание Стадия эскизный проект Стадия технический проект Разработка интерфейса программы Оформление отчета	Дневник Аттестационный лист Отзыв характеристика Отчет

обеспечения	Разработка диаграмм программной системы Разработка модулей и их интеграция в программную систему Отладка программного продукта с использованием специализированных программных средств. Разработка тестовых наборов и тестовых сценариев. Инспектирование компонент программного продукта на предмет соответствия стандартам кодирования	
-------------	--	--

4.3. Форма аттестационного листа

Аттестационный лист

_____,
 Ф.И.О. студента
 проходившего(ей) учебную/производственную (преддипломную) практику
 по профессиональному модулю _____
 наименование профессионального модуля
 в организации _____,
 наименование организации, юридический адрес
 в объеме _____ час. с «__» _____ 20__ г. по «__» _____ 20__ г.
 Виды и объем работ, выполненные обучающимся во время практики _____
 _____.

Виды

Профессиональные компетенции (код и наименование)	Основные показатели оценки результата	Оценка выполнения работ (по 5-бальной системе)
ПК 2.1. Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент	ОПОР 1.1.....	
	ОПОР 1.2.....	
		
ПК 2.2. Выполнять интеграцию модулей в программное обеспечение		
		...
ПК2.3. Выполнять отладку программного модуля с использованием специализированных программных средств		
ПК2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для		

ПК2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования		

«__» _____ 20__ г.

Оценка работодателя _____ (отлично, хорошо, удовлетворительно, неудовлетворительно)

Работодатель _____ / _____

Ниже заполняется в колледже:

Оценка отчета _____

Итоговая оценка по результатам защиты отчета _____

Руководитель практики _____ / _____

«__» _____ 20__ г.

ПАКЕТ ЭКЗАМЕНАТОРА

МДК. 2.1 Технология разработки программного обеспечения

ПАСПОРТ

Назначение:

КОМ предназначен для контроля и оценки результатов освоения ПМ02 02.01 Технология разработки программного обеспечения специальности СПО 09.02.07 «Информационные системы и программирование»

Итогом экзамена является пятибалльная оценка.

Проверяемые результаты обучения: ПК2.2 ПК2.3 ПК2.4 ПК2.5

Инструкция

Внимательно прочитайте задание.

Вы можете воспользоваться средой программирования, BPWin

Время выполнения задания – 1 час

Теоретические вопросы

1. Понятия требований, классификация, уровни требований. Методологии и стандарты, регламентирующие работу с требованиями.
2. Современные принципы и методы разработки программных приложений
3. Методы организации работы в команде разработчиков. Системы контроля версий
4. Основные подходы к интегрированию программных модулей
5. Стандарты кодирования
6. Описание и оформление требований (спецификация). Анализ требований и стратегии выбора решения
7. Выбор архитектуры ПО
8. Структура и формат данных
9. Модульное программирование
10. Анализ требований и определение спецификаций при структурном подходе
11. Анализ требований и определение спецификаций при объектном подходе
12. Диаграммы потоков данных. Диаграммы сущность-связь.
13. Описание требований: унифицированный язык моделирования - краткий словарь. Диаграммы UML.

14. Цели и задачи и виды тестирования. Стандарты качества программной документации. Меры и метрики.
15. Тестовое покрытие.
16. Тестовый сценарий, тестовый пакет.
17. Анализ спецификаций.
18. Верификация программного обеспечения.
19. Аттестация программного обеспечения.

Практическая часть

Задание 1

1. Матрица $A [M, N]$ (M кратно 4) разделена по вертикали на две половины. Определить сумму элементов каждого столбца левой половины и сумму элементов каждого четного столбца правой половины матрицы A . Написать программу и диаграмму потоков данных.
2. Сведения об ученике состоят из его имени, фамилии и названия класса (года обучения и буквы), в котором он учится. Выяснить, имеются ли в школе однофамильцы. Результаты вывести в табличной форме. Написать программу и функциональную диаграмму.

Задание 2

1. Вычислить сумму и число положительных элементов матрицы $A[N, N]$, находящихся над главной диагональю. Написать программу и диаграмму потоков данных.
2. Сведения об автомобиле состоят из его марки, номера и фамилии владельца. Найти фамилии владельцев и номера автомобилей данной марки. Результаты вывести в табличной форме. Написать программу и функциональную диаграмму.

Задание 3

1. Дана квадратная целочисленная матрица порядка n . Сформировать результирующий одномерный массив, элементами которого являются строчные суммы тех строк, которые начинаются с k идущих подряд положительных чисел. Написать программу и диаграмму потоков данных.
2. Некоторый массив содержит различные даты. Каждая дата — это число, номер месяца и год. Найти и вывести на экран самую позднюю дату. Результат вывести в табличной форме. Написать программу и функциональную диаграмму.

Задание 4

1. Матрица $A[N, M]$ (M кратно 4) разделена по вертикали на две половины. Определить сумму элементов каждого столбца левой половины и сумму элементов каждого четного столбца правой половины матрицы A . Написать программу и диаграмму потоков данных.
2. Некоторый массив содержит данные о кубиках: размер каждого кубика (длина ребра в сантиметрах), его цвет (красный, желтый, зеленый или синий) и материал (деревянный, металлический, картонный). Найти количество кубиков каждого их перечисленных цветов и их суммарный объем. Результаты вывести в табличной форме. Написать программу и функциональную диаграмму.

Задание 5

1. Определить номера строк матрицы $R[M, N]$, хотя бы один элемент которых равен c , и элементы этих строк умножить на d . Написать программу и диаграмму потоков данных.
2. Некоторый массив содержит различные даты. Каждая дата — это число, название месяца и год. Найти и вывести на экран все весенние даты. Написать программу и функциональную диаграмму.
3. Контекстная и детализированная диаграмма в BPWIN для системы описывающей порядок организации общеинститутского студенческого мероприятия.

Задание 6

1. Расположить столбцы матрицы $D[M, N]$ в порядке возрастания элементов k -ой строки ($1 < k < M$). Написать программу и диаграмму потоков данных.
2. Сведения о книгах включают фамилию автора, название и год издания. Найти названия книг данного автора, изданных с 1980 г. Результаты вывести в табличной форме. Написать программу и функциональную диаграмму.

Задание 7

1. Определить наименьший элемент каждой четной строки матрицы $A[M, N]$. Написать программу и диаграмму потоков данных.
2. Сведения о книгах — это фамилия автора, название и год издания. Определить, имеется ли книга с названием «Информатика». Если да, то вывести фамилию автора и год издания. Если таких книг несколько, то вывести информацию обо всех этих книгах в табличной форме. Написать программу и функциональную диаграмму.

Задание 8

1. Определить номера тех строк целочисленной матрицы $A[N, K]$, которые совпадают с массивом $D[K]$. Если таких строк нет, выдать соответствующее сообщение. Написать программу и диаграмму потоков данных.
2. Словарная статья включает английское слово, часть речи и русский перевод. Упорядочить словарь по английскому алфавиту. Результаты вывести в табличной форме. Написать программу и функциональную диаграмму.

Задание 9

1. Пусть дана действительная матрица размером $n \times m$. Требуется преобразовать матрицу: поэлементно вычесть последнюю строку из всех строк, кроме последней. Написать программу и диаграмму потоков данных.
2. Сведения об ученике состоят из его фамилии и названия класса (года обучения и буквы), в котором он учится. Выяснить, имеются ли однофамильцы в параллельных классах. Результаты вывести в табличной форме. Написать программу и функциональную диаграмму.

Задание 10

1. Дана действительная квадратная матрица порядка n . Преобразовать матрицу по правилу: строку с номером n сделать столбцом с номером n , а столбец с номером n – строкой с номером n . Написать программу и диаграмму потоков данных.
2. Сведения о книге состоят из фамилии автора, названия книги, наименования издательства и года издания. Вывести на экран (в табличной форме) информацию о книгах, выпущенных издательством “Наука” начиная с 1995 г. Написать программу и функциональную диаграмму.

Задание 11

1. В данной действительной квадратной матрице порядка n найти наибольший по модулю элемент. Получить квадратную матрицу порядка $n - 1$ путем отбрасывания из исходной матрицы строки и столбца, на пересечении которых расположен элемент с найденным значением. Написать программу и диаграмму потоков данных.
2. Словарная статья включает английское слово, часть речи и русский перевод. Вывести информацию о словах, начинающихся с букв ‘a’, ‘b’, ‘c’ (без учета регистра). Написать программу и функциональную диаграмму.

Задание 12

1. В данной действительной квадратной матрице порядка n найти сумму элементов строки, в которой расположен элемент с наименьшим значением. Предполагается, что такой элемент единственный. Написать программу и диаграмму потоков данных.
2. Некоторый массив содержит данные о кубиках: размер каждого кубика (длина ребра в сантиметрах), его номер и цвет. Упорядочить кубики по возрастанию объема. Вычислить суммарный объем синих кубиков. Результаты вывести на экран в табличной форме. Написать программу и функциональную диаграмму.

Задание 13

1. Дана прямоугольная матрица. Найти строку с наибольшей и наименьшей суммой элементов. Вывести на печать найденные строки и суммы их элементов. Написать программу и диаграмму потоков данных.
2. Данные о пассажирах включают фамилию пассажира, номер места и общий вес багажа. Найти пассажиров, имеющих багаж с минимальным и максимальным весом и вывести данные о них в табличной форме. Написать программу и функциональную диаграмму.

Задание 14

1. Найти наибольший и наименьший элементы прямоугольной матрицы и поменять их местами. Написать программу и диаграмму потоков данных.
2. Данные о пассажирах включают фамилию пассажира, номер места и вес багажа. Вывести в табличной форме данные о пассажирах, занимающих места с 1-го по 10-е. Написать программу и функциональную диаграмму.

Задание 15

1. Для целочисленной квадратной матрицы найти число элементов, кратных k , и наибольший из полученных результатов. Написать программу и диаграмму потоков данных.
2. В железнодорожных кассах хранится информация о свободных местах в поездах по всем направлениям на ближайшую неделю. Данная информация представлена в следующем виде: дата, конечный пункт назначения, время, число свободных мест. Вывести данные о поездах, отправляющихся из Барнаула с 10 до 17 часов. Написать программу и функциональную диаграмму.

Задание 16

1. Задана матрица порядка n и число k . Разделить элементы k -ой строки на диагональный элемент, расположенный в этой строке. Написать программу и диаграмму потоков данных.
2. В риэлтерской фирме имеются следующие данные о квартирах: адрес, этаж и общая площадь. Вывести данные о квартирах, площадь которых больше заданной и расположенных ниже заданного этажа. Написать программу и функциональную диаграмму.

Задание 17

1. Квадратная матрица, симметричная относительно главной диагонали, задана верхним треугольником в виде одномерного массива. Восстановить исходную матрицу и напечатать по строкам. Написать программу и диаграмму потоков данных.
2. Данные об экспортируемых товарах включают наименование товара, страну, импортирующую товар и объем поставляемой партии в штуках. Найти страны, в которые экспортируется данный товар, и общий объем его экспорта. Написать программу и функциональную диаграмму.

Задание 18

1. Задана квадратная матрица. Получить транспонированную матрицу, т.е. матрицу, где столбцы и строки меняются местами. Написать программу и диаграмму потоков данных.
2. На кафедре имеется следующая информация о студентах перед экзаменами: ФИО, общий рейтинг и номер зачетки. Вывести на экран (в табличной форме) данные о студентах, рейтинг которых меньше заданного. Написать программу и функциональную диаграмму.

Задание 19

1. Для заданной квадратной матрицы сформировать одномерный массив из ее диагональных элементов. Найти след матрицы, суммируя элементы одномерного массива. Преобразовать исходную матрицу по правилу: четные строки разделить на полученное значение, нечетные оставить без изменения. Написать программу и диаграмму потоков данных.

2. В театральной кассе имеется следующая информация о спектаклях: дата, время, названия спектакля. Выяснить, когда в театре идет спектакль «Гамлет» (вывести дату и время). Написать программу и функциональную диаграмму.

Задание 20

1. Для заданной квадратной матрицы сформировать одномерный массив из ее диагональных элементов. Найти след матрицы, суммируя элементы одномерного массива. Преобразовать исходную матрицу по правилу: четные строки разделить на полученное значение, нечетные оставить без изменения. Написать программу и диаграмму потоков данных.
2. В кассе “Театра драмы” имеется следующая информация о спектаклях: дата, время, название, стоимость билета. Вывести на экран (в табличной форме) программу театра на 26 апреля. Написать программу и функциональную диаграмму.

Задание 21

1. Дана действительная квадратная матрица порядка N (N – нечетное), все элементы которой различны. Найти наибольший элемент среди стоящих на главной и побочной диагоналях и поменять его местами с элементом, стоящим на пересечении этих диагоналей. Написать программу и диаграмму потоков данных.
2. Данные об авиарейсах включают номер рейса, пункт отправления, пункт назначения, дату и время. Вывести на экран информацию об авиарейсах, выполняющихся из г. Барнаула по четным числам. Написать программу и функциональную диаграмму.

Задание 22

1. Дана действительная матрица размером $n \times m$, все элементы которой различны. В каждой строке выбирается элемент с наименьшим значением, затем среди этих чисел выбирается наибольшее. Указать индексы элемента с найденным значением. Написать программу и диаграмму потоков данных.
2. В радиоателье хранятся квитанции о сданной в ремонт радиоаппаратуре. Каждая квитанция содержит следующую информацию: наименование группы изделий (телевизор, радиоприемник и т.п.), марка изделия, дата приемки в ремонт, состояние готовности заказа (выполнен, не выполнен). Необходимо выдать информацию начальнику ателье о состоянии заказов на текущие сутки по группам изделий. Написать программу и функциональную диаграмму.

Задание 23

1. Дана квадратная матрица $A[N, N]$. Записать на место отрицательных элементов матрицы нули, а на место положительных – единицы. Вывести на печать нижнюю треугольную матрицу в общепринятом виде. Написать программу и диаграмму потоков данных.

2. Дана ведомость абитуриентов, сдавших вступительные экзамены в институт. В каждой строке данной ведомости записана фамилия абитуриента и полученные им оценки по отдельным дисциплинам (например, физике, математике, литературе). Необходимо определить средний балл по институту и распечатать список абитуриентов, средний балл которых выше среднего балла по институту. Первыми в списке должны идти студенты сдавшие экзамены на 5. Написать программу и функциональную диаграмму.

Задание 24

1. Задана матрица размером $n \times m$. Найти максимальный по модулю элемент матрицы. Переставить строки и столбцы матрицы таким образом, чтобы максимальный по модулю элемент был расположен на пересечении k -ой строки и k -го столбца. Написать программу и диаграмму потоков данных.
2. В справочной аэропорта хранится расписание вылета самолетов на следующие сутки. Для каждого рейса указаны его номер, тип самолета, пункт назначения, время вылета. Определить все номера рейсов, типы самолетов и время их вылета для заданного пункта назначения. Написать программу и функциональную диаграмму.

Задание 25

1. Упорядочить по возрастанию элементы каждой строки матрицы размером $n \times m$. Написать программу и диаграмму потоков данных.
2. У администратора железнодорожных касс хранится информация о свободных местах в поездах по всем направлениям на ближайшую неделю. Данная информация представлена в следующем виде: дата, конечный пункт назначения, время, число свободных мест. Оргкомитет международной конференции обращается к администратору с просьбой зарезервировать 50 мест до Берлина на 21 июня, при этом время отправления поезда должно быть не позднее 10 часов вечера. Выдать на печать время отправления или сообщение о невозможности выполнить заказ в полном объеме. Написать программу и функциональную диаграмму.

Задание 26

1. Дана целочисленная квадратная матрица. Найти в каждой строке наибольший элемент и поменять его местами с элементом главной диагонали. Написать программу и диаграмму потоков данных.
2. Во время предвыборной компании проводится регистрация кандидатов в депутаты. Каждый кандидат, подавая заявление на регистрацию, указывает номер округа, в котором он собирается баллотироваться, наименование партии, которую он представляет, свой возраст и профессию. Пресс-служба центральной избирательной комиссии выдает информационный бюллетень, в котором приводит следующую информацию: число кандидатов от каждой политической партии. Написать соответствующую программу. Написать программу и функциональную диаграмму.

Задание 27

1. Определить, является ли заданная целая квадратная матрица n -го порядка симметричной (относительно главной диагонали). Написать программу и диаграмму потоков данных.
2. Данные о пассажирах включают фамилию пассажира, номер места и общий вес багажа. Вывести на экран фамилии пассажиров, вес багажа которых меньше заданного. Результаты вывести в табличной форме. Написать программу и функциональную диаграмму.

Задание 28

1. Задана квадратная матрица. Переставить строку с максимальным элементом на главной диагонали со строкой с заданным номером m . Написать программу и диаграмму потоков данных.
2. Во время предвыборной компании проводится регистрация кандидатов в депутаты. Каждый кандидат, подавая заявление на регистрацию, указывает номер округа, в котором он собирается баллотироваться, наименование партии, которую он представляет, свой возраст и профессию. Вывести список кандидатов по заданному наименованию партии и номеру округа. Написать программу и функциональную диаграмму.

Задание 29

1. Дана целая квадратная матрица n -го порядка. Определить, является ли она магическим квадратом, т.е. такой, в которой суммы элементов во всех строках и столбцах одинаковы. Написать программу и диаграмму потоков данных.
2. Сведения о книге состоят из фамилии автора, названия книги, наименования издательства и года издания. Упорядочить данные по фамилии автора. Вывести данные о книгах, изданных заданным автором в 1998 г. Написать программу и функциональную диаграмму.

Задание 30

1. Дана матрица $B[N, M]$. Найти в каждой строке матрицы максимальный и минимальный элементы и поменять их с первым и последним элементом строки соответственно. Написать программу и диаграмму потоков данных.
2. На кафедре имеются следующие данные о методических пособиях: название, фамилии авторов, год издания, цена, количество. Вывести на экран данные о методических пособиях, изданных с 1995 г. Подсчитать общее число таких пособий. Написать программу и функциональную диаграмму.

КРИТЕРИИ ОЦЕНКИ:

- «5»- правильное решение всех заданий, может обосновать решение
- «4» - правильное решение, двух заданий может обосновать решение
- «3» - правильное решение одного задания, может обосновать решение
- «2» - задания не решены, не может обосновать решение

5.ПАКЕТ ЭКЗАМЕНАТОРА

МДК 02.02 Инструментальные средства разработки программного обеспечения

ПАСПОРТ

Назначение:

КОМ предназначен для контроля и оценки результатов освоения ПМ02 02.02 Инструментальные средства разработки программного обеспечения специальности СПО 09.02.07 «Информационные системы и программирование»

Итогом экзамена является пятибалльная оценка.

Проверяемые результаты обучения: ПК2.2 ПК2.3 ПК2.4 ПК2.5

Инструкция

Внимательно прочитайте задание.

Вы можете воспользоваться средой программирования, средствами отладки

Время выполнения задания – 1 час

Теоретические вопросы

1. Понятие репозитория проекта, структура проекта
2. Виды, цели и уровни интеграции программных модулей. Автоматизация бизнес-процессов.
3. Выбор источников и приемников данных, сопоставление объектов данных.
4. Транспортные протоколы. Стандарты форматирования сообщений.
5. Организация работы команды в системе контроля версий.
6. Отладка программных продуктов. Инструменты отладки. Отладочные классы.
7. Ручное и автоматизированное тестирование.
8. Методы и средства организации тестирования.
9. Инструментарии анализа качества программных продуктов в среде разработке.
10. Обработка исключительных ситуаций.
11. Методы и способы идентификации сбоев и ошибок.
12. Выявление ошибок системных компонентов

Практическая часть

Задание 1

1. Матрица $A [M, N]$ (M кратно 4) разделена по вертикали на две половины. Определить сумму элементов каждого столбца левой половины и сумму элементов каждого четного столбца правой половины матрицы A . Написать программу и отчет по тестированию.
2. Определить класс «Матрица» и разработать подпрограммы для ввода, вывода, сложения и вычитания переменных данного типа. Написать программу и отчет по тестированию.

Задание 2

1. Определить номера строк матрицы $R [M, N]$, хотя бы один элемент которых равен s , и элементы этих строк умножить на d . Написать программу и отчет по тестированию.
2. Определить класс «Методическое пособие» (должен включать название, фамилию автора, год издания и цену). Разработать подпрограммы для ввода информации с клавиатуры, вывода на экран, а также чтения/записи в файл. Написать программу и отчет по тестированию.

Задание 3

1. Задана квадратная матрица. Поменять местами строку с максимальным элементом на главной диагонали со строкой с заданным номером m . Написать программу и отчет по тестированию.
2. Объявить класс «Дробное число» и написать подпрограммы для сложения, вычитания, умножения и деления переменных данного типа. Написать программу и отчет по тестированию.

Задание 4

1. Дана матрица $B [N, M]$. Найти в каждой строке матрицы максимальный и минимальный элементы и поменять их местами с первым и последним элементом строки соответственно. Написать программу и отчет по тестированию.
2. Определить класс «Ученик» (должен включать имя, фамилию и название класса, т.е. год обучения и букву). Разработать подпрограммы для ввода информации с клавиатуры (проверять правильность ввода номера класса), вывода на экран, а также чтения/записи в файл. Написать программу и отчет по тестированию.

Задание 5

1. Дана целая квадратная матрица n -го порядка. Определить, является ли она магическим квадратом, т.е. такой, в которой суммы элементов во всех строках и столбцах одинаковы. Написать программу и отчет по тестированию.
2. Определить класс «Одномерный массив» и разработать подпрограммы для ввода, вывода, поиска максимального/минимального элементов. Написать программу и отчет по тестированию.

Задание 6

1. Упорядочить по возрастанию элементы каждой строки матрицы размером $n \times m$. Написать программу и отчет по тестированию.
2. Определить класс «Матрица» и разработать подпрограммы для ввода, вывода, умножения матрицы на число, а также умножения строки/столбца матрицы (с заданным номером) на число. Написать программу и отчет по тестированию.

Задание 7

1. Дана матрица размером $n \times m$. Переставляя ее строки и столбцы, добиться того, чтобы наибольший элемент (или один из них) оказался в верхнем левом углу. Написать программу и отчет по тестированию.
2. Определить класс «Автомобиль» (должен включать марку, номер и фамилию владельца). Разработать подпрограммы для ввода информации с клавиатуры, вывода на экран, а также чтения/записи в файл. Написать программу и отчет по тестированию.

Задание 8

1. Определить, является ли заданная целая квадратная матрица n -го порядка симметричной (относительно главной диагонали). Написать программу и отчет по тестированию.
2. Определить класс «Очередь» и разработать подпрограммы для добавления элемента (в конец очереди) и удаления элемента (из ее начала). Реализовать очередь с использованием массивов. Написать программу и отчет по тестированию.

Задание 9

1. Дана целочисленная квадратная матрица. Найти в каждой строке наибольший элемент и поменять его местами с элементами главной диагонали. Написать программу и отчет по тестированию.
2. Определить класс «Спектакль» (должен включать дату, время, название спектакля). Разработать подпрограммы для ввода информации с клавиатуры, вывода на экран, а также чтения/записи в файл. Написать программу и отчет по тестированию.

Задание 10

1. Упорядочить по возрастанию элементы каждой строки матрицы размером $n \times m$. Написать программу и отчет по тестированию.
2. Определить класс «Книга» (должен включать номер книги, фамилию автора, название, общее количество экземпляров и количество экземпляров на руках). Написать подпрограммы для ввода/вывода информации о книге (кроме перечисленных полей, на экран выводить и количество оставшихся экземпляров). Разработать подпрограмму «Выдача книги» (должна увеличивать на единицу количество экземпляров на руках). Если общее количество экземпляров равно 1, то при вызове данная функция должна выводить сообщение «Книга <название> выдается только в читальный зал». Написать программу и отчет по тестированию.

Задание 11

1. Задана матрица размером $n \times m$. Найти максимальный по модулю элемент матрицы. Переставить строки и столбцы матрицы таким образом, чтобы максимальный по модулю элемент был расположен на пересечении k -й строки и k -го столбца. Написать программу и отчет по тестированию.
2. Определить класс «Одномерный массив» и разработать подпрограммы для ввода, вывода массива, а также вставки и удаления элемента с заданным номером. Написать программу и отчет по тестированию.

Задание 12

1. Дана квадратная матрица $A[N,N]$. Записать на место отрицательных элементов матрицы нули, а на место положительных – единицы. Вывести на печать нижнюю треугольную матрицу в общепринятом виде. Написать программу и отчет по тестированию.
2. Определить класс «Абитуриент» (должен включать ФИО абитуриента и полученные им оценки по трем дисциплинам). Разработать подпрограммы для ввода информации с клавиатуры, вывода на экран, вычисления среднего балла, а также чтения/записи в файл. Написать программу и отчет по тестированию.

Задание 13

1. Дана действительная матрица размером $n \times m$, все элементы которого различны. В каждой строке выбирается элемент с наименьшим значением, затем среди этих чисел выбирается наибольшее. Указать индексы элемента с найденным значением. Написать программу и отчет по тестированию.
2. Определить класс «Книга» (должен включать название, фамилию автора, год издания и цену). Разработать подпрограммы для ввода информации с клавиатуры, вывода на экран, а также чтения/записи в файл. Написать программу и отчет по тестированию.

Задание 14

1. Дана действительная квадратная матрица порядка N (N – нечетное), все элементы которой различны. Найти наибольший элемент среди стоящих на главной и побочной диагоналях и поменять его местами с элементом, стоящим на пересечении этих диагоналей. Написать программу и отчет по тестированию.
2. Определить класс «Словарная статья» (включает английское слово, часть речи и русский перевод). Разработать подпрограммы для ввода информации с клавиатуры, вывода на экран, а также чтения/записи в файл. Написать программу и отчет по тестированию.

Задание 15

1. Для заданной квадратной матрицы сформировать одномерный массив из ее диагональных элементов. Найти след матрицы, суммируя элементы одномерного массива. Преобразовать исходную матрицу по правилу: четные строки разделить на полученное значение, нечетные оставить без изменения. Написать программу и отчет по тестированию.
2. Определить класс «Пассажир» (должен включать фамилию, номер места и общий вес багажа). Разработать подпрограммы для ввода информации с клавиатуры (проверять корректность ввода номера места и веса багажа), вывода на экран, а также чтения/записи в файл. Написать программу и отчет по тестированию.

Задание 16

1. Задана квадратная матрица. Получить транспонированную матрицу. Написать программу и отчет по тестированию.
2. Определить класс «Студент» (должен включать ФИО, аттестационные оценки по пяти дисциплинам и номер зачетки). Разработать подпрограммы для ввода информации с клавиатуры, вывода на экран, чтения/записи в файл, а также для вычисления и вывода общего рейтинга студента. Написать программу и отчет по тестированию.

Задание 17

1. Квадратная матрица, симметричная относительно главной диагонали, задана верхним треугольником в виде одномерного массива. Восстановить исходную матрицу и напечатать по строкам. Написать программу и отчет по тестированию.
2. Объявить класс «Дробное число» и написать подпрограммы для сложения, вычитания, умножения и деления переменных данного типа. Написать программу и отчет по тестированию.

Задание 18

1. Задана матрица порядка n и число k . Разделить элементы k -й строки на диагональный элемент, расположенный в этой строке. Написать программу и отчет по тестированию.
2. Определить класс «Ученик» (должен включать имя, фамилию и название класса, т.е. год обучения и букву). Разработать подпрограммы для ввода информации с клавиатуры (проверять правильность ввода номера класса), вывода на экран, а также чтения/записи в файл. Написать программу и отчет по тестированию.

Задание 19

1. Для целочисленной квадратной матрицы найти число элементов, кратных k , и наибольший из них. Написать программу и отчет по тестированию.
2. Определить класс «Одномерный массив» и разработать подпрограммы для ввода, вывода, поиска максимального/минимального элементов. Написать программу и отчет по тестированию.

Задание 20

1. Найти наибольший и наименьший элементы прямоугольной матрицы и поменять их местами. Написать программу и отчет по тестированию.
2. Определить класс «Матрица» и разработать подпрограммы для ввода, вывода, умножения матрицы на число, а также умножения строки/столбца матрицы (с заданным номером) на число. Написать программу и отчет по тестированию.

Задание 21

1. Дана прямоугольная матрица. Найти строку с наибольшей и наименьшей суммой элементов. Вывести на печать найденные строки и суммы их элементов. Написать программу и отчет по тестированию.
2. Определить класс «Автомобиль» (должен включать марку, номер и фамилию владельца). Разработать подпрограммы для ввода информации с клавиатуры, вывода на экран, а также чтения/записи в файл. Написать программу и отчет по тестированию.

Задание 22

1. В данной действительной квадратной матрице порядка n найти сумму элементов строки, в которой расположен элемент с наименьшим значением. Предполагается, что такой элемент единствен. Написать программу и отчет по тестированию.
2. Определить класс «Очередь» и разработать подпрограммы для добавления элемента (в конец очереди) и удаления элемента (из ее начала). Реализовать очередь с использованием массивов. Написать программу и отчет по тестированию.

Задание 23

1. В данной действительной квадратной матрице порядка n найти наибольший по модулю элемент. Получить квадратную матрицу порядка $n - 1$ путем отбрасывания в исходной матрице строки и столбца, на пересечении которых расположен элемент с найденным значением. Написать программу и отчет по тестированию.
2. Определить класс «Спектакль» (должен включать дату, время, название спектакля). Разработать подпрограммы для ввода информации с клавиатуры, вывода на экран, а также чтения/записи в файл. Написать программу и отчет по тестированию.

Задание 24

1. Дана действительная квадратная матрица порядка n . Преобразовать матрицу по следующему правилу: строку с номером n сделать столбцом с номером n , а столбец с номером n – строкой с номером n . Написать программу и отчет по тестированию.

2. Определить класс «Книга» (должен включать номер книги, фамилию автора, название, общее количество экземпляров и количество экземпляров на руках). Написать подпрограммы для ввода/вывода информации о книге (кроме перечисленных полей, на экран выводить и количество оставшихся экземпляров). Разработать подпрограмму «Выдача книги» (должна увеличивать на единицу количество экземпляров на руках). Если общее количество экземпляров равно 1, то при вызове данная функция должна выводить сообщение «Книга <название> выдается только в читальный зал». Написать программу и отчет по тестированию.

Задание 25

1. Пусть действительная матрица размером $n \times m$. Требуется преобразовать матрицу следующим образом: поэлементно вычесть последнюю строку из всех строк, кроме последней. Написать программу и отчет по тестированию.
2. Определить класс «Одномерный массив» и разработать подпрограммы для ввода, вывода массива, а также вставки и удаления элемента с заданным номером. Написать программу и отчет по тестированию.

Задание 26

1. Определить номера тех строк целочисленной матрицы $A[N,K]$, которые совпадают с массивом $D[K]$. Если таких строк нет, выдать соответствующее сообщение. Написать программу и отчет по тестированию.
2. Определить класс «Абитуриент» (должен включать ФИО абитуриента и полученные им оценки по трем дисциплинам). Разработать подпрограммы для ввода информации с клавиатуры, вывода на экран, вычисления среднего балла, а также чтения/записи в файл. Написать программу и отчет по тестированию.

Задание 27

1. Определить наименьший элемент каждой четной строки матрицы $A[M,N]$. Написать программу и отчет по тестированию.
2. Определить класс «Книга» (должен включать название, фамилию автора, год издания и цену). Разработать подпрограммы для ввода информации с клавиатуры, вывода на экран, а также чтения/записи в файл. Написать программу и отчет по тестированию.

Задание 28

1. Дана прямоугольная матрица. Найти столбец с наибольшей и наименьшей суммой элементов. Вывести на печать найденные столбцы и суммы их элементов. Написать программу и отчет по тестированию.
2. Определить класс «Словарная статья» (включает английское слово, часть речи и русский перевод). Разработать подпрограммы для ввода информации с клавиатуры, вывода на экран, а также чтения/записи в файл. Написать программу и отчет по тестированию.

Задание 29

1. Дана матрица A размером $n \times m$. Определить k – количество элементов величина которых находится между максимальным и минимальным значением матрицы. Написать программу и отчет по тестированию.

2. Определить класс «Пассажир» (должен включать фамилию, номер места и общий вес багажа). Разработать подпрограммы для ввода информации с клавиатуры (проверять корректность ввода номера места и веса багажа), вывода на экран, а также чтения/записи в файл. Написать программу и отчет по тестированию.

Задание 30

1. Вычислить сумму и число положительных элементов матрицы $A[N,N]$, находящихся над главной диагональю. Написать программу и отчет по тестированию.
2. Определить класс «Студент» (должен включать ФИО, аттестационные оценки по пяти дисциплинам и номер зачетки). Разработать подпрограммы для ввода информации с клавиатуры, вывода на экран, чтения/записи в файл, а также для вычисления и вывода общего рейтинга студента. Написать программу и отчет по тестированию.

КРИТЕРИИ ОЦЕНКИ:

- «5»- правильное решение всех заданий, может обосновать решение
- «4» - правильное решение, двух заданий может обосновать решение
- «3» - правильное решение одного задания, может обосновать решение
- «2» - задания не решены, не может обосновать решение

Материал для дифференцированного зачета ПМ03 МДК.2.3 Математическое моделирование

Назначение:

КОМ предназначен для контроля и оценки результатов освоения ПМ03 03.01 Технология разработки программного обеспечения специальности СПО 09.02.07 «Информационные системы и программирование»

Итогом дифференцированного зачета является пятибалльная оценка.

Проверяемые результаты обучения: ПК2.2 ПК2.3 ПК2.4 ПК2.5

Инструкция

К дифференцированному зачету допускаются студенты, выполнившие все практические и лабораторные работы

Время выполнения задания: 1ч 30 мин.

Итоговая отметка за семестр выставляется с учетом выполнения дифференцированного зачета текущей успеваемости и РК.

Задание

Вариант № 1

- 1) Решить ЗЛП графическим способом:

$$\begin{aligned} F(x) &= 5x_1 + 3x_2 \rightarrow \max \\ 3x_1 + 5x_2 &\leq 15 \\ 5x_1 + 2x_2 &\leq 10 \\ x_1 &\geq 0, \quad x_2 \geq 0 \end{aligned}$$

- 2) Исходя из общего правила, составить задачу, двойственную к исходной:

$$\begin{cases} 2x_1 + 3x_2 \leq 9 \\ 2x_1 + x_2 \leq 8 \\ x_1, x_2 \geq 0 \end{cases}$$

- 3) Составить начальный опорный план транспортной задачи методом северо-западного угла, найти оптимальный план перевозок путем прямого построения циклов:

$$\|C_{ij}\| = \begin{array}{ccccc} & & & & a_i \\ & 5 & 2 & 4 & 1 & 70 \\ & 2 & 8 & 1 & 5 & 40 \\ & 3 & 2 & 4 & 4 & 20 \\ b_j & 40 & 25 & 35 & 20 \end{array}$$

- 4) Определить критический путь, ранние и поздние сроки наступления событий:

(1,2)	(1,3)	(2,4)	(3,4)	(3,5)	(4,6)	(5,6)
9	4	6	8	10	11	10

Вариант №2

- 1) Решить ЗЛП графическим способом:

$$\begin{aligned} F(x) &= 2x_1 + 3x_2 \rightarrow \max \\ x_1 + x_2 &\leq 2 \\ x_1 + x_2 &\geq -1 \\ 3x_1 + 2x_2 &\leq 10 \quad x_1 \geq 0, \quad x_2 \geq 0 \end{aligned}$$

- 2) Исходя из общего правила, составить задачу, двойственную к исходной:

$$\begin{cases} 2x_1 + 5x_2 \leq 12 \\ 4x_1 + 3x_2 \leq 12 \\ x_1, x_2 \geq 0 \end{cases}$$

- 3) Решить симплекс-методом задачу линейного программирования

$$F(x) = 2x_1 - 4x_2 \rightarrow \max$$

$$\begin{cases} 8x_1 - 5x_2 \leq 16 \\ x_1 + 3x_2 \leq 2 \\ 2x_1 + 7x_2 \leq 9 \\ x_1 \geq 0, \quad x_2 \geq 0 \end{cases}$$

- 4) Построить сеть методом динамического программирования, найти на сети наиболее экономный маршрут.

(1,2)	(1,3)	(1,4)	(2,4)	(3,4)	(4,5)	(3,5)
18	14	9	6	11	4	10

КРИТЕРИИ ОЦЕНКИ:

- «5» - правильное решение всех заданий, может обосновать решение
- «4» - правильное решение, двух заданий может обосновать решение
- «3» - правильное решение одного задания, может обосновать решение
- «2» - задания не решены, не может обосновать решение

ПАКЕТ ЭКЗАМЕНАТОРА: экзамен по профессиональному модулю

1. ПАСПОРТ

Назначение: предназначен для контроля и оценки результатов освоения профессионального модуля ПМ.02 ОСУЩЕСТВЛЕНИЕ ИНТЕГРАЦИИ ПРОГРАММНЫХ специальностей СПО 09.02.07 «Информационные системы и программирование»

2. ЗАДАНИЕ ДЛЯ ЭКЗАМЕНУЮЩЕГОСЯ Вариант №__

Инструкция

Внимательно прочитайте задание.

Вы можете воспользоваться средой программирования, BPWin

Время выполнения задания – 1,5 часа

Коды проверяемых профессиональных и общих компетенций:

ПК 2.1, ПК 2.2, ПК 2.4

Задание

1. Разработать модель программной системы
2. Разработать техническое задание
3. Написать программу.
4. Оформить отчет по тестированию

Сведения об автомобиле состоят из его марки, номера и фамилии владельца. Найти фамилии владельцев и номера автомобилей данной марки. Результаты вывести в табличной форме. Написать программу и функциональную диаграмму.

ПАКЕТ ЭКЗАМЕНАТОРА ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ

Задание 1

Сведения об ученике состоят из его имени, фамилии и названия класса (года обучения и буквы), в котором он учится. Выяснить, имеются ли в школе однофамильцы. Результаты вывести в табличной форме. Написать программу.

Задание 2

Сведения об автомобиле состоят из его марки, номера и фамилии владельца. Найти фамилии владельцев и номера автомобилей данной марки. Результаты вывести в табличной форме. Написать программу.

Задание 3

Некоторый массив содержит различные даты. Каждая дата — это число, номер месяца и год. Найти и вывести на экран самую позднюю дату. Результат вывести в табличной форме. Написать программу.

Задание 4

Некоторый массив содержит данные о кубиках: размер каждого кубика (длина ребра в сантиметрах), его цвет (красный, желтый, зеленый или синий) и материал (деревянный, металлический, картонный). Найти количество кубиков каждого их перечисленных цветов и их суммарный объем. Результаты вывести в табличной форме. Написать программу.

Задание 5

Некоторый массив содержит различные даты. Каждая дата — это число, название месяца и год. Найти и вывести на экран все весенние даты. Написать программу.

Задание 6

Сведения о книгах включают фамилию автора, название и год издания. Найти названия книг данного автора, изданных с 1980 г. Результаты вывести в табличной форме. Написать программу.

Задание 7

Сведения о книгах — это фамилия автора, название и год издания. Определить, имеется ли книга с названием «Информатика». Если да, то вывести фамилию автора и год издания. Если таких книг несколько, то вывести информацию обо всех этих книгах в табличной форме. Написать программу.

Задание 8

Словарная статья включает английское слово, часть речи и русский перевод. Упорядочить словарь по английскому алфавиту. Результаты вывести в табличной форме. Написать программу.

Задание 9

Сведения об ученике состоят из его фамилии и названия класса (года обучения и буквы), в котором он учится. Выяснить, имеются ли однофамильцы в параллельных классах. Результаты вывести в табличной форме.

Задание 10

Сведения о книге состоят из фамилии автора, названия книги, наименования издательства и года издания. Вывести на экран (в табличной форме) информацию о книгах, выпущенных издательством “Наука” начиная с 1995 г. Написать программу.

Задание 11

Словарная статья включает английское слово, часть речи и русский перевод. Вывести информацию о словах, начинающихся с букв ‘a’, ‘b’, ‘c’ (без учета регистра). Написать программу.

Задание 12

Некоторый массив содержит данные о кубиках: размер каждого кубика (длина ребра в сантиметрах), его номер и цвет. Упорядочить кубики по возрастанию объема. Вычислить суммарный объем синих кубиков. Результаты вывести на экран в табличной форме. Написать программу.

Задание 13

Данные о пассажирах включают фамилию пассажира, номер места и общий вес багажа. Найти пассажиров, имеющих багаж с минимальным и максимальным весом и вывести данные о них в табличной форме. Написать программу.

Задание 14

Данные о пассажирах включают фамилию пассажира, номер места и вес багажа. Вывести в табличной форме данные о пассажирах, занимающих места с 1-го по 10-е. Написать программу.

Задание 15

В железнодорожных кассах хранится информация о свободных местах в поездах по всем направлениям на ближайшую неделю. Данная информация представлена в следующем виде: дата, конечный пункт назначения, время, число свободных мест. Вывести данные о поездах, отправляющихся из Барнаула с 10 до 17 часов. Написать программу.

Задание 16

В риэлтерской фирме имеются следующие данные о квартирах: адрес, этаж и общая площадь. Вывести данные о квартирах, площадь которых больше заданной и расположенных ниже заданного этажа. Написать программу.

Задание 17

Данные об экспортируемых товарах включают наименование товара, страну, импортирующую товар и объем поставляемой партии в штуках. Найти страны, в которые экспортируется данный товар, и общий объем его экспорта. Написать программу.

Задание 18

На кафедре имеется следующая информация о студентах перед экзаменами: ФИО, общий рейтинг и номер зачетки. Вывести на экран (в табличной форме) данные о студентах, рейтинг которых меньше заданного. Написать программу.

Задание 19

В театральной кассе имеется следующая информация о спектаклях: дата, время, названия спектакля. Выяснить, когда в театре идет спектакль “Гамлет” (вывести дату и время). Написать программу.

Задание 20

В кассе “Театра драмы” имеется следующая информация о спектаклях: дата, время, название, стоимость билета. Вывести на экран (в табличной форме) программу театра на 26 апреля. Написать программу.

Задание 21

Данные об авиарейсах включают номер рейса, пункт отправления, пункт назначения, дату и время. Вывести на экран информацию об авиарейсах, выполняющихся из г. Барнаула по четным числам. Написать программу.

Задание 22

В радиоателье хранятся квитанции о сданной в ремонт радиоаппаратуре. Каждая квитанция содержит следующую информацию: наименование группы изделий (телевизор, радиоприемник и т.п.), марка изделия, дата приемки в ремонт, состояние готовности заказа (выполнен, не выполнен). Необходимо выдать информацию начальнику ателье о состоянии заказов на текущие сутки по группам изделий. Написать программу.

Задание 23

Дана ведомость абитуриентов, сдавших вступительные экзамены в институт. В каждой строке данной ведомости записана фамилия абитуриента и полученные им оценки по отдельным дисциплинам (например, физике, математике, литературе). Необходимо определить средний балл по институту и распечатать список абитуриентов, средний балл которых выше среднего балла по институту. Первыми в списке должны идти студенты сдавшие экзамены на 5. Написать программу.

Задание 24

В справочной аэропорта хранится расписание вылета самолетов на следующие сутки. Для каждого рейса указаны его номер, тип самолета, пункт назначения, время вылета. Определить все номера рейсов, типы самолетов и время их вылета для заданного пункта назначения. Написать программу.

Задание 25

У администратора железнодорожных касс хранится информация о свободных местах в поездах по всем направлениям на ближайшую неделю. Данная информация представлена в следующем виде: дата, конечный пункт назначения, время, число свободных мест. Оргкомитет международной конференции обращается к администратору с просьбой зарезервировать 50 мест до Берлина на 21 июня, при этом время отправления поезда должно быть не позднее 10 часов вечера. Выдать на печать время отправления или сообщение о невозможности выполнить заказ в полном объеме. Написать программу.

Задание 26

Во время предвыборной компании проводится регистрация кандидатов в депутаты. Каждый кандидат, подавая заявление на регистрацию, указывает номер округа, в котором он собирается баллотироваться, наименование партии, которую он представляет, свой возраст и профессию. Пресс-служба центральной избирательной комиссии выдает информационный бюллетень, в котором приводит следующую информацию: число кандидатов от каждой политической партии. Написать соответствующую программу. Написать программу.

Задание 27

Данные о пассажирах включают фамилию пассажира, номер места и общий вес багажа. Вывести на экран фамилии пассажиров, вес багажа которых меньше заданного. Результаты вывести в табличной форме. Написать программу.

Задание 28

Во время предвыборной компании проводится регистрация кандидатов в депутаты. Каждый кандидат, подавая заявление на регистрацию, указывает номер округа, в котором он собирается баллотироваться, наименование партии, которую он представляет, свой возраст и профессию. Вывести список кандидатов по заданному наименованию партии и номеру округа. Написать программу.

Задание 29

Сведения о книге состоят из фамилии автора, названия книги, наименования издательства и года издания. Упорядочить данные по фамилии автора. Вывести данные о книгах, изданных заданным автором в 1998 г. Написать программу.

Задание 30

На кафедре имеются следующие данные о методических пособиях: название, фамилии авторов, год издания, цена, количество. Вывести на экран данные о методических пособиях, изданных с 1995 г. Подсчитать общее число таких пособий. Написать программу.

Литература:

Печатные издания

1. Рудаков А.В. Технология разработки программных продуктов ,Москва, Издательский центр «Академия» ,207 стр. 2015г.

Электронные издания (электронные ресурсы)

1. От модели объектов - к модели классов. Единое окно доступа к образовательным ресурсам. http://real.tepkom.ru/Real_OM-CM_A.asp

Дополнительные источники

- 1 Гагарина, Л.Г. Технология разработки программного обеспечения: учеб. пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Виснадул; Под ред. Л.Г. Гагариной. - М.: ФОРУМ: ИНФРА-М, 2017.-400 с.
- 2 Орлов С.А. «Технологии разработки программного обеспечения» Питер, 2015 г. Электронное пособие
- 3 Бобровский С. «Delphi7. Учебный курс» Питер, 2015г.
- 4 Леффингуал, Дин, Ундри, Дон Принципы работы с требованиями к ПО. Унифицированный подход. М., Электронное пособие,2004г.
- 5 Сэм Канер и др. Тестирования программного обеспечения. Электронное пособие, Киев, 2000 г.
- 6 А. Якобсон, Г. Буч, Дж. Рамбо Унифицированный процесс разработки ПО, Электронное пособие, М., 2000 г.
- 7 Крэг Ларман Применение UML и шаблонов проектирования. Электронное пособие, М., Вильямс, 2000 г.
- 8 Шниер Толковый словарь компьютерных технологий. Электронное пособие М., 2000 г.
- 9 Терри Кватрани, Джим Палистрант Визуальное моделирование с помощью IBM Rational Sostware Architect и UML. М.: КУДИЦ-ПРЕСС, 2007 г.
- 10 Луиза Тампе Введение в тестирование программного обеспечения. М.: Вильямс, 2005 г.
- 11 Иан Соммервилл Инженерия программного обеспечения. М.: Вильямс, 2004 г.
- 12 Информатика: Учебник. – 3-е изд. перераб., / Под ред. Н.В. Макаровой. – М.: Финансы и статистика, 2004.
- 13 Жоголев Е.А. Введение в технологию программирования: Конспект лекций. – М.: ДИАЛОГ-МГУ, 2005.
- 14 Вендров А.М. Практикум по проектированию программного обеспечения экономических информационных систем. – М.: Финансы и статистика, 2006.
- 15 Партыка Т.Л., Попов И.И. Информационная безопасность. – М.: Форум: ИНФРА-М, 2006.
- 16 Ван-Тассел Д. Стил, разработка, эффективность, отладка и испытание программ. – М.: Мир, 2005. – 578 с.
- 17 Шураков В.В. Надёжность программного обеспечения. – М.: Финансы и статистика,2003. – 320 с.
- 18 Бутаков Е.А. Методы создания качественного программного обеспечения ЭВМ. – М.: Энергоатомиздат,2003. – 457 с.

6. Сводная таблица по ПМ

Элемент учебной дисциплины, ПМ	Текущий и рубежный контроль		Промежуточная аттестация по ПМ		Экзамен (квалификационный)	
	Форма контроля	Проверяемые ОК, ПК	Форма контроля	Проверяемые ОК, ПК	Форма контроля	Проверяемые ОК, ПК
Раздел 1. Разработка программного обеспечения						
МДК. 2.1 Технология разработки программного обеспечения						
Тема 2.1.1 Основные понятия и стандартизация требований к программному обеспечению	Решение ситуационных задач, Защита ЛПЗ	ОК 1–ОК 11, ПК2.1, ПК2.5	Экзамен	ОК 1–ОК 11, ПК2.1, ПК2.5	Устное обоснование результатов работы, Подготовленный продукт / осуществленный процесс	ОК 1–ОК 11, ПК2.1, ПК2.5
Тема 2.1.2. Описание и анализ требований. Диаграммы IDEF	Решение ситуационных задач, Защита ЛПЗ	ОК 1–ОК 11, ПК2.1, ПК2.5	Экзамен	ОК 1–ОК 11, ПК2.1, ПК2.5	Устное обоснование результатов работы, Подготовленный продукт / осуществленный процесс	ОК 1–ОК 11, ПК2.1, ПК2.5
Тема 2.1.3. Оценка качества программных средств	Решение ситуационных задач, Защита ЛПЗ	ОК 1–ОК 11, ПК2.1, ПК2.5	Экзамен	ОК 1–ОК 11, ПК2.1, ПК2.5	Устное обоснование результатов работы, Подготовленный продукт / осуществленный процесс	ОК 1–ОК 11, ПК2.1, ПК2.5
Курсовое проектирование	Написание пояснительной записки	ОК 1–ОК 11, ПК2.1, ПК2.5	Защита курсового проекта	ОК 1–ОК 11, ПК2.1, ПК2.5	Устное обоснование результатов работы, Подготовленный продукт / осуществленный процесс	ОК 1–ОК 11, ПК2.1, ПК2.5

Раздел 2. Средства разработки программного обеспечения						
МДК.2.2 Инструментальные средства разработки программного обеспечения						
Тема 2.2.1 Современные технологии и инструменты интеграции.	Решение ситуационных задач,	ОК 1–ОК 11, ПК2.1, ПК2.5	Экзамен	ОК 1–ОК 11, ПК2.1, ПК2.5	Устное обоснование результатов работы, Подготовленный продукт / осуществленный процесс	ОК 1–ОК 11, ПК2.1, ПК2.5
Тема 2.2.2 Инструментарий тестирования и анализа качества программных средств	Защита ЛПЗ	ОК 1–ОК 11, ПК2.1, ПК2.5	Экзамен	ОК 1–ОК 11, ПК2.1, ПК2.5	Устное обоснование результатов работы, Подготовленный продукт / осуществленный процесс	ОК 1–ОК 11, ПК2.1, ПК2.5
Раздел. 3 Моделирование в программных системах						
МДК 2.3 Математическое моделирование						
Тема 2.3.1. Основы моделирования. Детерминированные задачи	Решение ситуационных задач, Защита ЛПЗ	ОК 1–ОК 11, ПК 2.1, ПК 2.4, ПК 2.5	Диф.зачет	ОК 1–ОК 11, ПК 2.1, ПК 2.4, ПК 2.5	Устное обоснование результатов работы, Подготовленный продукт / осуществленный процесс	ОК 1–ОК 11, ПК 2.1, ПК 2.4, ПК 2.5
Тема 2.3.2 Задачи в условиях неопределенности	Решение ситуационных задач, Защита ЛПЗ	ОК 1–ОК 11, ПК 2.1, ПК 2.4, ПК 2.5	Диф.зачет	ОК 1–ОК 11, ПК 2.1, ПК 2.4, ПК 2.5	Устное обоснование результатов работы, Подготовленный продукт / осуществленный процесс	ОК 1–ОК 11, ПК 2.1, ПК 2.4, ПК 2.5