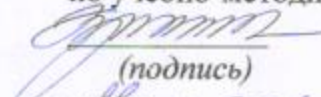


**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ НОВОСИБИРСКОЙ ОБЛАСТИ
«НОВОСИБИРСКИЙ ЭЛЕКТРОМЕХАНИЧЕСКИЙ КОЛЛЕДЖ»
(ГБПОУ НСО «НЭК»)**

УТВЕРЖДАЮ

Заместитель директора
по учебно-методической работе

 Кравченко Е.О.
(подпись) (Ф.И.О.)
«18» июля 2025 г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ
«ПМ.01. Разработка модулей программного обеспечения для
компьютерных систем»**

Специальность 09.02.07. Информационные системы и программирование

Рассмотрено
на заседании кафедры общеобразовательных дисциплин
Протокол № 10 «16» июня 2025 г.

Заведующий кафедрой  Белимова В.Г.
(подпись) (Ф.И.О.)

Оценочный фонд профессионального модуля разработана на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности/профессии 09.02.07. Информационные системы и программирование с учетом примерной образовательной программы и рабочей программы воспитания по специальности 09.02.07. Информационные системы и программирование.

Организация-разработчик: ГБПОУ НСО «НЭК»

Согласовано:
Методист



Рахметова Т.Ю.

«18» июля 2025 г.

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ
профессионального модуля

**ПМ.01. РАЗРАБОТКА МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
 ДЛЯ КОМПЬЮТЕРНЫХ СИСТЕМ**

	Контролируемые разделы, темы, модули	Код контролируемой компетенции	Наименование оценочного средства
МДК.01.02. ПОДДЕРЖКА И ТЕСТИРОВАНИЕ ПРОГРАММНЫХ МОДУЛЕЙ			
1.	Раздел 1. Отладка и тестирование Программного обеспечения	ОК 01; ОК 02; ОК 03; ОК 04; ОК 05; ОК 06; ОК 07; ОК 08; ОК 09; ОК 10; ПК 1.3; ПК 1.4; ПК 1.5.	коллоквиум; тестирование; самостоятельная работа
2.	Раздел 2. Основные принципы отладки и тестирования программных продуктов	ОК 01; ОК 02; ОК 03; ОК 04; ОК 05; ОК 06; ОК 07; ОК 08; ОК 09; ОК 10; ПК 1.3; ПК 1.4; ПК 1.5.	коллоквиум; тестирование; самостоятельная работа
3.	Раздел 3. Модульное тестирование программных продуктов	ОК 01; ОК 02; ОК 03; ОК 04; ОК 05; ОК 06; ОК 07; ОК 08; ОК 09; ОК 10; ПК 1.3; ПК 1.4; ПК 1.5.	коллоквиум; тестирование; самостоятельная работа
МДК.01.03. РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ			
1.	Раздел 1. Основные платформы и языки разработки мобильных приложений	ОК 1; ОК25; ОК3; ОК4; ОК5; ОК6, ОК7, ОК8, ОК9, ОК10. ПК.1.1, ПК 1.6.	коллоквиум; тестирование;
2.	Раздел 2 Создание и тестирование модулей для мобильных приложений	ОК 1; ОК25; ОК3; ОК4; ОК5; ОК6, ОК7, ОК8, ОК9, ОК10. ПК.1.1, ПК 1.6.	коллоквиум; тестирование;
МДК.01.04. СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ			
1.	Раздел 1. Программирование на языке низкого уровня	ОК 01; ОК 02; ОК 03; ОК 04; ОК 05; ОК 06; ОК 07; ОК 08; ОК 09; ОК 10; ПК 1.3; ПК 1.4; ПК 1.5.	коллоквиум; тестирование; самостоятельная работа

2.	Раздел 2. Язык программирования Ассемблер	ОК 01; ОК 02; ОК 03; ОК 04; ОК 05; ОК 06; ОК 07; ОК 08; ОК 09; ОК 10; ПК 1.3; ПК 1.4; ПК 1.5.	КОЛЛОКВИУМ; тестирование; самостоятельная работа
----	--	---	--

Примерный перечень оценочных средств

№ п/п	Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в фонде
1	2	3	4
1.	Коллоквиум	Средство контроля усвоения учебного материала темы, раздела или разделов дисциплины, организованное как учебное занятие в виде собеседования педагогического работника с обучающимися.	Вопросы по темам/разделам дисциплины
2.	Расчетно-графическая работа	Средство проверки умений применять полученные знания по заранее определенной методике для решения задач или заданий по модулю или дисциплине в целом.	Комплект заданий для выполнения расчетно-графической работы
3.	Реферат	Продукт самостоятельной работы студента, представляющий собой краткое изложение в письменном виде полученных результатов теоретического анализа определенной научной (учебно-исследовательской) темы, где автор раскрывает суть исследуемой проблемы, приводит различные точки зрения, а также собственные взгляды на нее.	Темы рефератов
4.	Тест	Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося	Фонд тестовых заданий

КРИТЕРИИ ОЦЕНКИ

МДК.01.02. Поддержка и тестирование программных модулей

Критерии оценки:

Оценка **«отлично»** выставляется, если студент дает полный и правильный ответ на поставленные и дополнительные (если в таковых была необходимость) вопросы:

- обнаруживает всестороннее системное и глубокое знание материала;
- обстоятельно раскрывает соответствующие теоретические положения;
- демонстрирует знание современной учебной и научной литературы;
- владеет понятийным аппаратом;
- демонстрирует способность к анализу и сопоставлению различных подходов к решению заявленной проблематики;
- подтверждает теоретические постулаты примерами из юридической практики;
- способен творчески применять знание теории к решению профессиональных задач;
- имеет собственную оценочную позицию и умеет аргументировано и убедительно ее раскрыть;
- четко излагает материал в логической последовательности.

Оценка **«хорошо»** выставляется, если студент дает ответ, отличающийся меньшей обстоятельностью и глубиной изложения:

- обнаруживает при этом твердое знание материала;
- допускает несущественные ошибки и неточности в изложении теоретического материала; исправленные после дополнительного вопроса;
- опирается при построении ответа только на обязательную литературу;
- подтверждает теоретические постулаты отдельными примерами из юридической практики;
- способен применять знание теории к решению задач профессионального характера;
- наблюдается незначительное нарушение логики изложения материала.

Оценка **«удовлетворительно»** выставляется, если студент в основном знает программный материал в объеме, необходимом для предстоящей работы по профессии, но ответ, отличается недостаточной полнотой и обстоятельностью изложения:

- допускает существенные ошибки и неточности в изложении теоретического материала;
- в целом усвоил основную литературу;
- обнаруживает неумение применять государственно-правовые принципы, закономерности и категории для объяснения конкретных фактов и явлений;
- требуется помощь со стороны (путем наводящих вопросов, небольших разъяснений и т.п.);
- испытывает существенные трудности при определении собственной оценочной позиции;
- наблюдается нарушение логики изложения материала.

Оценка **«неудовлетворительно»** выставляется, если студент обнаруживает незнание или непонимание большей, или наиболее существенной части содержания учебного материала:

- не способен применять знание теории к решению задач профессионального характера;
- не умеет определить собственную оценочную позицию;
- допускает грубое нарушение логики изложения материала;
- допускает принципиальные ошибки в ответе на вопросы;
- не может исправить ошибки с помощью наводящих вопросов.

Вопросы для коллоквиумов, собеседования по дисциплине

«Поддержка и тестирование программных модулей»

1. Концепция тестирования. Тестирование как способ обеспечения качества программного обеспечения.
2. Тестирование, верификация и валидация – различия в понятиях.
3. Цели и задачи процесса тестирования.
4. Жизненный цикл тестирования.
5. Процессы тестирования при разработке программного обеспечения.
6. Техники тестирования требований.
7. Чек-листы, тест – кейсы, наборы тест – кейсов.
8. Анализ и тестирование требований
9. Организация тестирования. Фазы тестирования, основные проблемы тестирования.
10. Тестирование на основе потока управления.
11. Тестирование на основе потока данных. Тестовые сценарии.
12. Критерии выбора тестов. Требования к идеальному критерию тестирования и классы частых критериев.
13. Тестирование программного продукта по структурным критериям
14. Мутационный критерий тестирования.
15. Стохастический критерий тестирования.
16. Тестирование по методу «белого ящика»
17. Тестирование по методу «черного ящика»
18. Модульное и интеграционное тестирование. Особенности модульного тестирования, подходы к тестированию на основе потока управления, потока данных. Организация модульного тестирования.
19. Динамические и статические методы при структурном и объектном подходах
20. Тестирование по функциональному критерию
21. Интеграционное тестирование
22. Взаимосвязь сборки модулей и методов интеграционного тестирования. Подходы монолитного, инкрементального, нисходящего и восходящего тестирования.
23. Оценка тестирования проекта: метрики и методика оценки.
24. Системное тестирование
25. Регрессионное тестирование: особенности и виды регрессионного тестирования
26. Особенности регрессионного тестирования для ООП
27. Документирование процесса тестирования. Тестовые процедуры программного продукта.
28. Документирование и жизненный цикл дефекта.
29. Отчеты о дефектах.
30. Ошибки, дефекты, сбои, отказы.
31. Тест – план.
32. Метрики покрытия требований и метрики покрытия кода.
33. Составление отчета о тестировании.
34. Отчет о дефектах и его жизненный цикл. Атрибуты (поля) отчета о дефектах. Тестовое покрытие

СТРУКТУРА ИТОГОВОГО ТЕСТА:

Тест на платформе Moodle. Результат выдается сразу после тестирования и формируется отчет протестированных студентов.

Время на подготовку и выполнение: Выполнение – 20 минут. За правильный ответ выставляется по

1 баллу, затем результаты суммируются, и выставляется оценка. За неправильный ответ 0 баллов.

Критерии оценки промежуточной аттестации:

Оценка «отлично» выставляется, если имеются все конспекты лекции, обучающимися выполнены 100% практических работ, оценка за итоговое тестирование – «отлично», средний балл по аттестациям не ниже 4,5.

Оценка «хорошо» выставляется, если имеются все конспекты лекции, обучающимися выполнены 100% практических работ, оценка за итоговое тестирование – «хорошо», средний балл по аттестациям не ниже 3,5.

Оценка «удовлетворительно» выставляется, если имеются все конспекты лекции, обучающимися выполнены 100% практических работ, оценка за итоговое тестирование – «удовлетворительно», средний балл по аттестациям не ниже 2,5.

Оценка «неудовлетворительно» выставляется, если имеются все конспекты лекции обучающимися выполнено менее 100% практических работ, оценка за итоговое тестирование – «неудовлетворительно», средний балл по аттестациям ниже 2,5.

Цель итогового тестирования:

Тестирование по учебной дисциплине **«Поддержка и тестирование программных модулей»** предназначено для проверки теоретических знаний и понятийного аппарата, которые лежат в основе профессионального образования и найдут самое широкое применение в будущей профессиональной деятельности учащихся. Обеспечение информационной безопасности автоматизированных систем.

Критерии оценки знаний:

Процент правильных ответов, %	Оценка знаний
90-100	5 «отлично»
75-89	4 «хорошо»
60-74	3 «удовлетворительно»
Менее 60	2 «неудовлетворительно»

Список теоретических заданий для подготовки к итоговому тестированию (ТЗ) по дисциплине «Поддержка и тестирование программных модулей»**Тест 1**

- Какой процесс отвечает на вопрос: «Правильно ли мы делаем продукт?»
 - Валидация
 - Верификация
 - Валидация и верификация
- Какой процесс отвечает на вопрос: «Тот ли продукт мы делаем?»
 - Верификация
 - Валидация
 - Контроль качества (QC)
- Что из перечисленного является методом верификации, а не валидации?
 - Функциональное тестирование
 - Приемочное тестирование (UAT)
 - Статический анализ кода
- На каких этапах жизненного цикла проводится верификация?
 - Только на поздних стадиях, после разработки
 - Только на этапе тестирования
 - На протяжении всего жизненного цикла разработки
- Что является основным инструментом валидации?
 - Обзоры кода (Code Review)
 - Динамическое тестирование
 - Статический анализ

6. В какой модели разработки тестирование является отдельной фазой, следующей после разработки?

- a) Гибкая модель (Agile)
- b) Модель «Водопад» (Waterfall)
- c) Модель «Спираль»

7. Какая из перечисленных задач НЕ входит в основные цели тестировщика (QA-инженера)?

- a) Предотвращение появления дефектов
- b) Написание кода для новых функций
- c) Предоставление информации о качестве продукта

8. Какой процесс является проактивным и направлен на предотвращение дефектов?

- a) Контроль качества (QC)
- b) Обеспечение качества (QA)
- c) Тестирование

9. К чему может привести отсутствие или некачественное тестирование?

- a) Только к финансовым потерям
- b) Только к недовольству пользователей
- c) К финансовым потерям, потере репутации, угрозам безопасности и другим рискам

10. Какой вид тестирования проверяет, как система работает, а не что она делает?

- a) Функциональное тестирование
- b) Нефункциональное тестирование
- c) Приемочное тестирование

11. Какой уровень тестирования проверяет взаимодействие между отдельными модулями программы?

- a) Модульное тестирование (Unit)
- b) Интеграционное тестирование
- c) Системное тестирование

12. Кто обычно проводит модульное (unit) тестирование?

- a) Конечные пользователи
- b) Тестировщики
- c) Разработчики

13. Что такое бета-тестирование?

- a) Тестирование внутри компании командой, не участвовавшей в разработке
- b) Тестирование "живыми" пользователями в реальных условиях
- c) Тестирование отдельных функций программы

14. Какой метод тестирования предполагает полное знание внутренней структуры (кода) приложения?

- a) Тестирование черного ящика
- b) Тестирование белого ящика
- c) Тестирование серого ящика

15. Как называется техника тестирования, которая проверяет поведение системы на границах диапазонов входных данных?

- a) Эквивалентное разбиение
- b) Анализ граничных значений
- c) Покрытие ветвей

16. Какой документ описывает общую стратегию тестирования?

- a) Тест-кейс
- b) Тестовый план (Test Plan)
- c) Баг-репорт

17. Что такое тест-кейс?
- a) Список пунктов для быстрой проверки
 - b) Документ, описывающий найденный дефект
 - c) Пошаговый сценарий для проверки определенной функции
18. Какое поле в баг-репорте описывает, насколько сильно ошибка влияет на работу системы?
- a) Приоритет (Priority)
 - b) Серьезность (Severity)
 - c) Статус (Status)
19. Что является главной целью автоматизации тестирования?
- a) Полная замена ручного тестирования
 - b) Повышение скорости, надежности и эффективности тестирования
 - c) Увеличение количества тестирующих в команде
20. Какой инструмент является самым известным фреймворком для автоматизации тестирования веб-приложений?
- a) Appium
 - b) JUnit
 - c) Selenium

Ответы для проверки:

1. b, 2. b, 3. c, 4. c, 5. b, 6. b, 7. b, 8. b, 9. c, 10. b, 11. b, 12. c, 13. b, 14. b, 15. b, 16. b, 17. c, 18. b, 19. b, 20. c

Тест 2

1. Какой фундаментальный вопрос задает процесс верификации?
- a) Тот ли продукт мы делаем?
 - b) Правильно ли мы делаем продукт?
 - c) Сколько стоит продукт?
2. На каких этапах жизненного цикла разработки ПО (SDLC) проводится верификация?
- a) Только на этапе тестирования
 - b) Только на ранних этапах (анализ, дизайн)
 - c) На всех этапах (анализ, дизайн, кодирование, тестирование)
3. Чем процесс верификации принципиально отличается от валидации по своей сути?
- a) Верификация дороже валидации
 - b) Верификация проверяет соответствие спецификациям, а валидация — ожиданиям пользователя
 - c) Верификация проводится тестирующими, а валидация — разработчиками
4. В какой модели разработки для каждого этапа разработки существует соответствующий этап тестирования, что подчеркивает важность верификации?
- a) Каскадная модель (Waterfall)
 - b) V-модель
 - c) Гибкая методология (Agile)
5. Почему верификация на ранних этапах разработки считается такой ценной?
- a) Она позволяет сразу выпустить продукт
 - b) Она позволяет найти и исправить дефекты дешевле
 - c) Она полностью заменяет тестирование
6. Какие два основных типа статических методов верификации представлены в лекции?
- a) Ручные и автоматизированные
 - b) Функциональные и нефункциональные
 - c) Модульные и интеграционные

7. Какой ручной метод верификации является наиболее распространенным в современных командах и часто проводится через систему контроля версий?
- a) Формальная верификация
 - b) Инспекция (Inspection)
 - c) Обзор кода (Code Review) через Pull/Merge Request
8. Что является целью менее формального подхода «Проход по коду» (Walkthrough)?
- a) Математическое доказательство корректности кода
 - b) Обмен идеями, знаниями и объяснение логики кода автором
 - c) Автоматическое сканирование кода на уязвимости
9. Что ищут инструменты статического анализа в коде? (Назовите 2-3 примера)
- a) Только опечатки в комментариях
 - b) Ошибки (например, деление на ноль), уязвимости безопасности, «запахи кода»
 - c) Скорость работы программы под нагрузкой
10. Для каких систем обычно применяется сложный и дорогой метод формальной верификации?
- a) Для любых мобильных приложений
 - b) Для критически важных систем (аэрокосмическая отрасль, медицинские устройства)
 - c) Для небольших веб-сайтов
11. Куда современные команды интегрируют инструменты статического анализа для автоматической проверки кода?
- a) В инструкцию для пользователя
 - b) В процессы непрерывной интеграции и доставки (CI/CD)
 - c) В договор с заказчиком
12. Какой основной метод динамической верификации?
- a) Инспекция требований
 - b) Тестирование
 - c) Парное программирование
13. Какой уровень тестирования проверяет взаимодействие между двумя и более модулями?
- a) Модульное тестирование (Unit)
 - b) Интеграционное тестирование (Integration)
 - c) Приемочное тестирование (Acceptance)
14. Какой вид функционального тестирования проводится, чтобы убедиться, что новые изменения не сломали существующую функциональность?
- a) Дымовое тестирование (Smoke)
 - b) Регрессионное тестирование (Regression)
 - c) Стресс-тестирование (Stress)
15. Какой вид нефункционального тестирования имитирует экстремальную нагрузку, чтобы найти точку отказа системы?
- a) Тестирование удобства использования (Usability)
 - b) Конфигурационное тестирование (Configuration)
 - c) Стресс-тестирование (Stress Testing)
16. Какая стратегия тестирования предполагает, что тестировщик не имеет доступа к исходному коду и ориентируется только на требования?
- a) Тестирование «белого ящика» (White-Box)
 - b) Тестирование «черного ящика» (Black-Box)
 - c) Тестирование «серого ящика» (Gray-Box)
17. Какая стратегия тестирования требует полного доступа к коду и использует техники покрытия операторов и ветвей?
- a) Тестирование «черного ящика» (Black-Box)
 - b) Тестирование «белого ящика» (White-Box)
 - c) Исследовательское тестирование (Exploratory)

18. Согласно концепции «Тестовой пирамиды», тесты какого уровня должны составлять большинство и почему?

- a) UI-тесты, потому что они самые важные
- b) Модульные тесты (Unit), потому что они быстрые, надежные и дешевые
- c) Интеграционные тесты, потому что они проверяют взаимодействие

19. Что является анти-паттерном, противоположным «Тестовой пирамиде», когда UI-тестов много, а модульных — мало?

- a) «Золотой треугольник»
- b) «Конус мороженого»
- c) «Счастливый путь»

20. Является ли верификация частью обеспечения качества (QA)?

- a) Нет, это совершенно отдельный процесс
- b) Да, верификация — это неотъемлемая часть QA
- c) QA — это часть верификации

Ответы для проверки:

1. b, 2. c, 3. b, 4. b, 5. b, 6. a, 7. c, 8. b, 9. b, 10. b, 11. b, 12. b, 13. b, 14. b, 15. c, 16. b, 17. b, 18. b, 19. b, 20. b

Тест 3

1. Как называется процесс поиска и исправления «просчетов» в программном обеспечении?
 - a) Тестирование
 - b) Компиляция
 - c) Отладка (дебаггинг)
2. Какой процент рабочего времени программист тратит на отладку, тестирование и поддержку кода?
 - a) До 10-20%
 - b) До 50-60%
 - c) До 90%
3. Почему ошибки в одной части сложной программной системы могут вызвать неожиданные последствия в другой?
 - a) Из-за взаимодействия множества модулей и фреймворков
 - b) Из-за недостатка комментариев в коде
 - c) Из-за проблем с оборудованием
4. Что из перечисленного НЕ является фактором возникновения ошибок в коде?
 - a) Человеческий фактор
 - b) Сложность систем
 - c) Использование отладчика
5. Как называется ошибка, которая возникает, когда код нарушает правила синтаксиса языка программирования?
 - a) Логическая ошибка
 - b) Синтаксическая ошибка
 - c) Ошибка времени выполнения (runtime)
6. Как называется ошибка, при которой программа работает, но выдает неверный результат из-за неправильной логики?
 - a) Ошибка компиляции
 - b) Синтаксическая ошибка
 - c) Логическая ошибка

7. Как называется ошибка, которая возникает во время выполнения программы, например, деление на ноль?
 - a) Ошибка времени выполнения (runtime)
 - b) Синтаксическая ошибка
 - c) Логическая ошибка
8. Какой тип ошибок легче всего обнаружить?
 - a) Логические ошибки
 - b) Ошибки времени выполнения
 - c) Синтаксические ошибки
9. Какой тип ошибок сложнее всего обнаружить?
 - a) Ошибки компиляции
 - b) Логические ошибки
 - c) Ошибки времени выполнения
10. Что является первым и самым важным шагом в процессе отладки?
 - a) Попытка исправить код наугад
 - b) Четкое воспроизведение ошибки
 - c) Поиск виноватого
11. Какой метод отладки предполагает деление кода на части для локализации проблемы?
 - a) Метод «разделяй и властвуй»
 - b) Метод «наугад»
 - c) Метод «сравнения»
12. Как называется процесс выдвижения предположений о причине ошибки и их проверки?
 - a) Выдвижение гипотез
 - b) Экспериментирование
 - c) Поиск в интернете
13. Какой главный инструмент помогает разработчику пошагово выполнять код и проверять значения переменных?
 - a) Антивирус
 - b) Текстовый редактор
 - c) Отладчик (дебаггер)
14. Что помогает предотвращать ошибки еще до их появления?
 - a) Поиск виноватых
 - b) Отладка
 - c) Статические анализаторы и юнит-тесты
15. Что такое юнит-тестирование?
 - a) Тестирование всего проекта целиком
 - b) Тестирование отдельных, наименьших частей программы
 - c) Тестирование, выполняемое конечными пользователями
16. Что такое логирование?
 - a) Запись информации о работе программы в специальный файл
 - b) Установка отладчика
 - c) Удаление ненужных файлов
17. Почему отладка считается процессом улучшения?
 - a) Потому что она помогает найти виноватого
 - b) Потому что она делает конечный продукт более надежным и качественным
 - c) Потому что она увеличивает скорость написания кода
18. Что приходит с практикой и помогает быстрее находить и исправлять ошибки?
 - a) Увеличение зарплаты
 - b) Развитие «чутья на ошибки» и аналитического мышления
 - c) Повышение в должности

19. Как называется систематический подход к отладке, который является ключом к эффективности?
- a) Хаотичные попытки исправления
 - b) Использование логики и анализа
 - c) Оповещение руководителя
20. Что, согласно лекции, делает программиста «архитектором цифровых систем»?
- a) Поиск и исправление ошибок
 - b) Написание чистого кода
 - c) Построение сложных систем из множества модулей

Ответы для проверки:

- 1. c
- 2. b
- 3. a
- 4. c
- 5. b
- 6. c
- 7. a
- 8. c
- 9. b
- 10. b
- 11. a
- 12. a
- 13. c
- 14. c
- 15. b
- 16. a
- 17. b
- 18. b
- 19. c
- 20. a

Тест 4

1. Какой стандарт дает следующее определение: «Тестирование — это процесс анализа программного обеспечения с целью выявления различий между существующими и требуемыми условиями»?
- a) ISTQB
 - b) IEEE 610.12-1990
 - c) ISO 9001
2. Какой из перечисленных аспектов не является основной целью тестирования, согласно лекции?
- a) Повышение уверенности в качестве
 - b) Обнаружение дефектов
 - c) Ускорение процесса разработки
3. Процесс, который отвечает на вопрос «Правильно ли мы создаем продукт?», называется:
- a) Валидация
 - b) Верификация
 - c) Тестирование
4. Почему исправление ошибки, допущенной на этапе требований, является самым дешевым?
- a) Она обнаруживается на самой ранней стадии
 - b) Ее можно исправить, изменив только документацию
 - c) Оба варианта верны

5. Какая из моделей жизненного цикла ПО наглядно демонстрирует связь между этапами разработки и уровнями тестирования?
 - a) Водопадная модель
 - b) Гибкие методологии (Agile)
 - c) V-модель
6. Согласно принципу «Парадокс пестицида», что нужно делать с тестовыми наборами?
 - a) Выполнять их как можно чаще, чтобы повысить эффективность
 - b) Регулярно пересматривать и дополнять новыми сценариями
 - c) Ничего не делать, так как они всегда будут находить дефекты
7. Какой принцип гласит, что большинство дефектов (около 80%) обычно обнаруживается в небольшом количестве модулей (около 20%)?
 - a) Принцип Парето
 - b) Парадокс пестицида
 - c) Раннее тестирование
8. Что из перечисленного является примером Обеспечения качества (QA), а не Контроля качества (QC)?
 - a) Выполнение тест-кейсов
 - b) Проведение код-ревью
 - c) Инспекция готового продукта
9. Какое понятие является частью Контроля качества (QC) и выступает его ключевым инструментом?
 - a) Обеспечение качества (QA)
 - b) Тестирование
 - c) Разработка
10. Какой метод тестирования предполагает, что внутренняя структура системы неизвестна тестирующему?
 - a) Метод «белого ящика»
 - b) Метод «серого ящика»
 - c) Метод «черного ящика»
11. Какой метод тестирования чаще всего применяется разработчиками на этапе модульного (unit) тестирования?
 - a) Метод «белого ящика»
 - b) Метод «черного ящика»
 - c) Метод «серого ящика»
12. Какая метрика показывает, сколько строк кода было выполнено в ходе тестов?
 - a) Покрытие ветвей
 - b) Покрытие операторов
 - c) Покрытие путей
13. Что является ключевой идеей модульного тестирования?
 - a) Проверка взаимодействия между модулями
 - b) Проверка всей системы как единого целого
 - c) Изоляция тестируемой части кода от внешних зависимостей
14. Какой вспомогательный код используется в интеграционном тестировании по стратегии «снизу вверх» для вызова низкоуровневых модулей?
 - a) Заглушки (Stubs)
 - b) Драйверы (Drivers)
 - c) Моки (Mocks)
15. Какой из уровней тестирования направлен на поиск ошибок во взаимодействии и на стыках между интегрированными компонентами?
 - a) Модульное тестирование
 - b) Системное тестирование
 - c) Интеграционное тестирование

16. Что является главной целью приемочного тестирования?
 - a) Найти как можно больше багов
 - b) Формально подтвердить, что продукт готов к использованию
 - c) Проверить производительность системы
17. Какой вид приемочного тестирования проводится внутри компании, но не самой командой разработки?
 - a) Пользовательское приемочное тестирование (UAT)
 - b) Бета-тестирование
 - c) Альфа-тестирование
18. Тестирование производительности и безопасности относится к какой категории тестирования?
 - a) Функциональное тестирование
 - b) Нефункциональное тестирование
 - c) Системное тестирование
19. Какой вид тестирования отвечает на вопрос: «ЧТО система делает?»
 - a) Функциональное тестирование
 - b) Нефункциональное тестирование
 - c) Приемочное тестирование
20. Что является примером техники тест-дизайна «анализ граничных значений»?
 - a) Проверка значений 18, 60, 17 и 61 для поля возраста (18-60)
 - b) Разбиение всех входных данных на валидные и невалидные группы
 - c) Создание таблицы, где результат зависит от комбинации условий

Ответы для проверки:

1. b
2. c
3. b
4. c
5. c
6. b
7. a
8. b
9. b
10. c
11. a
12. b
13. c
14. b
15. c
16. b
17. c
18. b
19. a
20. a

КРИТЕРИИ ОЦЕНКИ

МДК.01.03. РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

Критерии оценки:

Оценка «отлично»: студент владеет знаниями предмета в полном объеме учебной программы, достаточно глубоко осмысливает дисциплину, самостоятельно, в логической последовательности и исчерпывающе отвечает на все вопросы, подчеркивал при этом самое существенное, умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделять в нем главное: устанавливать причинно-следственные связи. Четко формирует ответы, решает ситуационные задачи повышенной сложности, хорошо знаком с основной литературой, увязывает теоретические аспекты предмета с задачами практического характера.

Оценка «хорошо»: студент владеет знаниями дисциплины почти в полном объеме программы (имеются пробелы знаний только в некоторых, особенно сложных разделах). Самостоятельно и отчасти при наводящих вопросах дает полноценные ответы, не всегда выделяет наиболее существенное, не допускает вместе с тем серьезных ошибок в ответах, умеет решать легкие и средней тяжести ситуационные задачи.

Оценка «удовлетворительно»: студент владеет основным объемом знаний по дисциплине; проявляет затруднения в самостоятельных ответах, оперирует неточными формулировками. В процессе ответов допускаются ошибки по существу вопросов. Студент способен решать лишь наиболее легкие задачи, владеет только обязательным минимумом методов исследований.

Оценка «неудовлетворительно»: студент обнаруживает незнание ответа на соответствующее задание, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно и неуверенно излагает материал, отмечаются такие недостатки в подготовке студента, которые являются серьезным препятствием к успешному овладению последующим материалом.

Критерии оценки устного опроса по дисциплине «Разработка мобильных приложений»

Устные опросы проводятся во время лекций, практических занятий и возможны при проведении промежуточной аттестации в качестве дополнительного испытания при недостаточности результатов тестирования. Основные вопросы для устного опроса доводятся до сведения студентов на предыдущем занятии.

Развернутый ответ студента должен представлять собой связное, логически последовательное сообщение на заданную тему, показывать его умение применять определения, правила в конкретных случаях.

По результатам ответа «отлично» выставляется студенту, если содержание ответа соответствует освещаемому вопросу, полностью раскрыта в ответе тема, ответ структурирован, даны правильные, аргументированные ответы на уточняющие вопросы, демонстрируется высокий уровень участия в дискуссии.

По результатам ответа «хорошо» выставляется студенту, если содержание ответа соответствует освещаемому вопросу, полностью раскрыта в ответе тема, даны правильные, аргументированные ответы на уточняющие вопросы, но имеются неточности, при этом ответ не структурирован и демонстрируется средний уровень участия в дискуссии.

По результатам ответа «удовлетворительно» выставляется студенту, если содержание ответа соответствует освещаемому вопросу, но при полном раскрытии темы имеются неточности, даны правильные, но не аргументированные ответы на уточняющие вопросы, демонстрируется низкий уровень участия в дискуссии, ответ не структурирован, информация трудна для восприятия.

По результатам ответа **«неудовлетворительно»** выставляется студенту, если содержание ответа соответствует освещаемому вопросу, но тема в ответе не полностью раскрыта, демонстрируется слабое владение категориальным аппаратом, происходит подмена понятий, даны неправильные, не аргументированные ответы на уточняющие вопросы, участие в дискуссии полностью отсутствует, ответ не структурирован, информация трудна для восприятия.

При несоответствии содержания ответа, освещаемому вопросу студент также получает **«неудовлетворительно»**.

Вопросы для подготовки к экзамену:

1. Понятие защищенной информационной системы.
2. Свойства защищенной ОС.
3. Безопасность информационных систем в нормативных документах.
4. Классификация защищенности ОС по международным стандартам.
5. Политика безопасности, формальное представление политик.
6. Классификация изъянов защиты.
7. Категории изъянов защиты в ОС.
8. Мобильное программирование, платформы для разработки.
9. Основы работы с сенсорным вводом. Обработка нескольких касаний.
10. Использование изображений.
11. Акселерометр и служба определения местоположения, вторичные потоки выполнения, обработка асинхронных операций и доступ к Веб-сервисам.
12. Особенности использования pivot и panorama.
12. Краткая история ОС Android.
13. Intel для Android: партнерство и инструментарий разработчика.
14. Архитектура приложений для Android. Ресурсы приложения. Пользовательский интерфейс. Инструментарий разработки приложений для Android.
15. Обзор шагов разработки типичного приложения под Android. Особенности разработки с использованием эмулятора. Отладка кода в эмуляторе и на реальных приложениях. Пример простейших программ Android-приложения. Запуск приложения на эмуляторе.
16. Тестирование приложения с помощью Dalvik Debug Monitor Server (DDMS).
17. Планирование покадровой анимации, анимирование, анимация шаблонов, видов, использование класса Camera.
18. Проверка безопасности, работа со службами, основанными на местоположении, использование HTTP-служб, службы AIDL.
19. Подготовка AndroidManifest.xml для загрузки, локализация приложения, подготовка ярлыка приложения, подготовка APK-файла для загрузки, работа пользователя с Android Market.
20. Работа с инструментами Intel для оптимизации отладки Android-приложений

Типовые задания для оценки освоения учебной дисциплины

1. Основные платформы мобильных приложений, сравнительная характеристика
2. Нативные приложения, веб-приложения, гибридные и кроссплатформенные приложения, их области применения
3. Основные языки для разработки мобильных приложений (Java, Objective-C и др.)
4. Инструменты разработки мобильных приложений (JDK/Andr
5. Инструментарий среды разработки мобильных приложений. Структура типичного мобильного приложения.

6. Архитектура платформы Android. Dalvik Virtual Machine. Среда разработки для Android. Eclipse IDE. П
7. Плагин ADT. Android Virtual Device.
8. Android SDK. Версии SDK и Android API Level.
9. Инструментарий среды разработки мобильных приложений
10. Структура типичного мобильного приложения
11. Элементы управления и контейнеры
12. Работа со списками
13. Способы хранения данных
14. Структура проекта Android-приложения в Eclipse. Каталоги ресурсов. Файл R.java
15. Графический интерфейс пользователя в Android-приложениях. XML разметка интерфейса.
16. XML-разметка интерфейса пользователя
17. Ресурсы в Android-приложениях
18. Ресурсы в Windows Phone-приложениях.
19. Многопоточные приложения в Android и Windows Phone. Использование системных таймеров и системного времени.
20. Использование объектов Intent. Intent-фильтры
21. Использование ресурсов. Ссылки на ресурсы. Загрузка простых типов из ресурсов. Загрузка файлов произвольного типа.
22. Пользовательские настройки. Использование Shared Preferences. Виды настроек. Службы в Android. Компонент Service.
23. Датчики мобильных устройств. Управление датчиками в приложении. Виды датчиков и особенности их использования
24. Программный доступ к дисплею устройства. Менеджер окон. Параметры дисплея.

Критерии оценки практических работ

Оценка «**отлично**» выставляется, если: выполнены все задания практической работы, студент четко и без ошибок ответил на все контрольные вопросы.

Оценка «**хорошо**» выставляется, если: выполнены все задания практической работы, студент ответил на все контрольные вопросы с замечаниями.

Оценка «**удовлетворительно**» выставляется, если: выполнены все задания практической работы, с замечаниями; студент ответил на все контрольные вопросы с замечаниями.

Оценка «**неудовлетворительно**» выставляется, если: студент не выполнил или выполнил неправильно задания практической работы, студент ответил на контрольные вопросы с ошибками или не ответил на контрольные вопросы.

Типовые задания по практическим работам Практические задания

Цель: Разработать мобильное приложение по заданной теме.

1. Разработайте с помощью MS Visual Studio .NET серверное веб-приложение, использующее управляющий ASP.NET Mobile
2. элемент:<MOBILE:CALENDAR>

Добавьте для этого элемента обработчик события OnSelectionChanged, который должен отображать выбранный пользователем мобильного устройства день календаря с помощью элемента: <MOBILE:LABEL> . Код обработчика события может выглядеть следующим образом:

```
<script runat = "server">  
    Protector void CalChanged (Odgect sender, EventArgs e) Label1.Text = "You  
selected:" + calendar1.SelectedDate;  
</script>
```

Создайте необходимый виртуальный каталог на веб-сервере и проверьте работу приложения с помощью эмулятора мобильного устройства.

3. Создайте простейшей веб-странице, отображающей на экранемобильного устройства сообщение "Hello world!".

4. Создать игру, реализованную как midp приложение. Создать меню игры, содержащее пункты:

Запуск игры Уровень сложности и другие настройки.

В игре должны подсчитываться набранные очки и лучший результат сохраняется в течение игры.

5. Создать сервер обмена текстовыми сообщениями.

Создать сервер, которые будет получать данные от клиента в виде строк и выводить на экран мидлета.

Сервер должен быть многопоточным, т.е. Обслуживать одновременно с несколько клиентов.

Для завершения работы клиент должен послать строку «exit». Создать клиента для сервера обмена сообщениями.

Клиентское приложение должно иметь поле для ввода данных для отправки.

По выбору студента приложение может выполнять функции и сервера, и клиента одновременно, либо могут быть реализованы, как два отдельных приложения.

6. Разработать приложение calculator с одним activity. Графическое представление приложения calculator реализовать с кнопками цифр, математических операций (сложения, вычитания, умножения, деления и др.), получения результата

7. Разработать приложение Taxi, состоящее из трех Activity (рисунок). В первом Activity создать три редактируемых текстовых поля (EditText) для ввода пользователем регистрационных данных (телефона, имени и фамилии), кнопку Registration для запуска второго Activity.

При нажатии на кнопку Registration выполнить явный вызов второго Activity с передачей данных о пользователе (телефон, имя и фамилия).

Во втором Activity создать два текстовых поля (TextView) для вывода переданной информации о пользователе (имя+фамилия, телефон), пустое по умолчанию текстовое поле (TextView) для вывода маршрута движения, кнопку Set path для ввода этого маршрута, кнопку вызова такси Call Taxi (недоступна, пока не введен маршрут движения).

При нажатии на кнопку Set path выполнить неявный вызов третьего Activity с помощью метода startActivityForResult.

В третьем Activity создать шесть редактируемых текстовых полей (EditText) для ввода параметров маршрута движения, кнопку ОК для возврата во второе Activity.

При нажатии на кнопку ОК реализовать возврат во второе Activity с передачей в качестве результата параметров маршрута движения.

После возврата во второе Activity в текстовое поле вывести информация о маршруте движения и предложение вызвать такси, кнопку вызова такси Call taxi сделать доступной.

При нажатии на кнопку Call Taxi вывести всплывающее сообщение об успешной отправке такси. Реализовать сохранение регистрационных данных пользователя в исходном Activity с помощью класса SharedPreferences и восстанавливать эту информацию при повторных запусках приложения.

При этом название кнопки Registration должно программно меняться на Log in.7. Разработать приложение MiniShop, состоящее из двух Activity (рисунок).

В первом Activity создать список ListView с Header и Footer.

В Footer разместить текстовое поле (TextView) для ввода количества активированных пользователем товаров, кнопку Show Checked Items для перехода в корзину товаров.

Реализовать кастомизированный список ListView с помощью собственного адаптера, наследующего класс BaseAdapter.

В каждом пункте списка отобразить следующую информацию о товаре: идентификационный номер, название, стоимость, чек-бокс для возможности выбора товара пользователем.

В текстовом поле (TextView) Footer списка динамически отображать общее текущее количество активированных товаров.

При нажатии на кнопку Show Checked Items реализовать переход во второе Activity с корзиной товаров.

Корзину товаров реализовать в виде нового кастомизированного списка с выбранными товарами. Продемонстрировать работу приложения MiniShop на эмуляторе или реальном устройстве.

8. Разработать приложение MyNotes представляющее собой View Pager. Поместить в View Pager четыре фрагмента: FragmentShow, FragmentAdd, FragmentDel, FragmentUpdate. В View Pager добавить верхнее меню вкладок (PagerTabStrip) с заголовками Show, Add, Del, Update.

Во фрагменте FragmentShow реализовать кастомизированный список заметок ListView с помощью собственного адаптера.

В каждом пункте списка отобразить следующую информацию о заметке пользователя: номер, описание заметки.

Хранение, а также предоставление информации о заметках адаптеру реализовать с помощью базы данных SQLite.

Во фрагменте FragmentAdd реализовать функционал добавления новой заметки посредством ввода описания заметки в поле EditText и добавления информации в базу данных SQLite по нажатию на кнопку Add.

Во фрагменте FragmentDel реализовать функционал удаления новой заметки посредством ввода ее номера в поле EditText и удаления информации из базы данных SQLite по нажатию на кнопку Del.

Во фрагменте FragmentUpdate реализовать функционал обновления существующей заметки посредством ввода ее номера в поле EditText, ввода нового описания в поле EditText и обновления информации в базе данных SQLite по нажатию на кнопку Update.

Предусмотреть обработку исключительной ситуации отсутствия заметки по указанному номеру посредством вывода пользователю всплывающего сообщения соответствующего содержания.

Продемонстрировать работу приложения MyNotes на эмуляторе или реальном устройстве.

Примеры практических заданий:

1. Требуется разработать приложение с графическим пользовательским интерфейсом, поддерживающее создание/редактирование/удаление/поиск заметок. Два варианта хранения заметок: А) в базе SQLite. Б) С использованием файловой системы.
2. Создать приложение с графическим пользовательским интерфейсом с функциями: Определение местоположения пользователя на карте Google Map; Определение скорости и направления движения пользователя; Масштабирование карты. Программа должна быть конфигурируемой. Настройки: Режим определения местоположения (через GPS либо по сотам); Включение/отключение режима поиска.
3. Разработать приложение-таймер с использованием датчика ориентации в виде песочных часов. Каждый раз для того, чтобы активировать таймер, необходимо перевернуть экран мобильного устройства вверх ногами. Используйте анимацию для показа «перетекающего песка» и переворота песочных часов. Для задания времени перетекания песка требуется разработать push-notification сервер. Через форму ввода на сервере можно отправлять на клиент (приложение-таймер) указанное время (числовой ввод).
4. Разработка программы для обмена мгновенными сообщениями. Требуется разработать приложение для обмена мгновенными сообщениями через Wi-Fi/Bluetooth. Поддерживаемые режимы: 1. Активный режим. Приложение занимает весь экран, содержит поля для отправки сообщений и список принятых сообщений. 2. Режим уведомлений. Приложение через уведомления показывает принятые сообщения.
5. Разработка мобильного сайта с адаптивным дизайном. Требуется разработать сайт, пригодный для просмотра на смартфонах и планшетах, с использованием принципов адаптивного дизайна: 1) Гибкая сетка. 2) Пропорциональные шрифты. 3) Масштабируемые изображения. 4) Медиа-запросы

Критерии оценки самостоятельной работы студентов

Цель самостоятельной работы - содействие оптимальному усвоению студентами учебного материала, развитие их познавательной активности, готовности и потребности в самообразовании.

Задачи самостоятельной работы:

- углубление и систематизация знаний;
- развитие аналитико-синтетических способностей умственной деятельности, умений работы с различной по объему и виду информацией, учебной и научной литературой; практическое применение знаний, умений;
- развитие навыков организации самостоятельного учебного труда и контроля за его эффективностью.

Самостоятельная работа студентов предусмотрена рабочей программой и организуется в соответствии с УМД. Качество выполнения внеаудиторной самостоятельной работы студентов оценивается посредством текущего контроля самостоятельной работы студентов.

Максимальное количество баллов **«отлично»** студент получает, если:

- студент свободно применяет знания на практике, не допускает ошибок в воспроизведении изученного материала, выделяет главные положения в изученном материале и не затрудняется в ответах на видеоизмененные вопросы;

- весь объем программного материала усвоен полностью;
- обстоятельно с достаточной полнотой излагает соответствующую тему;
- материал (задание) оформлен аккуратно в соответствии с требованиями;

Оценку **«хорошо»** студент получает, если:

- студент знает весь изученный материал, отвечает без особых затруднений на вопросы преподавателя;
- применяет полученные знания на практике;
- в условных ответах не допускает серьезных ошибок, легко устраняет определенные неточности с помощью дополнительных вопросов преподавателя; неполно, но правильно изложено задание;
- неполно, но правильно изложено задание;
- может обосновать свой ответ, привести необходимые примеры;
- материал оформлен недостаточно аккуратно и в соответствии с требованиями.

Оценку **«удовлетворительно»** студент получает, если:

- студент обнаруживает освоение основного материала, но испытывает затруднения при его самостоятельном воспроизведении и требует дополнительных дополняющих вопросов преподавателя;
- предпочитает отвечать на вопросы воспроизводящего характера и испытывает затруднения при ответах на воспроизводящие вопросы;
- излагает выполнение задания недостаточно логично и последовательно;
- затрудняется при ответах на вопросы преподавателя;
- материал оформлен не аккуратно или не в соответствии с требованиями.

Оценку **«неудовлетворительно»** студент получает, если:

- у студента имеются отдельные представления об изучаемом материале, но все же большая часть не усвоена;
- задание изложено неполно, без учета требований к оформлению;
- при изложении были допущены существенные ошибки, т.е. если оно не удовлетворяет требованиям, установленным преподавателем к данному виду работы.

Типовые задания для самостоятельной работы

Самостоятельная работа направлена на решение следующих задач:

- освоение, углубление и расширение знаний, умений, навыков в рамках формируемых компетенций, предусмотренных образовательной программой;
- формирование умений использовать нормативную, правовую, справочную документацию, специальную литературу, иные информационные источники для решения учебных и профессиональных задач;
- развитие познавательных способностей и активности обучающихся: творческой □ инициативы, самостоятельности, ответственности и организованности;
- формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- развитие исследовательских умений;
- формирование потребности в непрерывном образовании.

Формами контроля за самостоятельным теоретическим обучением являются устные теоретические опросы, которые осуществляются преподавателем на практических занятиях в устной форме, преследующие цель проверки знаний студентов по основным понятиям и терминам по теме самостоятельной работы. В случае представления

студентом, выполненного им в письменном виде сообщения (конспекта) по предложенным вопросам темы, возможна его защита на практическом занятии или в индивидуальном порядке.

Критериями оценки результатов самостоятельной работы обучающегося могут быть:

- уровень освоения учебного материала,
- умение использовать теоретические знания при выполнении практических задач,
- полнота обще-учебных представлений, знаний и умений по изучаемой теме,
- которой относится данная самостоятельная работа, обоснованность и четкость изложения ответа на поставленный по самостоятельной работе вопрос,
- оформление отчетного материала в соответствии с требованиями, предъявляемыми к подобного рода материалам.

Примерные темы для самостоятельной работы:

1. Основные языки для разработки мобильных приложений (Java, Objective-C и др.) (выполнение индивидуальных заданий)
2. Инструментарий среды разработки мобильных приложений. Структура типичного мобильного приложения (подготовка докладов)
3. Ресурсы в Android-приложениях (подготовка презентаций)
4. Датчики мобильных устройств. Управление датчиками в приложении. Виды датчиков и особенности их использования (подготовка презентаций)

Методические рекомендации по написанию и оформлению доклада.

Доклад – это вид самостоятельной работы студентов, заключающийся в разработке студентами темы на основе изучения литературы и развернутом публичном сообщении по данной проблеме.

Цель доклада - сформировать научно-исследовательские навыки и умения у студентов, способствовать овладению методами научного познания, освоить навыки публичного выступления, научиться критически мыслить. При этом главная составляющая это публичное выступление.

Этапы подготовки доклада:

- выбор темы доклада;
- подбор и изучение наиболее важных учебных, научных работ по данной теме, нормативных правовых актов;
- анализ изученного материала, выделение наиболее значимых для раскрытия темы доклада фактов, мнений ученых; составление плана доклада;
- написание текста доклада с соблюдением требований научного стиля.

Структура доклада:

1. Вступление, в котором указываются тема доклада; цель доклада; связь данной темы с другими темами; актуальность, проблематика темы; краткий обзор изученной литературы по данной теме и т.п.
2. Основная часть, которая содержит логичное, последовательное изложение материала.
3. Заключение, в котором: подводятся итоги, формулируются выводы; подчеркивается значение рассмотренной проблемы; выделяются основные проблемы, пути и способы их решения и т.п.; Приложения (схемы, таблицы для более наглядного освещения темы).

Требования к оформлению работы:

Размер бумаги - А4; поля: верхнее, нижнее - 2 см; левое, правое - 2,5 см; колонтитулы - 1,25 см; ориентация книжная; шрифт TimesNewRoman, высота 14pt; межстрочное расстояние – 1,5; выравнивание по ширине; отступ первой строки 1,25 см.

В случае невозможности выполнить работу в электронном варианте, допускается рукописное оформление доклада.

Требования к защите доклада:

1. Продолжительность выступления обычно не превышает 3-5 минут. Поэтому при подготовке доклада из текста работы отбирается самое главное. В докладе должно быть кратко отражено основное содержание всех глав и разделов исследовательской работы.
2. Для успешного выступления с докладом заучите значение всех терминов, которые употребляются в докладе.
3. При соблюдении этих правил у вас должен получиться интересный доклад, который, несомненно, будет высоко оценен преподавателем.

Методические рекомендации по подготовке презентации

Компьютерную презентацию, сопровождающую выступление докладчика, удобнее всего подготовить в программе MS PowerPoint. Презентация как документ представляет собой последовательность сменяющих друг друга слайдов - то есть электронных страничек, занимающих весь экран монитора (без присутствия панелей программы). Чаще всего демонстрация презентации проецируется на большом экране, реже - раздается собравшимся как печатный материал. Количество слайдов адекватно содержанию и продолжительности выступления (например, для 5- минутного выступления рекомендуется использовать не более 10 слайдов).

На первом слайде обязательно представляется тема выступления и сведения об авторах. Следующие слайды можно подготовить, используя две различные стратегии их подготовки:

1 стратегия: на слайды выносятся опорный конспект выступления и ключевые слова с тем, чтобы пользоваться ими как планом для выступления. В этом случае к слайдам предъявляются следующие требования: объем текста на слайде - не больше 7 строк;

- маркированный/нумерованный список содержит не более 7 элементов; отсутствуют знаки пунктуации в конце строк в маркированных и нумерованных в списках;
- значимая информация выделяется с помощью цвета, кегля, эффектов анимации.

Особо внимательно необходимо проверить текст на отсутствие ошибок и опечаток. Основная ошибка при выборе данной стратегии состоит в том, что выступающие заменяют свою речь чтением текста со слайдов.

2 стратегия: на слайды помещается фактический материал (таблицы, графики, фотографии и пр.), который является уместным и достаточным средством наглядности, помогает в раскрытии стержневой идеи выступления. В этом случае к слайдам предъявляются следующие требования:

- выбранные средства визуализации информации (таблицы, схемы, графики и т.д.)
- соответствуют содержанию;
- использованы иллюстрации хорошего качества (высокого разрешения), с четким изображением (как правило, никто из присутствующих не заинтересован вчитываться в текст на ваших слайдах и всматриваться в мелкие иллюстрации);

Максимальное количество графической информации на одном слайде - 2 рисунка (фотографии, схемы и т.д.) с текстовыми комментариями (не более 2 строк к каждому). Наиболее важная информация должна располагаться в центре экрана.

Основная ошибка при выборе данной стратегии - «соревнование» со своим иллюстративным материалов (аудитории не предоставляется достаточно времени, чтобы воспринять материал на слайдах). Обычный слайд, без эффектов анимации должен демонстрироваться на экране не менее

10 - 15 секунд. За меньшее время присутствующие не успеет осознать содержание слайда. Если какая-то картинка появилась на 5 секунд, а потом тут же сменилась другой, то аудитория будет считать, что докладчик ее подгоняет. Обратного (позитивного) эффекта можно достигнуть, если докладчик пролистывает множество слайдов со сложными таблицами и диаграммами, говоря при этом «Вот тут приведен разного рода вспомогательный материал, но я его хочу пропустить, чтобы не перегружать выступление подробностями». Правда, такой прием делать в начале и в конце презентации - рискованно, оптимальный вариант - в середине выступления.

Если на слайде приводится сложная диаграмма, ее необходимо предварить вводными словами (например, «На этой диаграмме приводится то-то и то-то, зеленым отмечены показатели А, синим - показатели Б»), с тем, чтобы дать время аудитории на ее рассмотрение, а только затем приступить к ее обсуждению. Каждый слайд, в среднем должен находиться на экране не меньше 40 - 60 секунд (без учета времени на случайно возникшее обсуждение). В связи с этим лучше настроить презентацию не на автоматический показ, а на смену слайдов самим докладчиком.

Особо тщательно необходимо отнестись к **оформлению презентации**. Для всех слайдов презентации по возможности необходимо использовать один и тот же шаблон оформления кегль - для заголовков - не меньше 24 пунктов, для информации - не менее

18. В презентациях не принято ставить переносы в словах.

Подумайте, не отвлекайте ли вы слушателей своей же презентацией? Яркие краски, сложные цветные построения, излишняя анимация, выпрыгивающий текст или иллюстрация - не самое лучшее дополнение к научному докладу. Также нежелательны звуковые эффекты в ходе демонстрации презентации. Наилучшими являются контрастные цвета фона и текста (белый фон - черный текст; темно-синий фон - светло-желтый текст и т. д.). Лучше не смешивать разные типы шрифтов в одной презентации. Рекомендуется не злоупотреблять прописными буквами (они читаются хуже).

Неконтрастные слайды будут смотреться тусклыми и невыразительными, особенно в светлых аудиториях. Для лучшей ориентации в презентации по ходу выступления лучше пронумеровать слайды. Желательно, чтобы на слайдах оставались поля, не менее 1 см с каждой стороны. Вспомогательная информация (управляющие кнопки) не должны преобладать над основной информацией (текстом, иллюстрациями). Использовать встроенные эффекты анимации можно только, когда без этого не обойтись (например, последовательное появление элементов диаграммы). Для акцентирования внимания на какой-то конкретной информации слайда можно воспользоваться лазерной указкой.

Диаграммы готовятся с использованием мастера диаграмм табличного процессора MS Excel. Для ввода числовых данных используется числовой формат с разделителем групп разрядов. Если данные (подписи данных) являются дробными числами, то число отображаемых десятичных знаков должно быть одинаково для всей группы этих данных (всего ряда подписей данных). Данные и подписи не должны накладываться друг на друга и сливаться с графическими элементами диаграммы. Структурные диаграммы готовятся при помощи стандартных средств рисования пакета MS Office. Если при форматировании слайда есть необходимость пропорционально уменьшить размер диаграммы, то размер шрифтов реквизитов должен быть увеличен с таким расчетом, чтобы реальное отображение объектов диаграммы соответствовало значениям, указанным в таблице. В таблицах не должно быть более 4 строк и 4 столбцов - в противном случае данные в таблице будет просто невозможно увидеть. Ячейки с названиями строк и столбцов и наиболее значимые данные рекомендуется выделять цветом.

Табличная информация вставляется в материалы как таблица текстового процессора MS Word или табличного процессора MS Excel. При вставке таблицы как объекта и пропорциональном изменении ее размера реальный отображаемый размер шрифта должен быть не менее 18 pt. Таблицы и диаграммы размещаются на светлом или белом фоне.

Если Вы предпочитаете воспользоваться помощью оператора (что тоже возможно), а не листать слайды самостоятельно, очень полезно предусмотреть ссылки на слайды в тексте доклада («Следующий слайд, пожалуйста...»).

Заключительный слайд презентации, содержащий текст

«Спасибо за внимание» или «Конец», вряд ли приемлем для презентации, сопровождающей публичное выступление, поскольку завершение показа слайдов еще не является завершением выступления. Кроме того, такие слайды, так же, как и слайд

«Вопросы?», дублируют устное сообщение. Оптимальным вариантом представляется повторение первого слайда в конце презентации, поскольку это дает возможность еще раз напомнить слушателям тему выступления и имя докладчика и либо перейти к вопросам, либо завершить выступление.

Для показа файл презентации необходимо сохранить в формате

«Демонстрация PowerPoint» (Файл - Сохранить как - Тип файла - Демонстрация PowerPoint). В этом случае презентация автоматически открывается в режиме полноэкранного показа (slideshow) и слушатели избавлены как от вида рабочего окна программы PowerPoint, так и от потерь времени в начале показа презентации.

После подготовки презентации полезно проконтролировать себя вопросами:

- удалось ли достичь конечной цели презентации (что удалось определить, объяснить, предложить или продемонстрировать с помощью нее?);
- к каким особенностям объекта презентации удалось привлечь внимание аудитории?
- не отвлекает ли созданная презентация от устного выступления? После подготовки презентации необходима репетиция выступления.

Контрольно-оценочные материалы для промежуточной аттестации по дисциплине

Задачей промежуточной аттестации по дисциплине является комплексная оценка уровней достижения планируемых результатов обучения.

В критерии итоговой оценки уровня подготовки обучаемого по дисциплине входят:

- уровень усвоения материала, предусмотренного рабочей программой;
- уровень практических умений, продемонстрированных обучаемым при выполнении практических заданий;
- уровень освоения компетенций, позволяющих решать ситуационные,
- профессиональные задачи; обоснованность, четкость, полнота ответа

Предметом оценки являются умения и знания, приобретенные в период освоения дисциплины.

Умения

У1 – осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.

У2 – использовать информационные технологии в профессиональной деятельности.

У3 – разрабатывать модули программного обеспечения для мобильных платформ

Знания

З1 – основные этапы разработки мобильных приложений

З2 – основные принципы технологии структурного и объектно-ориентированного программирования

ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.

ОК 09. Использовать информационные технологии в профессиональной деятельности.

Зачет - это форма промежуточной аттестации, задачей которой является комплексная оценка уровней достижения планируемых результатов обучения по дисциплине.

Для получения зачета необходимо иметь оценки, полученные в рамках текущего контроля успеваемости, по каждой теме, предусмотренной дисциплиной.

В критерии итоговой оценки уровня подготовки, обучающегося по дисциплине, входят:

- уровень усвоения студентом материала, предусмотренного рабочей программой;
- уровень практических умений, продемонстрированных студентом при выполнении практических заданий;
- уровень освоения компетенций, позволяющих выполнять практические задания;
- логика мышления, обоснованность, четкость, полнота ответов

СТРУКТУРА ИТОГОВОГО ТЕСТА:

Тест содержит на Moodle. Результат выдается сразу после тестирования и формируется отчет протестированных студентов на сервере.

Время на подготовку и выполнение: Выполнение – 20 минут. За правильный ответ выставляется по

1 баллу, затем результаты суммируются, и выставляется оценка. За неправильный ответ 0 баллов.

Критерии оценки промежуточной аттестации:

Оценка «отлично» выставляется, если имеются все конспекты лекции, обучающимися выполнены 100% практических работ, оценка за итоговое тестирование – «отлично», средний балл по аттестациям не ниже 4,5.

Оценка «хорошо» выставляется, если имеются все конспекты лекции, обучающимися выполнены 100% практических работ, оценка за итоговое тестирование – «хорошо», средний балл по аттестациям не ниже 3,5.

Оценка «удовлетворительно» выставляется, если имеются все конспекты лекции, обучающимися выполнены 100% практических работ, оценка за итоговое тестирование – «удовлетворительно», средний балл по аттестациям не ниже 2,5.

Оценка «неудовлетворительно» выставляется, если имеются все конспекты лекции обучающимися выполнено менее 100% практических работ, оценка за итоговое тестирование –

«неудовлетворительно», средний балл по аттестациям ниже 2,5.

Цель итогового тестирования:

Тестирование по дисциплине «**Разработка мобильных приложений**» предназначено для проверки теоретических знаний и понятийного аппарата, которые лежат в основе профессионального образования и найдут самое широкое применение в будущей профессиональной деятельности учащихся по специальностям

09.02.07 Информационные системы и программирование.

Критерии оценки знаний:

Процент правильных ответов, %	Оценка знаний
90-100	5 «отлично»
80-89	4 «хорошо»
70-79	3 «удовлетворительно»
Менее 70	2 «неудовлетворительно»

Список теоретических заданий для подготовки к итоговому тестированию

(ТЗ) по дисциплине «Разработка мобильных приложений».

1. Набор средств программирования, который содержит инструменты, необходимые для создания, компиляции и сборки мобильного приложения называется:
 - а) Android SDK
 - б) JDK
 - в) плагин ADT
 - г) Android NDK
2. С какой целью был создан Open Handset Alliance?
 - а) писать историю развития ОС Android
 - б) продавать смартфоны под управлением Android
 - в) рекламировать смартфоны под управлением Android
 - г) разрабатывать открытые стандарты для мобильных устройств
3. С какой целью инструмент Intel* Graphics Performance Analyzers (Intel* GPA) System Analyzer используется в среде разработки Intel*Beacon Mountain?
 - а) позволить разработчикам оптимизировать загрузку системы при использовании процедур OpenGL
 - б) для ускорения работы эмулятора в среде разработки
 - в) для оптимизированной обработки данных и изображений
 - г) позволить разработчикам эффективно распараллелить C++ мобильные приложения
4. Библиотеки, реализованные на базе PacketVideo OpenCORE:
 - а) Media Framework
 - б) SQLite
 - в) FreeType
 - г) 3D библиотеки
5. Какой движок баз данных используется в ОС Android?
 - а) InnoDB
 - б) DBM
 - в) MyISAM
 - г) SQLite
6. С какой целью инструмент Intel* Integrated Performance Primitives (Intel* IPP) используется в среде разработки Intel* Beacon Mountain?
 - а) для оптимизированной обработки данных и изображений
 - б) позволить разработчикам оптимизировать загрузку системы при использовании процедур OpenGL
 - в) для ускорения работы эмулятора в среде разработки
 - г) позволить разработчикам эффективно распараллелить C++ мобильные приложения
7. Intel XDK поддерживает разработку под:
 - а) JavaFX Mobile
 - б) Apple iOS, BlackBerry OS
 - в) MtkOS, Symbian OS, Microsoft Windows 8
 - г) Android, Apple iOS, Microsoft Windows 8, Tizen
8. Каждый приемник широкополосных сообщений является наследником класса ...
 - а) BroadcastReceiver
 - б) IntentReceiver
 - в) ContentProvider
 - г) BroadcastReceiver

9. Какой класс является основным строительным блоком для компонентов пользовательского интерфейса (UI), определяет прямоугольную область экрана и отвечает за прорисовку и обработку событий?

- а) GUI
- б) View
- в) UIComponent
- г) Widget

10. Какой слушатель используется для отслеживания события касания экрана устройства?

- а) OnPressListener б) OnTouchListener в) OnClickListener г) OnInputListener

11. В какой папке необходимо разместить XML файлы, которые определяют все меню приложения?

- а) res/values б) res/items в) res/layout г) res/menu

12. Фоновые приложения ...

- а) после настройки не предполагают взаимодействия с пользователем, большую часть времени находятся и работают в скрытом состоянии
- б) выполняют свои функции и когда видимы на экране, и когда скрыты другими приложениями
- в) небольшие приложения, отображаемые в виде графического объекта на рабочем столе
- г) большую часть времени работают в фоновом режиме, однако допускают взаимодействие с пользователем и после настройки

13. Полный иерархический список обязательных файлов и папок проекта можно увидеть на вкладке ...

- а) Package Explorer б) Internet Explorer в) Navigator
- г) Project Explorer

14. Какой компонент управляет распределенным множеством данных приложения?

- а) сервис (Service)
- б) активность (Activity)
- в) приемник широковещательных сообщений (Broadcast Receiver) г) контент-провайдер (Content Provider)

15. Выберите верную последовательность действий, необходимых для создания в приложении контент-провайдера.

- а) Создание класса наследника от класса ContentProvider; Определение способа организации данных; Заполнение контент-провайдера данными
- б) Проектирование способа хранения данных; Определение способа организации данных;
- в) Создание класса наследника от класса ContentProvider; Заполнение контент-провайдера данными; Определение способа работы с данными
- г) Проектирование способа хранения данных; Создание класса-наследника от класса ContentProvider; Определение строки авторизации провайдера, URI для его строк и имен столбцов

16. Выберите верные утверждения относительно объекта-намерения (Intent).

- а) представляет собой структуру данных, содержащую описание операции, которая должна быть выполнена, и обычно используется для запуска активности или сервиса
- б) используется для передачи сообщений пользователю в) используется для получения инструкций от пользователя г) используются для передачи сообщений между основными компонентами приложений.

- 17.** Расположение элементов мобильного приложения:
- а) полезно для передачи иерархии
 - б) влияет на удобство использования
 - в) полезно для создания пространственных отношений между объектами на экране и объектами реального мира
 - г) все варианты ответа верны
- 18.** Какие элементы управления применяются для действий по настройке?
- а) командные элементы управления
 - б) элементы выбора
 - в) элементы ввода
 - г) элементы отображения
- 19.** Примерами комбо-элементов не являются:
- а) комбо-список
 - б) все вышеперечисленное
 - в) комбо-кнопка
 - г) комбо-поле
- 20.** Дизайн или проектирование интерфейса для графических дизайнеров:
- а) все варианты ответа верны
 - б) прозрачность и понятность информации
 - в) тон, стиль, композиция, которые являются атрибутами бренда
 - г) передача информации о поведении посредством ожидаемого назначения
- 21.** Более крупные элементы:
- а) привлекают больше внимания
 - б) все варианты ответа верны
 - в) размер не влияет на уровень внимания
 - г) привлекают меньше внимания
- 22.** К традиционным типографическим инструментам не относят
- а) масштаб
 - б) цвет
 - в) разреженность
 - г) выравнивание по сетке
- 23.** К элементам ввода относят:
- а) ограничивающие элементы ввода
 - б) ползунки
 - в) счетчики
 - г) все вышеперечисленное
- 24.** Выделяют следующие категории плотности экрана для Android-устройств:
- а) HDPI, XHDPI, XXHDPI, и XXXHDPI
 - б) правильный вариант ответа отсутствует
 - в) LDPI, MDPI, HDPI, XHDPI, XXHDPI, и XXXHDPI
 - г) LDPI, MDPI, HDPI
- 25.** Следующие утверждения не верны:
- а) не используйте интерфейсные элемент
 - б) картинки работают быстрее, чем слова
 - в) на любом шаге должна быть возможность вернуться назад
 - г) если объекты похожи, они должны выполнять сходные действия

- 26.** Следующие утверждения верны:
- а) текстура бесполезна для передачи различий или привлечения внимания
 - б) восприятие направления затруднено при больших размерах объектов
 - в) все варианты ответа верны
 - г) люди легко воспринимают контрастность
- 27.** Основные вкладки (FixedTabs) удобны при отображении
- а) от четырех вкладок
 - б) двух вкладок
 - в) трех и более вкладок
 - г) трех и менее вкладок
- 28.** Диалоговое окно, содержащее линейку процесса выполнения какого-то действия — это
- а) DatePicker Dialog
 - б) AlertDialog
 - в) Progress Dialog
 - г) DialogFragment
- 29.** Уведомления стоит использовать, когда
- а) сообщение не требует ответа пользователя, но важно для продолжения его работы
 - б) сообщение является важным и требует немедленного прочтения и ответа
 - в) сообщение является важным, однако требует немедленного прочтения, но не ответа
 - г) сообщение является важным, однако не требует немедленного прочтения и ответа
- 30.** Какой метод запускает новую активность?
- а) startActivity ()
 - б) beginActivity ()
 - в) intentActivity ()
 - г) newActivity ()

КРИТЕРИИ ОЦЕНКИ

МДК 01.04 Системное программирование

Критерии оценки:

Оценка «отлично»: студент владеет знаниями предмета в полном объеме учебной программы, достаточно глубоко осмысливает дисциплину, самостоятельно, в логической последовательности и исчерпывающе отвечает на все вопросы, подчеркивал при этом самое существенное, умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделять в нем главное: устанавливать причинно-следственные связи. Четко формирует ответы, решает ситуационные задачи повышенной сложности, хорошо знаком с основной литературой, увязывает теоретические аспекты предмета с задачами практического характера.

Оценка «хорошо»: студент владеет знаниями дисциплины почти в полном объеме программы (имеются пробелы знаний только в некоторых, особенно сложных разделах). Самостоятельно и отчасти при наводящих вопросах дает полноценные ответы, не всегда выделяет наиболее существенное, не допускает вместе с тем серьезных ошибок в ответах, умеет решать легкие и средней тяжести ситуационные задачи.

Оценка «удовлетворительно»: студент владеет основным объемом знаний по дисциплине; проявляет затруднения в самостоятельных ответах, оперирует неточными формулировками. В процессе ответов допускаются ошибки по существу вопросов. Студент способен решать лишь наиболее легкие задачи, владеет только обязательным минимумом методов исследований.

Оценка «неудовлетворительно»: студент обнаруживает незнание ответа на соответствующее задание, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно и неуверенно излагает материал, отмечаются такие недостатки в подготовке студента, которые являются серьезным препятствием к успешному овладению последующим материалом.

Критерии оценки устного опроса по дисциплине «Разработка мобильных приложений»

Устные опросы проводятся во время лекций, практических занятий и возможны при проведении промежуточной аттестации в качестве дополнительного испытания при недостаточности результатов тестирования. Основные вопросы для устного опроса доводятся до сведения студентов на предыдущем занятии.

Развернутый ответ студента должен представлять собой связное, логически последовательное сообщение на заданную тему, показывать его умение применять определения, правила в конкретных случаях.

По результатам ответа **«отлично»** выставляется студенту, если содержание ответа соответствует освещаемому вопросу, полностью раскрыта в ответе тема, ответ структурирован, даны правильные, аргументированные ответы на уточняющие вопросы, демонстрируется высокий уровень участия в дискуссии.

По результатам ответа **«хорошо»** выставляется студенту, если содержание ответа соответствует освещаемому вопросу, полностью раскрыта в ответе тема, даны правильные, аргументированные ответы на уточняющие вопросы, но имеются неточности, при этом ответ не структурирован и демонстрируется средний уровень участия в дискуссии.

По результатам ответа **«удовлетворительно»** выставляется студенту, если содержание ответа соответствует освещаемому вопросу, но при полном раскрытии темы имеются неточности, даны правильные, но не аргументированные ответы на уточняющие вопросы, демонстрируется низкий уровень участия в дискуссии, ответ не структурирован, информация трудна для восприятия.

По результатам ответа **«неудовлетворительно»** выставляется студенту, если содержание ответа соответствует освещаемому вопросу, но тема в ответе не полностью раскрыта, демонстрируется слабое владение категориальным аппаратом, происходит подмена понятий, даны неправильные, не аргументированные ответы на уточняющие вопросы, участие в дискуссии полностью отсутствует, ответ не структурирован, информация трудна для восприятия.

При несоответствии содержания ответа, освещаемому вопросу студент также получает **«неудовлетворительно»**.

Вопросы для подготовки к экзамену:

Тест 1

1. Что является основной задачей ОС в контексте управления ресурсами?
 - a) Управлять только вычислительными ресурсами
 - b) Эффективно и справедливо распределять ресурсы между программами
 - c) Защищать программы от вирусов
2. Что такое «планирование» (scheduling)?
 - a) Процесс выделения процессора определенной программе
 - b) Процесс создания новой программы
 - c) Процесс распределения памяти
3. Что является вычислительным ресурсом?
 - a) Жесткий диск
 - b) Процессорное время
 - c) Сетевая карта
4. Какой ресурс ОС выделяет каждому процессу, защищая его от других?
 - a) Процессорное время
 - b) Оперативная память
 - c) Сетевое соединение
5. Что такое «состояние гонки» (race condition)?
 - a) Ситуация, когда два потока пытаются получить доступ к ресурсу
 - b) Ситуация, когда несколько процессов пытаются одновременно изменить общую переменную
 - c) Ситуация, когда один процесс работает быстрее другого
6. Что является основным инструментом для предотвращения «состояния гонки»?
 - a) Механизм прерываний
 - b) Мьютекс (Mutex) или семафор
 - c) Страничный файл
7. Что такое «взаимная блокировка» (deadlock)?
 - a) Состояние, когда два или более процесса ждут друг друга, чтобы освободить ресурсы
 - b) Ошибка, которая приводит к "падению" системы
 - c) Состояние, когда процессор не может выполнить задачу
8. Что из перечисленного НЕ является одним из четырех условий возникновения «deadlock»?
 - a) Круговое ожидание
 - b) Взаимное исключение
 - c) Захват без ожидания
9. Что такое «страничная ошибка» (page fault)?
 - a) Ошибка, которая возникает при попытке записи на диск
 - b) Событие, которое происходит, когда программе нужна страница данных, которой нет в физической памяти
 - c) Ошибка при работе с файлом
10. Что происходит, когда возникает страничная ошибка?
 - a) Программа завершает свою работу
 - b) ОС загружает нужную страницу с диска в оперативную память
 - c) Программа продолжает работать без нужной страницы
11. Какой ресурс является «узким местом» (bottleneck) для большинства современных браузеров?
 - a) Процессор
 - b) Память
 - c) Жесткий диск

12. Какую проблему решает механизм прерываний?
 - a) Неэффективное использование ресурсов
 - b) Блокировку процессов
 - c) Взаимодействие с внешними устройствами и событиями
13. Что такое «мьютекс» (Mutex)?
 - a) Механизм, который обеспечивает монопольный доступ к общему ресурсу
 - b) Название файловой системы
 - c) Название алгоритма планирования
14. Какая из структур хранит метаданные файла (размер, права доступа, расположение на диске) в файловой системе, использующей inodes?
 - a) Блок данных
 - b) Inode
 - c) Корневой каталог
15. Что такое «вытеснение» (preemption) в контексте ОС?
 - a) Передача ресурса от одного процесса к другому без его добровольного освобождения
 - b) Процесс, при котором процесс добровольно освобождает ресурс
 - c) Удаление процесса из памяти
16. Что произойдет, если в программе два потока одновременно будут инкрементировать одну и ту же глобальную переменную без защиты мьютексом?
 - a) Программа завершится с ошибкой
 - b) Результат будет всегда корректным
 - c) Возникнет «состояние гонки», и итоговый результат будет непредсказуемым
17. Какой тип ресурсов включает жесткие диски и сетевые карты?
 - a) Вычислительные ресурсы
 - b) Ресурсы памяти
 - c) Устройства ввода-вывода
18. Почему эффективное управление ресурсами является критически важным для системы?
 - a) Оно напрямую влияет на стабильность и производительность
 - b) Оно нужно только для многопоточных программ
 - c) Оно упрощает работу программистов
19. Что происходит, если программа не может получить нужный ресурс?
 - a) Она будет ждать, пока ресурс не освободится
 - b) Она сразу завершится
 - c) Она продолжит работать без ресурса
20. Что, согласно лекции, является «архитектором цифровых систем»?
 - a) Программист
 - b) Операционная система
 - c) Аппаратное обеспечение

Ответы для проверки:

1. b
2. b
3. a
4. b
5. c
6. b
7. b
8. b
9. b
10. a

- 11. c
- 12. b
- 13. b
- 14. c
- 15. a
- 16. c
- 17. c
- 18. a
- 19. a
- 20. b

Тест 2

1. Что такое «программа» в контексте операционных систем?
 - a) Активная, динамическая сущность
 - b) Конкретный экземпляр программы, который выполняется в данный момент
 - c) Пассивная, статическая сущность, набор инструкций, хранящийся на диске
2. Что происходит, когда вы запускаете программу?
 - a) Операционная система удаляет программу с диска
 - b) Операционная система создает для нее процесс
 - c) Программа сразу же начинает потреблять ресурсы без создания процесса
3. Какая из перечисленных аналогий соответствует «процессу»?
 - a) Рецепт торта
 - b) Поваренная книга
 - c) Акт приготовления торта
4. Какой сегмент памяти процесса содержит скомпилированный машинный код программы?
 - a) Стек (Stack)
 - b) Сегмент кода (Text Segment)
 - c) Куча (Heap)
5. Какая область памяти используется для динамического выделения ресурсов во время выполнения программы?
 - a) Куча (Heap)
 - b) Стек (Stack)
 - c) Сегмент данных (Data Segment)
6. Что хранится в стеке (Stack)?
 - a) Глобальные переменные
 - b) Локальные переменные и адреса возврата функций
 - c) Скомпилированный код программы
7. По какому принципу работает стек?
 - a) FIFO (First-In, First-Out)
 - b) LIFO (Last-In, First-Out)
 - c) Random access
8. Как называется область памяти для неинициализированных глобальных переменных?
 - a) Сегмент кода (Text Segment)
 - b) BSS (Block Started by Symbol)
 - c) Стек (Stack)
9. Что является основной причиной «утечки памяти» (memory leak)?
 - a) Переполнение стека
 - b) Некорректное освобождение динамически выделенной памяти
 - c) Ошибка в сегменте кода

10. Что может произойти, если стек и куча «встретятся»?
 - a) Утечка памяти
 - b) Переполнение (Stack overflow / Heap collision)
 - c) Блокировка
11. Какой сегмент памяти является «только для чтения» (read-only)?
 - a) Сегмент данных (Data Segment)
 - b) Куча (Heap)
 - c) Сегмент кода (Text Segment)
12. Почему сегмент кода является «разделяемым» (shareable)?
 - a) Чтобы сэкономить ресурсы памяти, когда запущено несколько процессов одной программы
 - b) Чтобы обеспечить быстрый доступ к коду
 - c) Чтобы предотвратить изменение инструкций
13. Какой сегмент памяти «растет» от старших адресов к младшим?
 - a) Стек (Stack)
 - b) Куча (Heap)
 - c) Сегмент данных (Data Segment)
14. Какое из утверждений верно, согласно лекции?
 - a) Одна программа может порождать только один процесс
 - b) Процесс не потребляет системные ресурсы
 - c) Одна программа может порождать множество процессов
15. Какая из характеристик отличает процесс от программы?
 - a) Пассивная природа
 - b) Отсутствие жизненного цикла
 - c) Потребление системных ресурсов
16. Что такое «контекст выполнения» процесса?
 - a) Набор инструкций программы
 - b) Значения регистров, счетчик команд, стек
 - c) Имя исполняемого файла
17. Какой из сегментов памяти имеет фиксированный размер, известный на этапе компиляции?
 - a) Куча
 - b) Стек
 - c) Сегмент данных
18. Для чего используется BSS-сегмент?
 - a) Для хранения кода программы
 - b) Для хранения неинициализированных глобальных переменных
 - c) Для динамического выделения памяти
19. Что является результатом некорректной работы с кучей?
 - a) Переполнение стека
 - b) Утечка памяти
 - c) Ошибка компиляции
20. Что является главной задачей ОС в контексте управления процессами?
 - a) Создание и завершение процессов
 - b) Эффективное и справедливое распределение ресурсов
 - c) Защита программы от вирусов

Ответы для проверки:

1. c
2. b
3. c
4. b
5. a
6. b
7. b
8. b
9. b
10. b
11. a
12. b
13. a
14. c
15. c
16. b
17. c
18. b
19. b
20. b

Тест 3

1. Что является наименьшей единицей выполнения кода, которой может управлять ОС?
 - a) Процесс
 - b) Поток
 - c) Программа
2. Какое из утверждений верно, согласно лекции?
 - a) Один процесс может содержать несколько потоков
 - b) Один поток может содержать несколько процессов
 - c) Каждый поток имеет своё изолированное адресное пространство
3. Какой из ресурсов потоки одного процесса разделяют между собой?
 - a) Стек
 - b) Набор регистров
 - c) Адресное пространство
4. Какой из компонентов контекста выполнения является уникальным для каждого потока?
 - a) Открытые файлы
 - b) Счётчик команд
 - c) Глобальные переменные
5. Почему взаимодействие между потоками считается более простым, чем между процессами?
 - a) Потому что потоки выполняются последовательно
 - b) Потому что они используют общие данные в памяти
 - c) Потому что потоки изолированы друг от друга
6. В чём основное преимущество потоков перед процессами с точки зрения производительности?
 - a) Они быстрее создаются
 - b) Они потребляют больше памяти
 - c) Они могут напрямую обращаться к памяти других процессов

7. Что является «основным инструментом» для синхронизации доступа к общим данным?
 - a) Счётчик команд
 - b) Мьютекс
 - c) IPC
8. Как называется ситуация, когда несколько потоков пытаются одновременно изменить одну и ту же общую переменную?
 - a) Взаимная блокировка
 - b) Состояние гонки
 - c) Голодание
9. Что может случиться, если не использовать синхронизацию при работе с общими данными?
 - a) Программа начнёт работать быстрее
 - b) Результат вычислений будет непредсказуемым
 - c) Программа завершится с ошибкой
10. Какой из ресурсов является уникальным для каждого процесса?
 - a) Стек
 - b) Глобальные переменные
 - c) Адресное пространство
11. Как называется механизм, который позволяет процессам взаимодействовать друг с другом?
 - a) IPC (Inter-Process Communication)
 - b) API
 - c) POSIX
12. Какой из компонентов контекста выполнения потока хранит локальные переменные и адреса возврата из функций?
 - a) Регистры
 - b) Стек
 - c) Куча
13. Что такое «пул потоков» (thread pool)?
 - a) Набор потоков, которые выполняют задачи по очереди
 - b) Способ эффективного управления множеством потоков
 - c) Специальная область памяти для потоков
14. Почему потоки иногда называют «облегчёнными процессами»?
 - a) Они используют меньше памяти
 - b) Они создаются и переключаются быстрее
 - c) Оба варианта верны
15. Какое из утверждений является верным?
 - a) Процессы разделяют память, а потоки — нет.
 - b) Потоки разделяют память, а процессы — нет.
 - c) И процессы, и потоки разделяют одну и ту же память.
16. Что является основной задачей механизмов синхронизации?
 - a) Ускорение работы программы
 - b) Защита общих данных от одновременного доступа
 - c) Управление жизненным циклом потоков
17. Какой из перечисленных ресурсов не является уникальным для каждого процесса?
 - a) Адресное пространство
 - b) Стек
 - c) Дескрипторы файлов

18. Для чего нужен «мьютекс» (Mutex)?
 - a) Для ускорения работы программы
 - b) Для обеспечения монопольного доступа к общему ресурсу
 - c) Для создания нового потока
19. Что является примером многопоточного приложения?
 - a) Калькулятор
 - b) Браузер с несколькими вкладками
 - c) Текстовый редактор
20. Какой из компонентов контекста выполнения потока указывает на следующую инструкцию для выполнения?
 - a) Счётчик команд (Instruction Pointer)
 - b) Набор регистров
 - c) Стек

Ответы для проверки:

1. b
2. a
3. c
4. c
5. a
6. b
7. b
8. a
9. a
10. c
11. b
12. b
13. b
14. c
15. b
16. b
17. b
18. a
19. a
20. a

Тест 4

1. Что такое «разделение времени» (time-sharing)?
 - a) Способ одновременного выполнения нескольких задач на одном ядре.
 - b) Переключение процессора между задачами.
 - c) Запуск нескольких программ на разных компьютерах.
2. Какой из перечисленных феноменов привел к появлению многоядерных архитектур?
 - a) Закон Мура.
 - b) Тепловой барьер.
 - c) Необходимость создания более сложных программ.

3. В чем состоит основное различие между параллелизмом и конкурентностью?
- а) Параллелизм — это фактическое одновременное выполнение, а конкурентность — это управление несколькими задачами, которые могут выполняться параллельно.
 - б) Конкурентность — это выполнение задач на одном ядре, а параллелизм — на нескольких.
 - в) Эти термины являются синонимами.
4. Как называется ситуация, когда несколько потоков одновременно пытаются получить доступ к общему ресурсу?
- а) Взаимная блокировка.
 - б) Состояние гонки.
 - в) Синхронизация.
5. Что такое «мьютекс» (Mutex)?
- а) Механизм, обеспечивающий монополярный доступ к ресурсу.
 - б) Название одного из законов производительности.
 - в) Специальный вид потока.
6. Что является одним из необходимых условий для возникновения «deadlock»?
- а) Использование нескольких потоков.
 - б) Отсутствие синхронизации.
 - в) Круговое ожидание.
7. Какой паттерн помогает избежать состояния гонки между двумя потоками, один из которых пишет данные, а другой — читает?
- а) Производитель-Потребитель.
 - б) Читатели-Писатели.
 - в) Клиент-Сервер.
8. Что описывает «закон Амдала»?
- а) Максимально достижимое ускорение программы при добавлении большего количества процессоров.
 - б) Количество потоков, которые могут работать одновременно.
 - в) Эффективность использования оперативной памяти.
9. Что такое «голодание» (starvation)?
- а) Состояние, когда потоку не хватает ресурсов для выполнения.
 - б) Ситуация, когда поток не может получить доступ к ресурсу из-за несправедливого распределения.
 - в) Недостаток памяти в системе.
10. Что такое «атомарная операция»?
- а) Операция, которая выполняется за один такт процессора.
 - б) Операция, которая не может быть прервана другим потоком.
 - в) Операция, которая работает с атомами.
11. Что такое «пул потоков» (thread pool)?
- а) Набор предварительно созданных потоков, готовых выполнять задачи.
 - б) Набор задач, которые ожидают выполнения.
 - в) Механизм для межпроцессного взаимодействия.
12. Почему параллельный код в первую очередь должен быть корректным?
- а) Потому что некорректный код может привести к непредсказуемым результатам.
 - б) Потому что это требование для сертификации.
 - в) Потому что так быстрее.
13. Что такое «конкурентность» (concurrency)?
- а) Выполнение нескольких задач одновременно.
 - б) Борьба потоков за ресурсы.
 - в) Управление несколькими задачами, которые могут выполняться параллельно.

14. Какой закон гласит, что масштабируемость программы зависит от размера задачи?
 - a) Закон Мура.
 - b) Закон Амдала.
 - c) Закон Густавсона.
15. Какой из подходов к управлению потоками является наиболее высокоуровневым?
 - a) Работа с мьютексами.
 - b) Использование пулов потоков.
 - c) Использование атомарных операций.
16. Что является ключевым аспектом в управлении потоками?
 - a) Скорость работы.
 - b) Синхронизация.
 - c) Размер кода.
17. Что такое «взаимная блокировка» (deadlock)?
 - a) Ошибка, когда два потока пытаются получить доступ к одному и тому же ресурсу.
 - b) Ситуация, когда два или более потока ждут друг друга, чтобы освободить ресурс.
 - c) Завершение работы программы из-за ошибки.
18. Какой из законов утверждает, что масштабируемость программы зависит от её последовательной части?
 - a) Закон Густавсона.
 - b) Закон Амдала.
 - c) Закон Мура.
19. Что является примером «разделяемого ресурса»?
 - a) Локальная переменная.
 - b) Стек.
 - c) Глобальная переменная.
20. Что, согласно лекции, является «ловушкой» для параллельного программиста?
 - a) Взаимные блокировки и голодание.
 - b) Написание кода.
 - c) Отсутствие документации.

Ответы для проверки:

1. a
2. b
3. a
4. b
5. a
6. c
7. a
8. b
9. a
10. a
11. c
12. a
13. c
14. a
15. b
16. b
17. b
18. c
19. a
20. a