

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Колледж



ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

профессионального модуля

ПМ.01. РАЗРАБОТКА МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
ДЛЯ КОМПЬЮТЕРНЫХ СИСТЕМ

09.02.07 Информационные системы и программирование

Махачкала 2021

Составитель / составители:

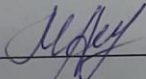
Шамсутдинова У.А. – преподаватель кафедры естественнонаучных и гуманитарных дисциплин колледжа ДГУ ФГБОУ ВО «Дагестанский государственный университет»

Изиева З.А. – преподаватель кафедры естественнонаучных и гуманитарных дисциплин Колледжа ДГУ ФГБОУ ВО «Дагестанский государственный университет»

Шахбанова М.И. - преподаватель кафедры естественно-научных и гуманитарных дисциплин колледжа ФГБОУ ВО «Дагестанский государственный университет

Фонд оценочных средств рассмотрен и рекомендован к утверждению на заседании кафедры специальных дисциплин колледжа ДГУ

Протокол № 1 от «31» 08 2021г.

Зав. кафедрой  /Магомедова А.М./

**ПАСПОРТ
ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ
по дисциплине**

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

	Контролируемые разделы, темы, модули	Код контролируемой компетенции	Наименование оценочного средства
МДК. 01.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ			
1.	Раздел 1. Разработка программных модулей (3 курс 5 семестр)	ОК 1; ОК25; ОК3; ОК4; ОК5; ОК6, ОК7, ОК8, ОК9, ОК10.	коллоквиум; тестирование; практическая работа
2.	Раздел 2 Разработка программных модулей (3 курс 6 семестр)	ПК 1.1; ПК 1.2	коллоквиум; тестирование; практическая работа
3.	Раздел 3. Разработка программных модулей (4 курс 7 семестр)	ОК 1; ОК25; ОК3; ОК4; ОК5; ОК6, ОК7, ОК8, ОК9, ОК10. ПК 1.1; ПК 1.2	коллоквиум; тестирование; практическая работа
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.	Деловая и/или ролевая игра	Совместная деятельность группы обучающихся и педагогического работника под управлением педагогического работника с целью решения учебных и профессионально-ориентированных задач путем игрового моделирования реальной проблемной ситуации. Позволяет оценивать умение анализировать и решать типичные профессиональные задачи.	Тема (проблема), концепция, роли и ожидаемый результат по каждой игре
2.	Кейс-задача	Проблемное задание, в котором обучающемуся предлагают осмыслить реальную профессионально-ориентированную ситуацию, необходимую для решения данной проблемы.	Задания для решения кейс-задачи
3.	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
4.	Контрольная работа	Средство проверки умений применять полученные знания для решения задач определенного типа по теме или разделу.	Комплект контрольных заданий по вариантам
5.	Круглый стол, дискуссия, полемика, диспут, дебаты	Оценочные средства, позволяющие включить обучающихся в процесс обсуждения спорного вопроса, проблемы и оценить их умение аргументировать собственную точку зрения.	Перечень дискуссионных тем.
6.	Портфолио	Целевая подборка работ студента, раскрывающая его индивидуальные образовательные достижения в одной или нескольких учебных дисциплинах.	Структура портфолио
7.	Рабочая тетрадь	Дидактический комплекс, предназначенный для самостоятельной работы обучающегося и позволяющий оценивать уровень усвоения им учебного материала.	Образец рабочей тетради

8.	Проект	Конечный продукт, получаемый в результате планирования и выполнения комплекса учебных и исследовательских заданий. Позволяет оценить умение обучающихся самостоятельно конструировать свои знания в процессе решения практических задач и проблем, ориентироваться в информационном пространстве и уровень сформированное аналитических, исследовательских навыков, навыков практического и творческого мышления. Может выполняться в индивидуальном порядке или группой обучающихся.	Темы групповых и/или индивидуальных проектов
9.	Разно-уровневые задачи и задания	<p><i>Различают задачи и задания:</i></p> <ul style="list-style-type: none"> – репродуктивного уровня, позволяющие оценивать и диагностировать знание фактического материала (базовые понятия, алгоритмы, факты) и умение правильно использовать специальные термины и понятия, узнавание объектов изучения в рамках определенного раздела дисциплины; – реконструктивного уровня, позволяющие оценивать и диагностировать умения синтезировать, анализировать, обобщать фактический и теоретический материал с формулированием конкретных выводов, установлением причинно-следственных связей; – творческого уровня, позволяющие оценивать и диагностировать умения, интегрировать знания различных областей, аргументировать собственную точку зрения. 	Комплект разно-уровневых задач и заданий
10.	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
11.	Реферат	Продукт самостоятельной работы студента, представляющий собой краткое изложение в письменном виде полученных результатов теоретического анализа определенной научной (учебно-исследовательской) темы,	Темы рефератов

		где автор раскрывает суть исследуемой проблемы, приводит различные точки зрения, а также собственные взгляды на нее.	
12.	Доклад, сообщение	Продукт самостоятельной работы студента, представляющий собой публичное выступление по представлению полученных результатов решения определенной учебно-практической, учебно-исследовательской или научной темы.	Темы докладов, сообщений
13.	Собеседование	Средство контроля, организованное как специальная беседа педагогического работника с обучающимся на темы, связанные с изучаемой дисциплиной, и рассчитанное на выяснение объема знаний обучающегося по определенному разделу, теме, проблеме и т.п.	Вопросы по
14.	Тест	Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося	Фонд тестовых заданий

КРИТЕРИИ ОЦЕНКИ

по дисциплине

МДК. 01.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

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.

Вопросы по дисциплине «Разработка программных модулей»:

6 семестр

1. Особенности создания программного продукта. Проблематика проектирования. Понятие технологии разработки программы. Модели жизненного цикла.

2. Унифицированный язык моделирования (UML). Определение прецедентов (вариантов использования). Построение концептуальной модели предметной области. Описание поведения системы с помощью диаграмм последовательности действий и диаграмм взаимодействий. 3. Определение функциональных и эксплуатационных требований к программным продуктам при структурном и объектном подходе. Словарь терминов. Функциональные диаграммы. Диаграммы потоков данных (DFD).

4. Проектирование программного обеспечения при структурном подходе. Структурная схема разрабатываемого программного обеспечения (логическая и физическая модели) Метод пошаговой детализации при составлении алгоритмов.

5. Пользовательские типы данных: структуры (записи), объединения, перечисления, множества. Синтаксис объявления пользовательских типов данных. Доступ к элементу пользовательского типа.

6. Основы файловой системы. Порядок работы с текстовыми и двоичными файлами. Функции для работы с файлами разного типа. Правила бесформатного и форматного ввода/вывода.

7. Разделение программы на подпрограммы. Параметры подпрограммы. Перегрузка подпрограмм. Группировка подпрограмм и связанных с ними данных в отдельные файлы - модули. Основные характеристики программного модуля. Модульная структура программных продуктов.

8. Проектирование программного обеспечения при объектном подходе. Структура проекта в объектно-ориентированной среде программирования. Событийно-управляемая модель программирования. Компоненты объектно-ориентированной среды. Принципы разработки графического интерфейса. Построение функциональной блок-схемы приложения. Особенности ввода/вывода в объектно-ориентированной среде программирования. Виды строк, функции преобразования типов.

9. Классы. Конструкторы и деструкторы. Наследование. Создание наследованного класса. Статический и динамический полиморфизм (виртуализация методов). Дружественные функции и классы. Перегрузка методов класса и конструкторов.

10. Механизм обработки и генерирования исключительных ситуаций. Способы создания DLL-библиотеки.

7 семестр

11. Серия стандартов ISO 9000. Процесс сертификации программ на базе информации об их использовании.

12. Виды программных документов. Пояснительная записка. Руководство пользователя. Руководство системного программиста.

13. Тестирование методами «белого ящика» и «черного ящика». Порядок разработки тестов. Автоматизация тестирования. Модульное, интеграционное, системное тестирование.

14. Отладка программ. Автономная и комплексная отладка программ. Инструментальные средства отладки программ.

15. Основные принципы построения приложений, основанных на СУБД. Компоненты объектно-ориентированной среды, необходимые для организации базы данных и работы с ней.

16. Принцип разработки программы, управляющей локальной базой данных. Способы навигации по набору данных.

17. Вставка, добавление и удаление записей таблиц базы данных. Поиск, фильтрация и сортировка данных. Возможности среды для создания новых полей.

18. Принцип разработки программы, управляющей удаленной базой данных. SQL-запросы на выборку, добавление, редактирование и удаление данных. Агрегатные функции. Группировка записей.

19. Статические и динамические параметрические SQL-запросы.

Список тем практических работ по дисциплине «Разработка программных модулей»:

6 семестр

1. Создание функциональных диаграмм.
2. Разработка диаграмм потоков данных.
3. Построение концептуальной модели предметной области.
4. Описание поведения системы с помощью диаграмм прецедентов и действий.
5. Построение структурной схемы разрабатываемого программного обеспечения
6. Реализация метода пошаговой детализации при составлении алгоритмов с помощью ER-диаграмм
7. Разработка программ со структурированными типами данных.
8. Разработка усложненных программ со структурированными типами данных.
9. Разработка программ с использованием перечисляемого типа.
10. Работа с данными типа множество.
11. Использование криптографических методов защиты информации.
12. Работа с файлом последовательного доступа.
13. Работа с двоичными файлами.
14. Разработка программ с чтением и записью файлов разных типов.
15. Организация файлового ввода/вывода.

7 семестр

16. Создание подпрограмм.
17. Использование подпрограмм.
18. Область видимости переменных.

19. Перегрузка функций.
20. Программирование модуля.
21. Создание простого проекта.
22. Создание проекта с использованием функций преобразования типов.
23. Использование визуальных и не визуальных компонентов RAD-среды.
24. Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени.
25. Разработка структуры программного обеспечения при объектном подходе.
26. Создание проекта с использованием компонентов стандартных диалогов и системы меню.
27. Разработка оконного приложения.
28. Разработка оконного приложения с несколькими формами.
29. Разработка приложения с использованием строк и функций преобразования типов.
30. Объявление класса, создание экземпляров класса.
31. Инициализация членов класса с помощью конструктора с параметрами.
32. Создание наследованного класса.
33. Перегрузка методов.
34. Виртуализация методов.
35. Обработка исключительных ситуаций.
36. Генерирование исключительных ситуаций.
37. Разработка библиотеки подпрограмм DLL.
38. Создание библиотеки подпрограмм DLL.
39. Разработка технического задания на программный продукт.
40. Разработка модульного программного продукта.
41. Разработка структуры программного документа «Пояснительная записка».
42. Разработка структуры программного документа «Руководство пользователя».
43. Организация модульного тестирования методом «белого ящика».
44. Организация модульного тестирования методом «черного ящика».
45. Применение инструментальных средств отладки программ.
46. Создание приложения для работы с базой данных с использованием технологии ADO.
47. Фильтрация и поиск данных.
48. Программный способ обработки информации базы данных.
49. Работа со связными таблицами базы данных.
50. Статические запросы на выборку данных с использованием агрегатных функций.
51. Статические запросы на выборку данных с использованием группировки и подзапросов.
52. Статические параметрические запросы.
53. Динамические запросы на выборку и сортировку данных.
54. Динамические запросы, реализующие вставку и удаление данных.

55. Динамические параметрические запросы

СТРУКТУРА ИТОГОВОГО ТЕСТА:

Тест содержит 20 вопросов случайным образом выбранных из списка. Тест проводится на персональном компьютере в оболочке для тестирования MyTest. Результат выдается сразу после тестирования и формируется отчет протестированных студентов на сервере.

2.1.3 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 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 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 программных модулей»:

1. Независимость программных продуктов от технического комплекса системы обработки данных, операционной среды, сетевой технологии обработки данных, специфики предметной области и т.п., означает их ...
 - а) модифицируемость
 - б) надежность
 - в) мобильность
 - г) эффективность
2. Бесбойность и устойчивость в работе программных продуктов, точность выполнения предписанных функций обработки, возможность диагностики возникающих в процессе работы программ ошибок, означает их ...
 - а) амодифицируемость
 - б) надежность
 - в) мобильность
 - г) эффективность
3. Возможность применения алгоритма к целому классу однотипных задач, различающихся конкретным значением, является одним из основных свойств алгоритма ...
 - а) детерминированностью
 - б) дискретностью
 - в) определенностью
 - г) массовостью
4. Максимально возможная интеграция программных продуктов с другими программами, обеспечение обмена данными в общих форматах представления (экспорт/импорт баз данных, внедрение или связывание объектов обработки и др.) означает их ...
 - а) модифицируемость
 - б) эффективность
 - в) мобильность
 - г) коммуникативность
5. Способность программных продуктов к внесению изменений, например расширение функций обработки, переход на другую техническую базу обработки и т.п., означает их ...
 - а) модифицируемость
 - б) эффективность

- в) мобильность
 - г) учет человеческого фактора
6. Наличие дружественного интерфейса, контекстно-зависимой подсказки, хорошей документации является таким показателем качества программного продукта, как ...
- а) коммуникативность
 - б) эффективность
 - в) мобильность
 - г) учет человеческого фактора
7. В соответствии с ГОСТ 19.102-77 детальное описание действий в виде:
- а) постановки задачи
 - б) выбора критериев эффективности
 - в) проведения предварительных научно-исследовательских работ
8. Разработки технического задания, определяет содержание этапа модели ЖЦ.
- а) эскизный проект
 - б) технический проект
 - в) техническое задание
 - г) рабочий проект
9. В соответствии с ГОСТ 19.102-77 детальное описание действий в виде:
- а) структуры входных и выходных данных
 - б) уточнения методов решения
 - в) общего алгоритма
10. Разработки документации эскизного проекта, определяет содержание этапа модели ЖЦ ...
- а) эскизный проект
 - б) технический проект
 - в) техническое задание
 - г) рабочий проект
11. В соответствии с ГОСТ 19.102-77 детальное описание действий в виде:
- а) уточнения структуры входных и выходных данных
 - б) разработки алгоритмов
 - в) форм данных, - семантики и синтаксиса языка
 - г) структуры программы
12. Конфигурации технических средств, плана работ, определяет содержание этапа модели ЖЦ ...
- а) эскизный проект
 - б) технический проект
 - в) техническое задание
 - г) рабочий проект
13. В соответствии с ГОСТ 19.102-77 детальное описание действий в виде: - программирования и отладки; - разработки документов; - подготовки и проведения испытаний; - корректировки программы и документов по итогам испытаний определяет содержание этапа модели ЖЦ ...
- а) эскизный проект

- б) внедрение
 - в) техническое задание
 - г) рабочий проект
14. Наличие обратных связей между этапами жизненного цикла, определяющих возможность проведения проверок и корректировок проектируемого ПС на каждой стадии разработки, является основной особенностью ...
- а) каскадной модели жизненного цикла
 - б) итерационной модели жизненного цикла
 - в) спиральной модели жизненного цикла ПО
 - г) модели жизненного цикла UML
15. Основными особенностями: - последовательным выполнением входящих в состав ЖЦ этапов,
- а) окончанием каждого предыдущего этапа до начала последующего
 - б) отсутствием временного перекрытия этапов
 - в) отсутствием возврата к предыдущим этапам
16. Наличием результата только в конце разработки, характеризуется ...
- а) каскадная модель жизненного цикла
 - б) итерационная модель жизненного цикла
 - в) спиральная модель жизненного цикла ПО
 - г) модель жизненного цикла UML
17. Выявлением и устранением ошибок только на стадии тестирования, которая может растянуться во времени или вообще никогда не завершиться, характеризуется ...
- а) каскадная модель жизненного цикла
 - б) итерационная модель жизненного цикла
 - в) спиральная модель жизненного цикла ПО
 - г) модель жизненного цикла UML
18. Функционирование программного продукта в соответствии с техническим заданием соответствует характеристике эксплуатационных требований ...
- а) правильность
 - б) универсальность
 - в) надежность
 - г) адаптируемость
19. Обеспечение правильной работы при любых допустимых данных и защиты от неправильных данных соответствует характеристике эксплуатационных требований ...
- а) правильность
 - б) универсальность
 - в) надежность
 - г) защищенность
20. Обеспечение правильности результатов при наличии различного рода сбоев соответствует характеристике эксплуатационных требований ...
- а) правильность
 - б) универсальность

- в) надежность
 - г) защищенность
21. Обеспечение погрешности результатов не выше заданной соответствует характеристике эксплуатационных требований ...
- а) правильность
 - б) проверяемость
 - в) точность результатов
 - г) защищенность
22. Использование минимально возможного количества ресурсов технических средств (например, времени микропроцессора, объема оперативной памяти, объема внешней памяти, количества внешних устройств и др.) соответствует характеристике эксплуатационных требований ...
- а) универсальность
 - б) надежность
 - в) аппаратная совместимость
 - г) эффективность
23. Возможность быстрой модификации с целью приспособления к изменяющимся условиям функционирования соответствует характеристике эксплуатационных требований ...
- а) универсальность
 - б) эффективность
 - в) адаптируемость
 - г) рентабельность
24. Возможность «параллельного» использования несколькими процессами соответствует характеристике эксплуатационных требований ...
- а) универсальность
 - б) программная совместимость
 - в) повторная входимость
 - г) рентабель
25. Функциональная диаграмма имеет аббревиатуру ...
- а) ERD
 - б) SADT
 - в) DFD
 - г) STD
26. Диаграмма потоков данных имеет аббревиатуру ...
- а) ERD
 - б) SADT
 - в) DFD
 - г) STD
27. Диаграмма переходов состояний имеет аббревиатуру ...
- а) ERD
 - б) SADT
 - в) DFD
 - г) STD

28. Диаграмма «сущность-связь» имеет аббревиатуру ...
- а) ERD
 - б) SADT
 - в) DFD
 - г) STD
29. Вершиной древовидной структуры диаграмм, представляющей собой самое общее описание системы и ее взаимодействия с внешней средой, является следующий вид диаграммы IDEF0 ...
- а) диаграмма декомпозиции
 - б) диаграмма дерева узлов
 - в) контекстная диаграмма
 - г) диаграмма для экспозиции
30. Спецификация разрабатываемого программного обеспечения при использовании языка визуального моделирования UML объединяет несколько моделей. Содержит описание функций программного обеспечения с точки зрения пользователя ...
- а) модель использования
 - б) логическая модель
 - в) модель реализации
 - г) модель развертывания
31. Спецификация разрабатываемого программного обеспечения при использовании языка визуального моделирования UML объединяет несколько моделей. Описывает ключевые понятия моделируемого программного обеспечения (классы, интерфейсы и т. п.), т. е. средства, обеспечивающие его функциональность ...
- а) логическая модель
 - б) модель реализации
 - в) модель процессов
 - г) модель развертывания
32. Спецификация разрабатываемого программного обеспечения при использовании языка визуального моделирования UML объединяет несколько моделей. Определяет реальную организацию программных модулей в среде разработки ...
- а) модель использования
 - б) модель реализации
 - в) модель процессов
 - г) модель развертывания
33. Спецификация разрабатываемого программного обеспечения при использовании языка визуального моделирования UML объединяет несколько моделей. Отображает организацию вычислений и позволяет оценить производительность, масштабируемость и надежность программного обеспечения ...
- а) модель использования
 - б) модель реализации
 - в) модель процессов

- г) модель развертывания
34. Спецификация разрабатываемого программного обеспечения при использовании языка визуального моделирования UML объединяет несколько моделей. Показывает, каким образом программные компоненты размещаются на конкретном оборудовании ...
- а) модель использования
 - б) модель реализации
 - в) модель процессов
 - г) модель развертывания
35. Служит связующим звеном между процессами, т. е. позволяет описать порядок взаимодействия процессов ...
- а) нотация
 - б) прецедент
 - в) поток событий
 - г) сценарий
36. Подробное процедурное описание вариантов использования системы всеми заинтересованными лицами, а также внешними системами, реализует ...
- а) нотация
 - б) прецедент
 - в) поток событий
 - г) сценарий
37. Идентификатором, сформированным по всем правилам языка C, можно назвать выражение ...
- а) lizonьka
 - б) lizo4ka_
 - в) no_naroshky
 - г) egoza-liza
38. Для описания основных (стандартных) типов данных на языке C определены следующие ключевые слова ...
- а) int, char, bool, double, void
 - б) int, char, float, double, bool
 - в) int, wchar_t, float, double, char
 - г) int, char, float, double, void
39. Символьная константа на языке C может быть представлена в виде ...
- а) 'Q1'
 - б) '<='
 - в) ';'
 - г) "z"
40. Десятичная константа на языке C может быть представлена в виде ...
- а) 020
 - б) 101
 - в) 01
 - г) 0x20

7 семестр

1. Верным на языке C является следующее объявление и инициализация переменных ...
 - а) `double n, k=3;`
 - б) `float n, k, int z=-1;`
 - в) `char a, b='No';`
 - г) `float *t=4;`
2. Строковой константой на языке C считается следующее выражение...
 - а) `a"l"v"f"`
 - б) `"Mus%tika!"`
 - в) `'Slonik'`
 - г) `"Slovo_no"1"1`
3. Идентификатором не является ...
 - а) имя объекта
 - б) название свойства
 - в) зарезервированное слово
 - г) имя поля в структуре
4. Верным на языке C является следующее объявление и инициализация переменных ...
 - а) `short char k, z='б';`
 - б) `float k, *m=&k;`
 - в) `short signed int c, m=1.0;`
 - г) `long int my='0', 8c=3;`
5. Наполнить ячейку памяти значением призван оператор ...
 - а) неполного ветвления
 - б) полного ветвления
 - в) присваивания
 - г) цикла
6. Фрагмент кода программы на языке C `char *s, c; int i=0, j=0; ...`
`while (s[i]){ if (s[i]!=c)`
`s[j++]=s[i]; i++; }`
`s[j]='\0'; ...`
 - а) меняет местами символы в строке
 - б) заменяет символ на нулевой байт
 - в) вставляет символ, если такого нет в строке
 - г) удаляет символ из строки
7. Функция на языке C
`void func1(int *a, int k) {`
`int i, j, m;`
`for (i = 0; i < k; i++) a[i]=i;`
`m=2;`
`while (m<k) {`
`if (a[m]!=0) {`

```

j=m*2;
while (j<k){
a[j]=0;
j=j+m;
}
m=m+1;
} возвращает k-элементный массив, в котором кроме нулей содержатся...

```

- а) четные числа
- б) нечетные числа
- в) простые числа
- г) четные удвоенные числа

8. Функция на языке C void

```

func(int **a, int n, int m) {
int i, j, k, n1;
if (n%2==0) n1=n/2;
else n1=n/2+1;
printf("n1 = %d", n1);
for (j = 0; j < m; j++) { k = 0;
for (i = n-1; i >=n1; i--)
{ a[i][j] = a[i][j] + a[k][j];
a[k][j] = a[i][j] - a[k][j];
a[i][j] = a[i][j] - a[k][j]; k+=1; } }
} возвращает двумерный массив, в котором ...

```

- а) изменен порядок следования элементов каждой строки на обратный
- б) изменен порядок следования двух рядом стоящих элементов массива
- в) изменен порядок следования элементов каждого столбца на обратный
- г) изменен порядок следования двух рядом стоящих элементов каждого столбца

9. После выполнения фрагмента кода программы char

```

ZNAC= '@';
int x=1, y=-1, z=2;
switch (ZNAC) {
case '~': x = x + y + z;
case '@': z = 2*y - 3*x; case
'7': x = 5*y - z; break; case
'a': y = x + 3*(y - z); case
'*': x = y * z; case '#': x = y - x - z; break;
default : x = y - 1; } переменная x будет иметь значение...

```

- а) -3
- б) -16
- в) -2
- г) 0

10. После выполнения следующего программного кода

```

int i, j, n = 0, sqrs[4][3] = { 1, 8, 2, 4, 3, 9, 4, 6, 5, 2, 6, 7 };
for (i = 0; i<4; i++) for (j=0; j<3; j++)

```

if (sqr[i][j]%2==0&& i%2!=0) n+=sqr[i][j]; пользователь увидит, что переменная n будет содержать

- а) сумму четных элементов нечетных столбцов двумерного массива
- б) сумму четных элементов нечетных строк двумерного массива
- в) сумму четных элементов четных строк двумерного массива
- г) сумму нечетных элементов четных столбцов двумерного массива

11. Прототип функции

char isn(int k, char *r, float *m);

позволяет обратиться к функции с помощью фрагмента программы ...

- а) int a = -2; char s[10] = "Virtual&", simb; float n = 6.1; *simb = isn(a, &s, &n);
- б) int a = -2; char s[10] = "Virtual%", simb; float n = 6.1; simb = isn(a, *s, *n);
- в) int a = -2; char s[10] = "Virtual3", simb; float n = 6.1; simb = isn(a, s, &n);
- г) int a = -2; char s[10] = "Virtual4", simb; float n = 6.1; simb = isn(a, &s, &n);

12. При обращении к функции в соответствии с требованиями хорошего стиля программирования рекомендуется (но не обязательно), чтобы ...

- а) имена формальных и фактических параметров были различными
- б) количество формальных параметров соответствовало количеству фактических параметров
- в) типы формальных параметров соответствовали типам фактических параметров
- г) порядок перечисления фактических параметров соответствовал порядку перечисления формальных параметров

13. В объектно-ориентированной среде программирования обработчики любых событий для компонентов размещаются в окне ...

- а) формы
- б) инспектора объектов
- в) кода
- г) дерева объектов

14. В объектно-ориентированной среде программирования к любому компоненту в программе можно обратиться по его свойству ...

- а) Caption
- б) Text
- в) Align
- г) Name

15. Запись вещественного числа aEb означает ...

- а) число a умножить на b в степени 10
- б) число b умножить на 10 в степени a
- в) число a умножить на 10 степени b
- г) число b умножить на a степени 10

16. Переменные, которые передаются подпрограмме в момент ее вызова из основной программы, называются ...

- а) формальными параметрами
- б) глобальными переменными
- в) фактическими параметрами

- г) локальными переменными
17. Переменные, которые определяются в заголовке подпрограммы при её объявлении и описании, называются ...
- а) формальными параметрами
 - б) глобальными переменными
 - в) фактическими параметрами
 - г) локальными переменными
18. Получить наименования всех отпущенных со склада товаров можно с помощью SQL-оператора...
- а) `SELECT TOVAR FROM RASHOD`
 - б) `SELECT DISTINCT TOVAR FROM RASHOD`
 - в) `SELECT * FROM RASHOD ORDER BY TOVAR`
 - г) `SELECT TOVAR FROM RASHOD WHERE TOVAR = :PARAM1`
19. Общее количество отпуска по каждому из товаров можно получить с помощью SQL оператора ...
- а) `SELECT R.TOVAR, SUM(R.KOLVO) AS OTPUSK FROM RASHOD R GROUP BY R.TOVAR`
 - б) `SELECT DAT_RASH, COUNT(DISTINCT POKUP) FROM RASHOD GROUP BY DAT_RASH`
 - в) `SELECT R.TOVAR, SUM(R.KOLVO * T.ZENA) FROM RASHOD R, TOVARY T WHERE T.TOVAR = R.TOVAR GROUP BY R.TOVAR`
 - г) `SELECT R.TOVAR, R.DAT_RASH, SUM(R.KOLVO * T.ZENA)`
 - д) `FROM RASHOD R, TOVARY T WHERE T.TOVAR = R.TOVAR GROUP BY R.TOVAR, R.DAT_RASH`
20. Общая цена каждого из отпущенных товаров определяется с помощью SQL оператора ...
- а) `SELECT R.TOVAR, SUM(R.KOLVO * T.ZENA) FROM RASHOD R, TOVARY T WHERE T.TOVAR = R.TOVAR GROUP BY R.TOVAR`
 - б) `SELECT R.TOVAR, SUM(R.KOLVO) AS OTPUSK FROM RASHOD R GROUP BY R.TOVAR`
 - в) `SELECT DAT_RASH, COUNT(DISTINCT POKUP) FROM RASHOD GROUP BY DAT_RASH`
 - г) `SELECT R.TOVAR, R.DAT_RASH, SUM(R.KOLVO * T.ZENA) FROM RASHOD R, TOVARY T WHERE T.TOVAR = R.TOVAR GROUP BY R.TOVAR, R.DAT_RASH`
- а) сумму отпущенного товара по каждому виду товара для тех покупок, в которых минимальное количество покупаемого товара не более 5000 единиц
 - б) общее количество отпущенного товара по каждому виду товара для тех покупок, в которых минимальное количество покупаемого товара не менее 5000 единиц.
 - в) сумму отпущенного товара по каждому виду товара для тех покупок, в которых минимальное количество покупаемого товара не менее 5000 единиц
 - г) количество отпущенного товара для тех покупок, в которых

максимальное количество покупаемого товара не менее 5000 единиц

21. SQL-оператор `SELECT R.TOV, SUM(KOLVO*T.ZENA) AS STOIM FROM RASHOD R, TOVARY T WHERE R.TOV = T.TOVAR GROUP BY R.TOV HAVING COUNT(R.TOV)>3`; определяет ...

- а) общую стоимость каждого из отпущенных товаров, количество отпуска которого более 3-х раз
- б) сумму отпущенного товара по каждому виду товара, количество отпуска которого более 3-х раз
- в) количество отпущенного товара по каждому виду товара, количество отпуска которого не более 3-х раз
- г) общую стоимость каждого из отпущенных товаров, количество отпуска которого не более 3-х раз

22. Абстрактным называется класс, который ...

- а) не содержит функций
- б) не содержит переменных
- в) содержит хотя бы одну виртуальную функцию
- г) содержит хотя бы одну чисто виртуальную функцию

23. Код программы на языке C++

```
template <class> void swapargs(X &a, X &b)
```

```
{ X temp; temp = a; a = b; b = temp; }
```

 содержит ...

- а) обобщенную функцию, меняющую местами две переменные
- б) шаблон, обнуляющий две переменные
- в) виртуальную функцию, меняющую местами две переменные
- г) функцию, обнуляющую две переменные

24. Код `Query1->Close()`;

```
Query1->SQL->Clear();
```

```
Query1->SQL->Add ("select TOVAR, ED_IZM, ZENA");
```

```
Query1->SQL->Add ("from TOVARY");
```

```
Query1->SQL->Add ("order by TOVAR");
```

```
Query1->Open();
```

 определяет запрос ...

- а) статический
- б) динамический
- в) статический параметрический
- г) динамический параметрический

25. Код

```
Query1->Close();
```

```
Query1->SQL->Clear();
```

```
Query1->SQL->Add ("select TOVAR, ED_IZM, ZENA");
```

```
Query1->SQL->Add ("from TOVARY");
```

```
Query1->SQL->Add ("where TOVAR = :TOVAR ");
```

```
Query1->Open();
```

 определяет запрос ...

- а) статический
- б) динамический
- в) статический параметрический
- г) динамический параметрический

26. Модульное тестирование предполагает ...

- а) тестирование минимально возможного для тестирования компонента, например отдельного класса или функции
- б) поиск каких-либо проблем в интерфейсах и взаимодействии между компонентами программы
- в) тестирование интегрированной системы на ее соответствие исходным требованиям
- г) имитацию реальной работы с системой штатными разработчиками либо реальной работы с системой потенциальными пользователями/заказчиком на стороне разработчика

27. Интеграционное тестирование предполагает ...

- а) тестирование минимально возможного для тестирования компонента, например отдельного класса или функции
- б) поиск каких-либо проблем в интерфейсах и взаимодействии между компонентами программы
- в) тестирование интегрированной системы на ее соответствие исходным требованиям
- г) распространение версии с ограничениями (по функциональности или времени работы) для некоторой группы лиц с тем, чтобы убедиться, что продукт содержит достаточно мало ошибок

28. Системное тестирование предполагает ...

- а) поиск каких-либо проблем в интерфейсах и взаимодействии между компонентами программы
- б) тестирование интегрированной системы на ее соответствие исходным требованиям
- в) имитацию реальной работы с системой штатными разработчиками либо реальной работы с системой потенциальными пользователями/заказчиком на стороне разработчика
- г) распространение версии с ограничениями (по функциональности или времени работы) для некоторой группы лиц с тем, чтобы убедиться, что продукт содержит достаточно мало ошибок

29. Альфа-тестирование предполагает ...

- а) поиск каких-либо проблем в интерфейсах и взаимодействии между компонентами программы
- б) тестирование интегрированной системы на ее соответствие исходным требованиям
- в) имитацию реальной работы с системой штатными разработчиками либо реальной работы с системой потенциальными пользователями/заказчиком на стороне разработчика
- г) распространение версии с ограничениями (по функциональности или времени работы) для некоторой группы лиц с тем, чтобы убедиться, что продукт содержит достаточно мало ошибок

30. Бета-тестирование предполагает ...

- а) поиск каких-либо проблем в интерфейсах и взаимодействии между компонентами программы

- б) тестирование интегрированной системы на ее соответствие исходным требованиям
 - в) имитацию реальной работы с системой штатными разработчиками либо реальной работы с системой потенциальными пользователями/заказчиком на стороне разработчика
 - г) распространение версии с ограничениями (по функциональности или времени работы) для некоторой группы лиц с тем, чтобы убедиться, что продукт содержит достаточно мало ошибок
31. Каждая команда программы должна быть выполнена хотя бы один раз в соответствии с методом тестирования ...
- а) покрытие операторов;
 - б) покрытие решений;
 - в) покрытие условий;
 - г) покрытие решений/условий;
32. Каждое направление перехода в программе должно быть реализовано, по крайней мере, один раз в соответствии с методом тестирования ...
- а) покрытие операторов;
 - б) покрытие решений;
 - в) покрытие условий;
 - г) покрытие решений/условий
33. Все возможные результаты каждого условия в решении должны быть выполнены, по крайней мере, один раз в соответствии с методом тестирования ...
- а) покрытие операторов
 - б) покрытие решений
 - в) покрытие условий
 - г) покрытие решений/условий
34. Видимость переменных и методов класса только в пределах данного модуля определяет спецификатор доступа ...
- а) private
 - б) protected
 - в) public
 - г) published
35. Видимость переменных и методов класса в классах потомков определяет спецификатор доступа ...
- а) private
 - б) protected
 - в) public
 - г) published
36. Видимость переменных и методов класса в любом месте программы определяет спецификатор доступа ...
- а) private
 - б) protected
 - в) public
 - г) published

37. Процесс, в ходе которого один объект может приобретать свойства другого, называется ...
- а) наследование
 - б) инкапсуляция
 - в) полиморфизм
 - г) динамический полиморфизм
38. Процесс перегрузки функций и операторов реализует принцип ...
- а) наследования
 - б) инкапсуляции
 - в) статического полиморфизма
 - г) динамического полиморфизма
39. На основе наследования и виртуальных функций реализуется принцип ...
- а) наследования
 - б) инкапсуляции
 - в) статического полиморфизма
 - г) динамического полиморфизм

Основная литература:

1. Колошкина, И. Е. Инженерная графика. САД: учебник и практикум для среднего профессионального образования / И. Е. Колошкина, В. А. Селезнев. - Москва: Издательство Юрайт, 2021. - 220 с. - (Профессиональное образование). - ISBN 978-5-534-12484-2. - Текст: электронный // Образовательная платформа Юрайт [сайт]. - URL: <https://urait.ru/bcode/475443> (дата обращения: 26.11.2021).
2. Лаврищева, Е. М. Программная инженерия. Парадигмы, технологии и CASE-средства: учебник для вузов / Е. М. Лаврищева. - 2-е изд., испр. - Москва: Издательство Юрайт, 2021. - 280 с. - (Высшее образование). - ISBN 978-5-534-01056-5. - Текст: электронный // Образовательная платформа Юрайт [сайт]. - URL: <https://urait.ru/bcode/470942> (дата обращения: 26.11.2021).
3. Гниденко, И. Г. Технология разработки программного обеспечения: учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. - Москва: Издательство Юрайт, 2021. - 235 с. - (Профессиональное образование). - ISBN 978-5-534-05047-9. - Текст: электронный // Образовательная платформа Юрайт [сайт]. - URL: <https://urait.ru/bcode/472502> (дата обращения: 26.11.2021).
4. Трофимов В. В. Информатика. Том 1: учебник для СПО - 3-е изд., перераб. и доп. - М.: Издательство Юрайт, 2020. - 553 с. [Электронный ресурс] // [Электронно-библиотечная система] URL.: <https://urait.ru/viewer/informatika-v-2-t-tom-1-448997#page/1>

КРИТЕРИИ ОЦЕНКИ

по дисциплине

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 твёрдое 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 объёме, 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 оттестированности проекта: метрики и методика оценки.
24. Системное тестирование
25. Регрессионное тестирование: особенности и виды регрессионного тестирования, методы отбора тестов, оценка эффективности.
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 содержит 20 вопросов случайным образом выбранных их списка. Тест проводится на персональном компьютере в оболочке для тестирования MyTest. 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 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 по специальности 10.02.05.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»

1. Какие существуют особенности управляющего графа программы в случае объектно-ориентированного программирования (ООП)?
 - а) она становится неприменимой

- б) она требует адаптации по обработке сообщений
 - в) она требует описания поведения программы
 - г) она требует описания не только структуры, но и поведения программы
2. Какие существуют особенности интеграционного тестирования для ООП?
- а) тестирование дерева классов
 - б) тестирование последовательностей прямых вызовов методов с помощью Р-путей
 - в) тестирование последовательностей обработки сообщений с помощью ММ-путей
 - г) тестирование исключительно цепочек вызовов процедур программ
3. Какие этапы включает методика ООП при тестировании программного комплекса?
- а) тестирование методов каждого класса программного комплекса
 - б) тестирование отношений между классами с помощью тестов на основе Р-путей или ММ-путей
 - в) тестирование взаимодействия модулей по всей иерархии комплекса
4. Какие этапы методики тестирования используются в ООП?
- а) тестирование методов каждого класса программы
 - б) тестирование методов класса, входящих в его контекст
 - в) тестирование дерева классов программного проекта, включающего оттестированный класс
5. Какие этапы методов тестирования класса используются в ООП?
- а) тестирование класса как модуля по выбранному критерию
 - б) тестирование класса как иерархической структуры
 - в) тестирование классов, входящих в модель проекта
6. Что такое управляющий граф программы (УГП)?
- а) множество операторов программы
 - б) граф, вершины которого кодируют операторы программы, а дуги - управления (порядок исполнения) операторов
 - в) множество операторов управления
7. Что такое путь в УГП?
- а) последовательность вершин и дуг управляющего графа программы с фиксированными начальной и конечной вершиной
 - б) последовательность ветвей управляющего графа программы с фиксированными начальной вершиной первой ветви и конечной вершиной последней ветви пути
 - в) множество связанных дуг управляющего графа программы
8. Что такое ветвь управляющего графа программы?
- а) последовательность вершин и дуг управляющего графа программы с фиксированными начальной и конечной вершиной, которые кодируют либо условные операторы, либо первый и последний операторы управляющего графа программы соответственно

- б) часть пути, в котором все внутренние вершины кодируют линейные операторы
 - в) начальная и конечная вершина пути
9. Можно ли гарантировать остановку программы на любом тесте?
- а) в общем случае нет
 - б) возможно в частных случаях
 - в) задача в общей постановке алгоритмически неразрешима
10. Какие задачи у модульного тестирования?
- а) выявление локальных ошибок реализации алгоритмов модулей
 - б) выявление ошибок при вызове модулей
 - в) выявление ошибок взаимодействия модуля с окружением
11. На основе каких принципов строятся тесты для модульного тестирования?
- а) анализ потоков управления модуля
 - б) анализ потоков данных модуля
 - в) анализ покрытия в соответствии с заданными критериями C0, C1, C2
12. Каковы фазы процесса построения тестовых путей?
- а) построение управляющего графа программы
 - б) выбор тестовых путей
 - в) генерация тестов, соответствующих выбранным тестовым путям
13. Какие существуют методы построения тестовых путей?
- а) статические
 - б) динамические
 - в) методы реализуемых путей
14. Как реализуются динамические методы построения тестовых путей?
- а) наращивание начальных отрезков реализованных путей продолжающими их фрагментами, чтобы увеличить покрытие
 - б) построение пути методом удлинения за счет добавления дуг
 - в) поиск всех реализуемых путей
15. Какие существуют разновидности интеграционного тестирования?
- а) монолитное тестирование
 - б) нисходящее тестирование
 - в) восходящее тестирование
 - г) Регрессионное тестирование
16. Каковы особенности нисходящего тестирования?
- а) необходимость разработки заглушек
 - б) необходимость разработки среды управления очередностью вызовов модулей
 - в) параллельная разработка эффективных модулей
17. Каковы особенности восходящего тестирования?
- а) минимизация разработки заглушек
 - б) запаздывание в проверке функциональности реализуемого приложения
 - в) необходимость разработки среды управления очередностью вызовов модулей

18. В чем заключаются особенности интеграционного тестирования для процедурного программирования?
- а) тестирование программных комплексов, заданных в виде иерархических структур модулей
 - б) использование диаграмм потока управления в качестве модели тестируемого комплекса
 - в) контроль соответствия спецификациям параметров модулей и межмодульных связей
 - г) контроль наследования
19. При создании очередной версии программы была добавлена функция А, функция D была удалена, функция С – изменена. К какой группе относится тест, покрывающий только функцию А?
- а) тесты, пригодные для повторного использования
 - б) тесты, требующие повторного запуска
 - в) устаревшие тесты
 - г) новые тесты
20. Возможно ли тестирование программы на всех допустимых значениях параметров?
- а) никогда
 - б) да, всегда
 - в) возможно в отдельных случаях
21. Каковы особенности системного тестирования?
- а) тестированию подлежит система в целом
 - б) тесты оперируют пользовательским или другими внешними интерфейсами
 - в) структура проекта тестируется на уровне подсистем
 - г) тестирование осуществляется по методу «черного ящика»
22. Какие задачи решаются на уровне системного тестирования?
- а) выявление дефектов в функционировании приложения или в работе с ним
 - б) выявление дефектов использования ресурсов
 - в) выявление несовместимости с окружением
 - г) выявление непредусмотренных сценариев применения
 - д) использования непредусмотренных комбинаций данных
23. Какие категории тестов разрабатываются для системного тестирования?
- а) тесты для проверки полноты функциональности
 - б) тесты для проверки корректности использования ресурсов
 - в) тесты для проверки стрессовых режимов и оценки производительности
 - г) тесты для проверки защиты от искаженных данных и некорректных действий
 - д) тесты для проверки инсталляции и конфигурации для платформ, предусмотренных спецификацией
 - е) тесты для проверки корректности пользовательской документации

24. Каковы особенности регрессионного тестирования?
- а) регрессионное тестирование является подмножеством системного тестирования
 - б) выбор между полным и частичным перетестированием и пополнением тестовых наборов
 - в) перетестирование предусматривает только контроль частей приложения, связанных с изменениями
25. Какие задачи решает тестировщик, проверяя изменения, внесенные разработчиком в код?
- а) проверка и подтверждение исправления дефекта
 - б) обеспечение гарантий, что качество исправленного кода не ухудшилось
 - в) перетестирование последствий исправлений на предмет обнаружения индуцированных ошибок
 - г) проверка возможности воспроизведения ошибки каким-либо другим способом
26. Какие типы дефектов выявляются при системном или при регрессионном тестировании?
- а) отсутствующая или некорректная функциональность
 - б) некорректность проектной документации
 - в) непредусмотренные данные или неподдерживаемые сценарии использования
 - г) ошибки переносимости на другие платформы
 - д) ошибки пользовательской документации
 - е) ошибки инсталляции и конфигурирования
27. Можно ли гарантировать безопасность метода регрессионного тестирования при отсутствии информации об изменениях в программе
- а) нет
 - б) да
28. Какие методы регрессионного тестирования применяются в условиях высоких требований к качеству программного продукта?
- а) метод повторного прогона всех тестов
 - б) случайные методы
 - в) безопасные методы
 - г) методы минимизации
 - д) методы, основанные на покрытии кода
29. Какими свойствами обладает метод повторного прогона всех тестов?
- а) полнота
 - б) точность
 - в) эффективность
 - г) универсальность
30. Можно ли гарантировать безопасность метода регрессионного тестирования в условиях отсутствия тестов, использовавшихся при тестировании предыдущих версий?
- а) тесты, пригодные для повторного использования

- б) тесты, требующие повторного запуска
- в) устаревшие тесты
- г) новые тесты

31. Какие из перечисленных методов тестирования наиболее затратны

- а) системное тестирование с моделируемым окружением
- б) статические методы тестирования
- в) интеграционное тестирование
- г) системное тестирование в реальном окружении и реальном времени
- д) модульное тестирование

32. Какие из перечисленных методов тестирования дают наиболее надежные результаты?

- а) статические методы
- б) модульное тестирование
- в) интеграционное тестирование
- г) системное тестирование с моделируемым окружением
- д) системное тестирование в реальном окружении и реальном времени

33. Что влияет на обеспечение качества программного продукта?

- а) тестирование
- б) анализ дизайна
- в) обзоры кода
- г) аудиты процесса разработки ревьюирование проектной документации
- д) ревьюирование проектной документации

34. Какие основные уровни выделены в процессе тестирования?

- а) модульное тестирование
- б) Регрессионное тестирование
- в) интеграционное тестирование
- г) системное тестирование

35. Какие методы регрессионного тестирования применяются в условиях высоких требований к программному продукту?

- а) метод повторного прогона всех тестов
- б) случайные методы
- в) безопасные методы
- г) методы, основанные на покрытии кода
- д) методы минимизации

36. В среднем, метод выборочного регрессионного тестирования...

- а) обнаруживает столько же ошибок, сколько и метод повторного прогона всех тестов
- б) обнаруживает больше ошибок, чем метод повторного прогона всех тестов
- в) обнаруживает меньше ошибок, чем метод повторного прогона всех тестов

- г) может обнаруживать больше или меньше ошибок, чем метод повторного прогона всех тестов
37. На каком этапе регрессионного тестирования удаляются устаревшие тесты?
- а) предсказание целесообразности
 - б) идентификация изменений
 - в) отбор тестов
 - г) выполнение тестов
 - д) создание дополнительных тестов
 - е) обновление базы данных
38. На каком этапе регрессионного тестирования проводится упорядочение тестов?
- а) предсказание целесообразности
 - б) идентификация изменений
 - в) отбор тестов
 - г) выполнение тестов
 - д) создание дополнительных тестов
 - е) обновление базы данных

Основная литература:

5. Колошкина, И. Е. Инженерная графика. САД: учебник и практикум для среднего профессионального образования / И. Е. Колошкина, В. А. Селезнев. - Москва: Издательство Юрайт, 2021. - 220 с. - (Профессиональное образование). - ISBN 978-5-534-12484-2. - Текст: электронный // Образовательная платформа Юрайт [сайт]. - URL: <https://urait.ru/bcode/475443> (дата обращения: 26.11.2021).

6. Лаврищева, Е. М. Программная инженерия. Парадигмы, технологии и CASE-средства: учебник для вузов / Е. М. Лаврищева. - 2-е изд., испр. - Москва: Издательство Юрайт, 2021. - 280 с. - (Высшее образование). - ISBN 978-5-534-01056-5. - Текст: электронный // Образовательная платформа Юрайт [сайт]. - URL: <https://urait.ru/bcode/470942> (дата обращения: 26.11.2021).

7. Гниденко, И. Г. Технология разработки программного обеспечения: учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. - Москва: Издательство Юрайт, 2021. - 235 с. - (Профессиональное образование). - ISBN 978-5-534-05047-9. - Текст: электронный // Образовательная платформа

Юрайт [сайт]. - URL: <https://urait.ru/bcode/472502> (дата обращения: 26.11.2021).

8. Трофимов В. В. Информатика. Том 1: учебник для СПО - 3-е изд., перераб. и доп. - М.: Издательство Юрайт, 2020. - 553 с. [Электронный ресурс] // [Электронно-библиотечная система] URL.: <https://urait.ru/viewer/informatika-v-2-t-tom-1-448997#page/1>

Дополнительная литература:

1. Демин А. Ю., Дорофеев В. А. Информатика. Лабораторный практикум: учебное пособие для СПО.- М.: Издательство Юрайт, 2020. - 133 с. [Электронный ресурс] // [Электронно-библиотечная система] URL.: <https://urait.ru/viewer/informatika-laboratornyy-praktikum-448945#page/1>

2. Кедрова Г. Е. Информатика для гуманитариев: учебник и практикум для СПО. – М.: Издательство Юрайт, 2020. - 439 с. [Электронный ресурс] // [Электронно-библиотечная система] URL.: <https://urait.ru/viewer/informatika-dlya-gumanitariyv-456496#page/2>

3. Зимин В. П. Информатика. Лабораторный практикум в 2 ч. Часть 1: учебное пособие для СПО - 2-е изд., испр. и доп. – М.: Издательство Юрайт, 2020. - 126 с. [Электронный ресурс] // [Электронно-библиотечная система] URL.: <https://urait.ru/viewer/informatika-laboratornyy-praktikum-v-2-ch-chast-1-453928#page/1>

КРИТЕРИИ ОЦЕНКИ

по дисциплине

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 неструктурирован 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 неструктурирован, 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 неструктурирован, 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.

Использование изображений.

10. Акселерометр и служба определения местоположения, вторичные потоки выполнения, обработка асинхронных операций и доступ к Веб-сервисам.
11. 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 кадрово́й 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/AndroidStudio/WebView/ Phonegap и др.)
5. Инструментарий среды разработки мобильных приложений. Структура типичного мобильного приложения.
6. Архитектура платформы Android. Dalvik Virtual Machine. Среда разработки для Android. Eclipse IDE. II
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. XMLразметка 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 SharedPreferences. 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

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 (Object 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.

2. Создайте простейшей веб-странице, отображающей на экране мобильного устройства сообщение "Hello world!".

3. 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.

4. Создать сервер обмена текстовыми сообщениями.

Создать сервер, который 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.

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

6. 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 видоизмененные 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.;

4. 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 материалом (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 успеют 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 ответов.

КОМ предназначен для контроля и оценки результатов освоения учебной дисциплины «Разработка мобильных приложений» по специальности СПО 09.02.07 «Информационные системы и программирование».

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

31 – основные этапы разработки мобильных приложений

32 – 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 20 вопросов случайным образом выбранных из списка. Тест проводится на персональном компьютере в оболочке для тестирования MyTest. 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:**
 - a) 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 загрузженность 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 загрузженность 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 широковещательных сообщений является наследником класса ...

- а) ViewReceiver
- 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/value
- 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. Какой язык разметки используется для описания иерархии компонентов графического пользовательского интерфейса Android-приложения?

- а) html
- б) xml
- в) gml
- г) xhtml

16. 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

17. 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

18. 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

19. Какие элементы управления применяются для действий по настройке?

- а) командные элементы управления
- б) элементы выбора
- в) элементы ввода
- г) элементы отображения

20. Примерами комбо-элементов не являются:

- а) комбо-список
- б) все вышеперечисленное
- в) комбо-кнопка
- г) комбо-поле

21. 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

22. Более крупные элементы:

- а) привлекают больше внимания
- б) все варианты ответа верны
- в) размер не влияет на уровень внимания
- г) привлекают меньше внимания

23. К традиционным типографическим инструментам не относят

- а) масштаб
- б) цвет
- в) разреженность
- г) выравнивание по сетке

24. К элементам ввода относят:

- а) ограничивающие элементы ввода
- б) ползунки
- в) счетчики
- г) все вышеперечисленное

25. 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

26. 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

27. 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

28. Основные вкладки (FixedTabs) удобны при отображении

- а) от четырех вкладок
- б) двух вкладок
- в) трех и более вкладок
- г) трех и менее вкладок

29.Диалоговое окно, содержащее линейку процесса выполнения какого-то действия — это

- а) DatePickerDialog
- б) AlertDialog
- в) ProgressDialog
- г) DialogFragment

30.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

31.Какой метод запускает новую активность?

- а) startActivity()
- б) beginActivity()
- в) intentActivity()
- г) newActivity()

Основная литература:

9. *Колошкина, И. Е.* Инженерная графика. CAD: учебник и практикум для среднего профессионального образования / И. Е. Колошкина, В. А. Селезнев. - Москва: Издательство Юрайт, 2021. - 220 с. - (Профессиональное образование). - ISBN 978-5-534-12484-2. - Текст: электронный // Образовательная платформа Юрайт [сайт]. - URL: <https://urait.ru/bcode/475443> (дата обращения: 26.11.2021).

10. *Лаврищева, Е. М.* Программная инженерия. Парадигмы, технологии и CASE-средства: учебник для вузов / Е. М. Лаврищева. - 2-е изд., испр. - Москва: Издательство Юрайт, 2021. - 280 с. - (Высшее образование). - ISBN 978-5-534-01056-5. - Текст: электронный //

Образовательная платформа Юрайт [сайт]. -
URL: <https://urait.ru/bcode/470942> (дата обращения: 26.11.2021).

11. Гниденко, И. Г. Технология разработки программного обеспечения: учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. - Москва: Издательство Юрайт, 2021. - 235 с. - (Профессиональное образование). - ISBN 978-5-534-05047-9. - Текст: электронный // Образовательная платформа Юрайт [сайт]. - URL: <https://urait.ru/bcode/472502> (дата обращения: 26.11.2021).

12. Трофимов В. В. Информатика. Том 1: учебник для СПО - 3-е изд., перераб. и доп. - М.: Издательство Юрайт, 2020. - 553 с. [Электронный ресурс] // [Электронно-библиотечная система] URL.: <https://urait.ru/viewer/informatika-v-2-t-tom-1-448997#page/1>