Приложение
к ППССЗ специальности 09.02.11 Разработка и управление программными модулями

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ ДАГЕСТАН

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
РЕСПУБЛИКИ ДАГЕСТАН «ТЕХНИЧЕСКИЙ КОЛЛЕДЖ ИМЕНИ Р.Н. АШУРАЛИЕВА»

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

междисциплинарного курса
МДК.02.07 Разработка приложений для мобильных платформ

Специальность: 09.02.11 Разработка и управление программными модулями

Квалификация выпускника: программист

Махачкала, 2026 г.

ОДОБРЕНО

предметной (цикловой) комиссией УГС 09.00.00. Информатика и вычислительная техника и 10.00.00 Информационная безопасность

Председатель П(Ц)К

 Ш.М. Мусаева

Протокол № 8 от 30 апреля 2026 г.

Рабочая программа междисциплинарного курса МДК.02.07 Разработка приложений для мобильных платформ разработана на основе:

- Приказа Минпросвещения России от 24.02.2025 N 138 "Об утверждении федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.11 Разработка и управление программным обеспечением" (Зарегистрировано в Минюсте России 31.03.2025 N 81696);

с учетом:

- Примерной образовательной программы «Профессионалитет» по специальности 09.02.11 Разработка и управление программным обеспечением, утвержденная протоколом ФУМО в системе СПО по УГПС 09.00.00 Информатика и вычислительная техника от 29.08.2025 № 6/2025, зарегистрирована в государственном реестре примерных образовательных программ – регистрационный номер 18/2025. Реквизиты утверждающего документа –приказ ФГБОУ ДПО ИРПО № 01-09-581/2025 от 13.10.2025;

в соответствии с учебным планом.

Разработчик:

- Магомедова Фаида Ибрагимовна, преподаватель ГБПОУ РД «Технический колледж имени Р.Н. Ашуралиева»

© Магомедова Фаида Ибрагимовна 2026

© ГБПОУ РД «Технический колледж Р.Н. Ашуралиева» 2026

СОДЕРЖАНИЕ

1. ОБЩИЕ ПОЛОЖЕНИЯ....4
2. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ МЕЖДИСЦИПЛИНАРНОГО КУРСА.4
3. ФОРМЫ И МЕТОДЫ КОНТРОЛЯ И ОЦЕНИВАНИЯ....11
4. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ МЕЖДИСЦИПЛИНАРНОГО КУРСА.14
5. ОЦЕНОЧНЫЕ СРЕДСТВА ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ....18
5.1. Типовые задания для оценки знаний и умений....18
5.2. Критерии оценивания....79
5.2.1. Критерии оценивания устного ответа....79
5.2.2. Критерии оценивания выполнения заданий на лабораторных и практических занятиях....79
5.2.3. Критерии оценивания тестовых заданий....79
5.2.4. Общая классификация ошибок....79
6. ОЦЕНОЧНЫЕ СРЕДСТВА ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ....80
ОБУЧАЮЩИХСЯ ПО МЕЖДИСЦИПЛИНАРНОМУ КУРСУ....80
6.1. Вопросы к экзамену по МДК.02.07 за I семестр....80
6.2. Вопросы к экзамену по МДК.02.07 за II семестр....81
6.3. Критерии оценивания ответов на экзамене....82
7. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ....83

1. ОБЩИЕ ПОЛОЖЕНИЯ

Фонд оценочных средств (далее – ФОС) дисциплины междисциплинарного курса МДК.02.07 Разработка приложений для мобильных платформ является частью программы подготовки специалистов среднего звена специальности 09.02.11 Разработка и управление программным обеспечением.

ФОС позволяет оценить достижение, запланированных по междисциплинарному курсу, результатов обучения.

ФОС включают оценочные средства для проведения текущего контроля успеваемости и промежуточной аттестации по междисциплинарному курсу.

Текущий контроль успеваемости осуществляется с целью регулярного наблюдения за ходом поэтапного освоения обучающимися междисциплинарного курса, оптимизации управления образовательной деятельностью обучающихся, своевременной корректировки персональных образовательных результатов обучающихся педагогическими средствами.

Текущему контролю успеваемости подлежат все обучающиеся, осваивающие междисциплинарный курс.

Текущий контроль проводится в пределах учебного времени, отведенного на изучение междисциплинарного курса традиционными и инновационными методами с использованием современных технологий.

Результаты текущего контроля успеваемости обучающихся в виде оценки в балльном выражении («5», «4», «3», «2») записываются в журнале учебных занятий.

Текущий контроль освоения обучающимися программного материала междисциплинарного курса может иметь следующие виды: оперативный и рубежный контроль.

Оперативный контроль проводится с целью объективной оценки качества освоения программы междисциплинарного курса, а также стимулирования учебной работы обучающихся, мониторинга результатов образовательной деятельности, подготовки к промежуточной аттестации и обеспечения максимальной эффективности учебно-воспитательного процесса.

Рубежный контроль является контрольной точкой и проводится с целью комплексной оценки уровня освоения программного материала.

Промежуточная аттестация по междисциплинарному курсу проводится с целью оценки уровня освоения теоретических знаний, умений, приобретенного практического опыта.

Формы и периодичность промежуточной аттестации по междисциплинарному курсу определяются учебным планом образовательной программы: экзамен в 5 и 6 семестрах.

Экзамен проводится непосредственно после завершения освоения междисциплинарного курса, в сроки, установленные календарным учебным графиком. Экзамен проводится в день, освобожденный от других форм учебной нагрузки.

Экзаменационные вопросы и задания составляются на основе рабочей программы междисциплинарного курса. Экзаменационные вопросы и задания должны соответствовать проверяемым результатам обучения и доводятся до сведения обучающихся в течение первых двух месяцев от начала обучения.

2. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ МЕЖДИСЦИПЛИНАРНОГО КУРСА

ПМ.02 Разработка и интеграция модулей программного обеспечения направлен на формирование общих и профессиональных компетенций.

Освоение междисциплинарного курса должно способствовать формированию общих компетенций:

– ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

– ОК 02. Использовать современные средства поиска, анализа и интерпретации информации,

и информационные технологии для выполнения задач профессиональной деятельности;

– ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

– ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках.

В результате освоения междисциплинарного курса обучающийся должен овладеть профессиональными компетенциями:

– ПК 2.6. Разрабатывать модули программного обеспечения для мобильных платформ.

– ПК 2.7. Проектировать и разрабатывать пользовательский интерфейс и пользовательский опыт.

– ПК 2.8. Проектировать и разрабатывать базы данных для мобильных платформ.

– ПК 2.9. Осуществлять внедрение мультимедиа в программное обеспечение для мобильных платформ.

В результате освоения междисциплинарного курса обучающийся должен владеть **навыками:**

- разработки модулей программного обеспечения для мобильных платформ;
- разработки многопоточных приложений;
- оптимизации производительности приложений;
- работы с интеграцией сторонних библиотек;
- создания пользовательских интерфейсов с использованием инструментов и библиотек, таких как UIKit (iOS) и Android XML (Android);
- разработки адаптивных и мультирезолюционных интерфейсов;
- тестирования пользовательского опыта;
- работы с SQLite и другими СУБД для мобильных платформ;
- разработки эффективных схем баз данных;
- разработки функций и возможностей для работы с базами данных в программном обеспечении для мобильных платформ;
- интеграции баз данных в пользовательский интерфейс приложений для удобного доступа и управления данными;
- создания и редактирования графических элементов для приложений с использованием специализированных инструментов;
- интеграции изображений и иконок в пользовательский интерфейс;
- разработки и анимации пользовательских элементов и переходов;
- разработки мультимедийных функций и возможностей в программном обеспечении для мобильных платформ;
- создания интерфейсов для работы с изображениями, видео и аудио в приложениях для мобильных устройств;
- оптимизации работы с мультимедиа для обеспечения высокой производительности и эффективного использования ресурсов устройства;
- получения медиа-данных с помощью механизмов в операционной системе.

В результате освоения междисциплинарного курса обучающийся должен **уметь**:

Код умения	**Название умения**
У1	разрабатывать программный код;
У2	отлаживать приложения на различных устройствах;
У3	работать с системами контроля версий;
У4	использовать паттерны проектирования;

У5	осуществлять тестирование кода;
У6	производить рефакторинг;
У7	интегрировать приложения с облачными сервисами;
У8	создавать интуитивно понятные и легко навигируемые интерфейсы;
У9	использовать анимацию и переходы для улучшения пользовательского опыта;
У10	оптимизировать интерфейс для работы на разных экранах и устройствах;
У11	интегрировать элементы пользовательского интерфейса с серверной частью или базой данных приложения;
У12	анализировать пользовательские данные и обратную связь для улучшения UX;
У13	разрабатывать макеты и прототипы приложений;
У14	владеть инструментами дизайна интерфейса;
У15	глубоко понимать принципы дизайна пользовательского интерфейса и пользовательского опыта;
У16	проектировать и оптимизировать базы данных;
У17	выполнять CRUD (Create, Read, Update, Delete) операции;
У18	обеспечивать синхронизацию данных между устройствами;
У19	работать с кэшированием данных;
У20	обрабатывать конфликты данных в распределенных системах;
У21	работать с многозадачностью и потоками данных;
У22	глубоко понимать принципы работы с базами данных в программном обеспечении для мобильных платформ;
У23	создавать и оптимизировать структуру баз данных для хранения и обработки данных в мобильных приложениях;
У24	работать с ORM (Object-Relational Mapping) инструментами для более удобного взаимодействия с базами данных;
У25	обеспечивать безопасность и защиту данных при работе с базами данных в мобильных приложениях;
У26	работать с разными форматами изображений и аудиофайлами;
У27	создавать графические ресурсы с высоким разрешением;
У28	проектировать интерфейс с учетом визуальных аспектов, таких как цвета, шрифты и стили;
У29	осуществлять анимацию интерфейсных элементов;
У30	обрабатывать и интегрировать аудио в приложение для воспроизведения звуков и музыки;
У31	владеть инструментами для работы с мультимедиа;
У32	понимать принципы работы с изображениями, видео и аудио в программном обеспечении для мобильных платформ;
У33	работать с анимацией и эффектами для создания привлекательных визуальных элементов в приложениях для мобильных устройств;
У34	оптимизировать мультимедийные элементы для обеспечения быстрой загрузки и плавной работы на мобильных устройствах.

В результате освоения дисциплины обучающийся должен **знать:**

Код знания	**Название знания**
З1	основы языков программирования;
З2	принципы ООП и функционального программирования;
З3	архитектуры мобильных приложений (MVC, MVVM, VIPER);
З4	принципы работы основных мобильных ОС (iOS, Android);
З5	жизненный цикл мобильного приложения;
З6	методы оптимизации производительности;
З7	основы работы с графическим интерфейсом и анимацией;
З8	основы безопасности в мобильной разработке;
З9	основы работы с сетью и API;
З10	принципы работы с базами данных на мобильных платформах;

311	платформы по кроссплатформенной разработке, таких как Flutter, React Native или MAUI;
312	принципы дизайна пользовательского интерфейса (UI) и пользовательского опыта (UX);
313	основы графического дизайна и типографики;
314	гайдлайны и стандарты для создания интерфейсов на платформах iOS и Android;
315	принципы адаптивного дизайна;
316	основы работы с векторной и растровой графикой;
317	процесс проектирования интерфейса от идеи до реализации;
318	основы реляционных баз данных;
319	основные принципы работы с базами данных в программном обеспечении для мобильных платформ;
320	современные тенденции в разработке мобильных приложений с использованием баз данных;
321	основные принципы проектирования баз данных для эффективного хранения и обработки данных в мобильных приложениях;
322	основные технологии разработки мобильных приложений, таких как Java, Kotlin, Swift или React Native, для работы с базами данных;
323	основы графического дизайна и композиции;
324	основы аудиодизайна и звуковой обработки;
325	принципы анимации и визуальной привлекательности в мобильных приложениях;
326	основные принципы работы с изображениями, видео и аудио в программном обеспечении для мобильных платформ;
327	современные тенденции в дизайне и использовании мультимедиа в приложениях для мобильных устройств;
328	основные принципы разработки мультимедийных функций с учетом ограниченных ресурсов мобильных устройств;
329	основные технологии разработки мобильных приложений, таких как Java, Kotlin, Swift или React Native.

Общие компетенции:

Код компетенции	Формулировка компетенции	Знания, умения
ОК 01	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам	**Умения:** распознавать задачу и/или проблему в профессиональном и/или социальном контексте, анализировать и выделять её составные части определять этапы решения задачи, составлять план действия, реализовывать составленный план, определять необходимые ресурсы выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы владеть актуальными методами работы в профессиональной и смежных сферах оценивать результат и последствия своих действий (самостоятельно или с помощью наставника) **Знания:** актуальный профессиональный и социальный контекст, в котором приходится работать и жить структура плана для решения задач, алгоритмы выполнения работ в профессиональной и смежных областях основные источники информации и ресурсы для решения задач и/или проблем в профессиональном и/или социальном контексте методы работы в профессиональной и смежных сферах

		порядок оценки результатов решения задач профессиональной деятельности
ОК 02	Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности	**Умения:** определять задачи для поиска информации, планировать процесс поиска, выбирать необходимые источники информации выделять наиболее значимое в перечне информации, структурировать получаемую информацию, оформлять результаты поиска оценивать практическую значимость результатов поиска применять средства информационных технологий для решения профессиональных задач использовать современное программное обеспечение в профессиональной деятельности использовать различные цифровые средства для решения профессиональных задач **Знания:** номенклатура информационных источников, применяемых в профессиональной деятельности приемы структурирования информации формат оформления результатов поиска информации современные средства и устройства информатизации, порядок их применения и программное обеспечение в профессиональной деятельности, в том числе цифровые средства
ОК 05	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста	**Умения:** грамотно излагать свои мысли и оформлять документы по профессиональной тематике на государственном языке проявлять толерантность в рабочем коллективе **Знания:** правила оформления документов правила построения устных сообщений особенности социального и культурного контекста
ОК 09	Пользоваться профессиональной документацией на государственном и иностранном языках	**Умения:** понимать общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые), понимать тексты на базовые профессиональные темы участвовать в диалогах на знакомые общие и профессиональные темы строить простые высказывания о себе и о своей профессиональной деятельности кратко обосновывать и объяснять свои действия (текущие и планируемые) писать простые связные сообщения на знакомые или интересующие профессиональные темы **Знания:** правила построения простых и сложных предложений на профессиональные темы основные общеупотребительные глаголы (бытовая и профессиональная лексика) лексический минимум, относящийся к описанию предметов, средств и процессов профессиональной деятельности особенности произношения правила чтения текстов профессиональной направленности

Профессиональные компетенции:

Код и формулировка компетенции	Показатели освоения компетенции
ПК 2.6. Разрабатывать модули программного обеспечения для мобильных платформ.	**Навыки:** разработки модулей программного обеспечения для мобильных платформ; разработки многопоточных приложений; оптимизации производительности приложений; работы с интеграцией сторонних библиотек.
	Умения: разрабатывать программный код; отлаживать приложения на различных устройствах; работать с системами контроля версий; использовать паттерны проектирования; осуществлять тестирование кода; производить рефакторинг; интегрировать приложения с облачными сервисами
	Знания: основы языков программирования; принципы ООП и функционального программирования; архитектуры мобильных приложений (MVC, MVVM, VIPER); принципы работы основных мобильных ОС (iOS, Android); жизненный цикл мобильного приложения; методы оптимизации производительности; основы работы с графическим интерфейсом и анимацией. основы безопасности в мобильной разработке; основы работы с сетью и API; принципы работы с базами данных на мобильных платформах; платформы по кроссплатформенной разработке, таких как Flutter, React Native или MAUI.
ПК 2.7. Проектировать и разрабатывать пользовательский интерфейс и пользовательский опыт.	**Навыки:** создания пользовательских интерфейсов с использованием инструментов и библиотек, таких как UIKit (iOS) и Android XML (Android); разработки адаптивных и мультирезолюционных интерфейсов; тестирования пользовательского опыта.
	Умения: создавать интуитивно понятные и легко навигируемые интерфейсы; использовать анимацию и переходы для улучшения пользовательского опыта; оптимизировать интерфейс для работы на разных экранах и устройствах; интегрировать элементы пользовательского интерфейса с серверной частью или базой данных приложения; анализировать пользовательские данные и обратную связь для улучшения UX; разрабатывать макеты и прототипы приложений; владеть инструментами дизайна интерфейса; глубоко понимать принципы дизайна пользовательского интерфейса и пользовательского опыта.
	Знания: принципы дизайна пользовательского интерфейса (UI) и пользовательского опыта (UX); основы графического дизайна и типографики; гайдлайны и стандарты для создания интерфейсов на платформах iOS и Android; принципы адаптивного дизайна ; основы работы с векторной и растровой графикой; процесс проектирования интерфейса от идеи до реализации.
ПК 2.8. Проектировать и	**Навыки:** работы с SQLite и другими СУБД для мобильных платформ;

разрабатывать базы данных для мобильных платформ.	разработки эффективных схем баз данных; разработки функций и возможностей для работы с базами данных в программном обеспечении для мобильных платформ; интеграции баз данных в пользовательский интерфейс приложений для удобного доступа и управления данными.
	Умения: проектировать и оптимизировать базы данных; выполнять CRUD (Create, Read, Update, Delete) операции; обеспечивать синхронизацию данных между устройствами; работать с кэшированием данных; обрабатывать конфликты данных в распределенных системах; работать с многозадачностью и потоками данных; глубоко понимать принципы работы с базами данных в программном обеспечении для мобильных платформ; создавать и оптимизировать структуру баз данных для хранения и обработки данных в мобильных приложениях; работать с ORM (Object-Relational Mapping) инструментами для более удобного взаимодействия с базами данных; обеспечивать безопасность и защиту данных при работе с базами данных в мобильных приложениях.
	Знания: основы реляционных баз данных; основные принципы работы с базами данных в программном обеспечении для мобильных платформ; современные тенденции в разработке мобильных приложений с использованием баз данных; основные принципы проектирования баз данных для эффективного хранения и обработки данных в мобильных приложениях; основные технологии разработки мобильных приложений, таких как Java, Kotlin, Swift или React Native, для работы с базами данных.
ПК 2.9. Осуществлять внедрение мультимедиа в программное обеспечение для мобильных платформ.	**Навыки:** создания и редактирования графических элементов для приложений с использованием специализированных инструментов; интеграции изображений и иконок в пользовательский интерфейс; разработки и анимации пользовательских элементов и переходов; разработки мультимедийных функций и возможностей в программном обеспечении для мобильных платформ; создания интерфейсов для работы с изображениями, видео и аудио в приложениях для мобильных устройств; оптимизации работы с мультимедиа для обеспечения высокой производительности и эффективного использования ресурсов устройства; получения медиа-данных с помощью механизмов в операционной системе.
	Умения: работать с разными форматами изображений и аудиофайлами; создавать графические ресурсы с высоким разрешением; проектировать интерфейс с учетом визуальных аспектов, таких как цвета, шрифты и стили; осуществлять анимацию интерфейсных элементов; обрабатывать и интегрировать аудио в приложение для воспроизведения звуков и музыки; владеть инструментами для работы с мультимедиа; понимать принципы работы с изображениями, видео и аудио в программном обеспечении для мобильных платформ; работать с анимацией и эффектами для создания привлекательных визуальных элементов в приложениях для мобильных устройств; оптимизировать мультимедийные элементы для обеспечения быстрой загрузки и плавной работы на мобильных устройствах.

	Знания: основы графического дизайна и композиции; основы аудиодизайна и звуковой обработки; принципы анимации и визуальной привлекательности в мобильных приложениях; основные принципы работы с изображениями, видео и аудио в программном обеспечении для мобильных платформ; современные тенденции в дизайне и использовании мультимедиа в приложениях для мобильных устройств; основные принципы разработки мультимедийных функций с учетом ограниченных ресурсов мобильных устройств; основные технологии разработки мобильных приложений, таких как Java, Kotlin, Swift или React Native.

3. ФОРМЫ И МЕТОДЫ КОНТРОЛЯ И ОЦЕНИВАНИЯ

Формы текущего контроля по междисциплинарному курсу:

- устный опрос (фронтальный, индивидуальный, комбинированный);
- тестирование (письменное или компьютерное);
- письменная проверка (ответы на вопросы, решение задач и примеров, составление тезисов, рефератов, выполнение схем, выполнение заданий для самостоятельной работы и др.);
- практическая проверка (при проведении практических и лабораторных занятий);
- самоконтроль и взаимопроверка.

Возможны и другие формы текущего контроля успеваемости, в том числе инновационные на основе информационно-коммуникационных технологий.

Преподаватель на одном учебном занятии может использовать одну или несколько форм текущего контроля.

Код и наименование профессиональных и общих компетенций, формируемых в рамках модуля	**Критерии оценки**	**Методы оценки**
ПК 2.6. Разрабатывать модули программного обеспечения для мобильных платформ	Оценка **«отлично»** - программный модуль разработан по имеющемуся алгоритму в среде разработки на указанном языке программирования методами объектно- ориентированного/ структурного программирования и полностью соответствует техническому заданию, соблюдены и пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам. Оценка **«хорошо»** - программный модуль разработан по имеющемуся алгоритму в среде разработки на указанном языке программирования методами объектно- ориентированного/ структурного программирования и практически соответствует техническому заданию с незначительными отклонениями, пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам. Оценка **«удовлетворительно»** - программный модуль разработан по имеющемуся алгоритму в среде разработки на указанном языке программирования	– Экзамен в форме собеседования: – Практическое задание по разработке программного модуля в соответствии с заданием. – Защита отчетов по лабораторным занятиям – Устный опрос – Тестирование – Интерпретация результатов наблюдений за деятельностью обучающегося

	методами объектно- ориентированного/ структурного программирования и соответствует техническому заданию; документация на модуль оформлена без существенных отклонений от стандартов	
ПК 2.7. Проектировать и разрабатывать пользовательский интерфейс и пользовательский опыт.	Оценка «отлично» - программный модуль разработан по имеющемуся алгоритму в среде **разработки на указанном языке программирования** методами объектно- ориентированного/ структурного программирования и полностью соответствует техническому заданию, соблюдены и пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам. Оценка «хорошо» - программный модуль разработан по имеющемуся алгоритму в среде разработки на указанном языке программирования методами объектно- ориентированного/ структурного программирования и практически соответствует техническому заданию с незначительными отклонениями, пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам. Оценка «удовлетворительно» - программный модуль разработан по имеющемуся алгоритму в среде разработки на указанном языке программирования методами объектно- ориентированного/ структурного программирования и соответствует техническому заданию; документация на модуль оформлена без существенных отклонений от стандартов	– Экзамен в форме собеседования: – Практическое задание по разработке программного модуля в соответствии с заданием. – Защита отчетов по лабораторным занятиям – Устный опрос – Тестирование – Интерпретация результатов наблюдений за деятельностью обучающегося
ПК 2.8. Проектировать и разрабатывать базы данных для мобильных платформ.	Оценка «**отлично**» - разработан модуль для заданного мобильного устройства с соблюдением основных этапов разработки на одном из современных языков программирования; при проверке работоспособности модуля на устройстве или эмуляторе установлено его соответствие спецификации. Оценка «**хорошо**» - разработан модуль для заданного мобильного устройства с учетом основных этапов разработки на одном из современных языков программирования; при проверке работоспособности модуля на устройстве или эмуляторе установлено соответствие выполняемых функций спецификации с незначительными отклонениями. Оценка «**удовлетворительно**» - разработан модуль для заданного мобильного устройства на одном из современных языков программирования; при проверке работоспособности модуля на устройстве или эмуляторе установлено соответствие основных выполняемых функций спецификации.	– Экзамен в форме собеседования: – Практическое задание по разработке программного модуля в соответствии с заданием. – Защита отчетов по лабораторным занятиям – Устный опрос – Тестирование – Интерпретация результатов наблюдений за деятельностью обучающегося
ПК 2.9. Осуществлять внедрение мультимедиа в программное обеспечение для мобильных платформ.	Оценка «**отлично**» - разработан модуль для заданного мобильного устройства с соблюдением основных этапов разработки на одном из современных языков программирования; при проверке работоспособности модуля на устройстве или эмуляторе установлено его соответствие спецификации. Оценка «**хорошо**» - разработан модуль для заданного мобильного устройства с учетом основных этапов разработки на одном из современных языков программирования; при проверке работоспособности модуля на устройстве или эмуляторе установлено	– Экзамен в форме собеседования: – Практическое задание по разработке программного модуля в соответствии с заданием. – Защита отчетов по лабораторным занятиям

	соответствие выполняемых функций спецификации с незначительными отклонениями. Оценка «**удовлетворительно**» - разработан модуль для заданного мобильного устройства на одном из современных языков программирования; при проверке работоспособности модуля на устройстве или эмуляторе установлено соответствие основных выполняемых функций спецификации.	– Устный опрос – Тестирование – Интерпретация результатов наблюдений за деятельностью обучающегося
ОК 01	– Обоснованность выбора методов и подходов при разработке. – Адекватная оценка эффективности решений.	– Экспертное наблюдение за выполнением работ
ОК 02	– Использование различных источников (документация, форумы, научные статьи). – Применение современных инструментов разработки и отладки.	
ОК 05	– Грамотность оформления документации и кода. – Ясность устных пояснений при защите проектов.	
ОК 09	– Умение работать с официальной документацией (Android, Apple) на английском языке. – Понимание терминов.	

4. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ МЕЖДИСЦИПЛИНАРНОГО КУРСА

МДК.01.03 Разработка мобильных приложений

№	Наименование темы	Результаты обучения (освоенные умения и знаний)	ПК, ОК	Текущий контроль успеваемости	Промежуточная аттестация
1	2	3	4	5	6
ПМ.02. Раздел 7 Разработка приложений для мобильных платформ: *МДК.02.07* Разработка приложений для мобильных платформ					
Тема 1.1 Введение в мобильную разработку					
1.	Введение в мобильную разработку. Сравнительная характеристика мобильных платформ.	31-311	ОК 01, ОК 02, ОК 05, ОК 09 ПК 2.6 - ПК 2.9	Устный опрос, тестирование	Экзамен
2.	Виды мобильных приложений. Нативные приложения, веб-приложения, гибридные приложения				
3.	Основные языки для разработки мобильных приложений. Инструменты для разработки мобильных приложений.				
4.	Знакомство с языком Kotlin. Переменные. Типы данных. Конструкции. Установка инструментария и настройка среды Intellij Idea для разработки мобильных приложений. Установка Android Studio и Android SDK. Интерфейс Android Studio.				
5.	Основы языка программирования Kotlin. Переменные и типы данных. Линейное программирование в Kotlin. Программирование ветвлений. Логические выражения	31-311 У1-У7		Отчет по лабораторному занятию	
6.	Программирование циклов в Kotlin. Цикл for. Цикл do-while. Одномерные массивы				
7.	Понятие мобильной операционной системы. Архитектура Android	31-312		Устный опрос, тестирование	
8.	Структура типичного мобильного приложения: Activity; Service; ContentProvider; BroadcastReceiver				
Тема 1.2. Основные элементы управления. Ресурсы. Activity и Intent					
9.	Основы создания интерфейса. Определение интерфейса в файле xml Структура проекта Android Studio. Корневые папки App и Gradle Scripts. Рабочая область для XML-файлов.	312-320	ОК 01, ОК 02, ОК 05, ОК 09 ПК 2.6 - ПК 2.9	Устный опрос, тестирование	
10.	Контейнеры. LinearLayout, RelativeLayout, TableLayout, FrameLayout, вложенные FrameLayout. Адаптивность интепфейсов.				
11.	Создание простого приложения под Android, запуск эмулятора. Структура типичного мобильного приложения. Выгрузка макета в Android Studio	31-311 У8-У20		Устный опрос, тестирование	
12.	Использование LinearLayout, RelativeLayout, TableLayout, FrameLayout, вложенные FrameLayout				

Тема 1.3. Основные элементы управления. Ресурсы. Activity и Intent					
13.	Основные элементы управления. TextView, EditText, Button. Всплывающие окна. Toast, метод Toast.makeText()	312-316	ОК 01, ОК 02, ОК 05, ОК 09 ПК 2.6 - ПК 2.9	Устный опрос, тестирование	
14.	Основные элементы управления. Checkbox, ToggleButton, RadioButton, DatePicker, TimePicker, Ползунок SeekBar.				
15.	Создание элементов пользовательского интерфейса. Определение размеров. Ширина и высота элементов.	312-316 У8-У20		Отчет по лабораторному занятию	
16.	Верстка макетов. Основные элементы управления ConstraintLayout.				
17.	Верстка макетов. Виджеты. Обработка нажатий. Список.				
18.	Работа с ресурсами. Ресурсы строк Ресурсы dimension. Ресурсы Color и установка цвета	322-325		Устный опрос, тестирование	
19.	Activity и жизненный цикл приложения. Файл манифеста AndroidManifest.xml				
20.	Введение в Intent. Запуск Activity. Передача данных между Activity. Сериализация. Получение результата из Activity.				
21.	Жизненный цикл приложения. Взаимодействие между Activity				
22.	Компоненты приложения. Приоритеты приложений и состояния процессов				
23.	Создание приложения Калькулятор. Разработка Activity для регистрации/авторизации пользователя. Запуск второй Activity	322-325 У21-У34		Отчет по лабораторному занятию	
24.	Обработка событий: переключение между экранами. Работа с Intent и запуск внешних Activity.				
Тема 1.4 Работа с изображениями					
25.	Ресурсы изображений, ImageView	323-329	ОК 01, ОК 02, ОК 05, ОК 09 ПК 2.6 - ПК 2.9	Устный опрос, тестирование	
26.	Классы для работы с графикой. Создание меню				
27.	Создание приложения с загрузкой изображений. ImageView.	323-329 У20-У34		Отчет по лабораторному занятию	
28.	Google Maps. Карта, камера, события				
Тема 1.5 Работа с мультимедиа					
29.	Работа с видео, воспроизведение аудио	326-329	ОК 01, ОК 02, ОК 05, ОК 09 ПК 2.6 - ПК 2.9	Устный опрос, тестирование	
30.	Создание проекта для воспроизведения видео- и аудиоинформации	326-329 У30-У34		Отчет по лабораторному занятию	
***Тема 1.6* Работа с файловой системой**					
31.	Чтение и сохранение файлов. Размещение файлов во внешнем хранилищ	326-329	ОК 01, ОК 02, ОК 05, ОК 09 ПК 2.6 - ПК 2.9	Устный опрос, тестирование	
32.	Создание приложения по работе с файлами	327-329 У32-У34		Отчет по лабораторному занятию	
2 семестр					
Тема 1.1 Основы Swift. Операторы управления					

33.	Знакомство с операционной системой iOS. Архитектура iOS	З1-З11 У1-У7	ОК 01, ОК 02, ОК 05, ОК 09 ПК 2.6 - ПК 2.9	Устный опрос, тестирование	
34.	Знакомство с интегрированной средой разработки XCode				
35.	Знакомство с языком Swift. Переменные. Константы. Аннотация типов				
36.	Числовые типы данных. Операции с числами. Логический тип данных				
37.	Строковые типы данных. Операции со строками. Преобразование типов данных				
38.	Установка среды разработки Xcode. Создание и настройка проекта. Тестирование кода в Playground	З1-З11 У8-У20		Отчет по лабораторному занятию	
39.	Создание первого кода. Переменные. Константы. Типы данных.				
40.	Преобразование типов данных				
41.	Операции с числами. Логические операторы.				
42.	Операции со строками. Интерполяция. Конкатинация				
Тема 1.2 Основы Swift. Операторы управления.					
43.	Опциональные типы данных	З1-З12	ОК 01, ОК 02, ОК 05, ОК 09 ПК 2.6 - ПК 2.9	Устный опрос, тестирование	
44.	Контейнерные типы данных. Кортежи				
45.	Массивы				
46.	Контейнерные типы данных. Множества. Словари				
47.	Операторы управления. Оператор условия if				
48.	Оператор раннего выхода guard. Оператор ветвления switch				
49.	Операторы повтоения while и repeat while. Операторы continue и break				
50.	Оператор повторения for				
51.	Функции. Замыкания				
52.	Опциональные типы данных. Контейнерные типы данных	З1-З12 У8-У20		Отчет по лабораторному занятию	
53.	Оператор условия if. Оператор ветвления switch.				
54.	Операторы повтоения while и repeat while. Оператор повторения for.				
Тема 1.3 Введение в ООП и ПОП					
55.	Знакомство с элементами ООП и ПОП	З13-З22	ОК 01, ОК 02, ОК 05, ОК 09 ПК 2.6 - ПК 2.9	Устный опрос, тестирование	
56.	Перечисления. Свойства перечислений. Инициализатор перечислений				
Тема 1.4 **Объектные типы данных**					
57.	Структуры. Свойства в структурах	З23-З32	ОК 01, ОК 02, ОК 05, ОК 09 ПК 2.6 - ПК 2.9	Устный опрос, тестирование	
58.	Классы. Наследование классов				
59.	Свойства. Сабскрипты				
60.	Протоколы. Расширения				
61.	Консольное приложение "Сумма двух чисел".	З23-З29 У21-У34		Отчет по лабораторному занятию	
62.	Функции. Функциональные типы				
63.	Замыкания. Возможности замыканий. Создание собственного типа данных. Перечисления				
64.	Структуры. Свойства в структурах				
65.	Классы. Наследования классов. Свойства. Сабскрипты. Протоколы				

66.	Знакомство с основами SwiftIU. Оптимизация мобильного приложения. Паттерны проектирования				
Самостоятельная работа обучающихся:					
– Через SDK Manager установить пакет Google Usb Driver – Разработать схему запуска приложения на реальном устройстве Android Разработать мобильное приложение: – Разработать мобильное приложение «Taxi» Изучить теоретический материал и составить тезисы (краткий конспект): – Визуальные стили и темы. Изображения. Анимация. – Элементы построения интерфейса мобильного приложения					

5. ОЦЕНОЧНЫЕ СРЕДСТВА ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ

МДК.02.07 Разработка приложений для мобильных платформ

5.1. Типовые задания для оценки знаний и умений

Тема 1.1 Основные платформы и языки разработки мобильных приложений

Задание для устного опроса по темам

1. Перечислить основные платформы мобильных приложений?
2. Дать характеристику ОС Android?
3. Перечислить достоинства и недостатки ОС Android?
4. Дать характеристику ОС Apple IOS?
5. Перечислить достоинства и недостатки ОС Apple IOS?
6. Дать сравнительный анализ рассмотренных мобильных платформ?
7. Что такое нативные приложения, достоинства и недостатки?
8. Что такое веб-приложения, достоинства и недостатки?
9. Что такое гибридные приложения, достоинства и недостатки?
10. Что такое кроссплатформенные приложения, достоинства и недостатки?
11. На каких языках разрабатываются мобильные приложения для Android
12. Основные языки, на которых пишут приложения для iOS
1. Достоинства и недостатки языка Kotlin
2. Рассказать о переменных и типах данных языка Kotlin
3. Раскрыть Условные выражения, Операции отношения, Условные конструкции.
4. Раскрыть цикл For, while, while - do..while
5. Операторы continue и break
6. Какие функции обеспечивают Мобильные операционные системы?
7. Перечислить основные уровни архитектуры Android?
8. Что представляет собой уровень ядра?
9. Дать характеристику уровню библиотек и среды выполнения?
10. Основные компоненты уровня фреймворка приложений?
11. Дать характеристику уровня приложений
12. Что называется SDK?
13. Что входит в состав Android SDK?
14. Где происходит отладка и тестирование приложения для Android?
15. Почему нужно использовать эмулятор Android-устройства?
16. Достоинства эмулятор Android-устройства
17. Для чего каждому приложению Android присваивается уникальный идентификатор?
18. Как создается безопасная среда для приложения Android?

Тестовые вопросы

Тема 1.1 Основные платформы и языки разработки мобильных приложений

Лекция 1

Вопросы с множественным выбором (выберите один правильный ответ)

1. **Какой язык программирования НЕ используется для разработки под Android?**
   a) Java

b) Kotlin
c) Swift
d) C++

2. **Основным барьером защиты iOS, согласно лекции, является:**
   a) Открытый исходный код
   b**) Строгий контроль приложений в AppStore и закрытость системы**
   c) Наличие антивируса по умолчанию
   d) Низкая популярность системы

3. **Почему разработка под Android может быть более сложной для разработчиков?**
   a) Из-за отсутствия качественных сред разработки
   b) **Из-за высокого разнообразия устройств и размеров экранов (фрагментации)**
   c) Из-за использования устаревшего языка программирования
   d) Из-за низкой скорости работы виртуальной машины

4. **Главное преимущество открытости платформы Android для пользователя:**
   a) Высокий уровень безопасности от хакерских атак
   b) Низкая стоимость устройств
   c) **Гибкие и простые возможности для персонализации**
   d) Отсутствие пиратства

5. **Основная причина, по которой разработчики часто выпускают платные приложения сначала для iOS:**
   a) Среда разработки XCode удобнее
   b) Язык Swift проще для изучения
   c) **Низкий уровень пиратства среди пользователей iOS**
   d) На iOS проще разрабатывать из-за фрагментации

6. Вопросы на соответствие

Характеристика	Платформа (Android / iOS)
1. Закрытая файловая система	**iOS**
2. Операционная система с открытым исходным кодом	**Android**
3. Высокая автономность работы	**iOS**
4. Предрасположенность к хакерским атакам	**Android**

Тестовые вопросы
Тема 1.2 Основные платформы и языки разработки мобильных приложений

Лекция 2

Вопросы с множественным выбором (выберите один правильный ответ)

1. **Приложение, разработанное на «родном» для платформы языке (Kotlin для Android, Swift для iOS), называется:**
   a) Веб-приложение
   b) Гибридное приложение
   c) **Нативное приложение**
   d) Кроссплатформенное приложение

2. **Главное преимущество веб-приложений перед другими типами:**
   a) Высокая скорость работы и стабильность
   b) Полный доступ к функциям устройства (камера, GPS)
   c) **Кроссплатформенность и отсутствие необходимости загрузки из магазина**
   d) Возможность работать без подключения к интернету

3. **Какой тип приложения сочетает в себе кроссплатформенность веб-технологий и возможность распространяться через магазины приложений?**
   a) Нативное
   b) Веб-приложение
   c**) Гибридное**
   d) Системное

4. **Основная идея кроссплатформенной разработки заключается в:**
   a) Создании приложения, идеально оптимизированного под одну платформу
   b) Использовании только нативных языков программирования
   c) **Создании приложений для разных платформ с одной кодовой базы**
   d) Отказе от использования магазинов приложений

5. **Какой из перечисленных недостатков характерен и для гибридных, и для веб-приложений?**
   a) Высокая стоимость и длительные сроки разработки
   b**) Низкая скорость и стабильность работы по сравнению с нативными**
   c) Работа только на одной целевой платформе
   d) Невозможность отправлять push-уведомления

6. **Установите соответствие между типом приложения и его ключевой характеристикой:**

Характеристика	Тип приложения
Разрабатывается на HTML, CSS, JavaScript	2.a) Нативное
Наивысшая производительность и доступ к функциям ПО устройства	1.b) Веб-приложение
«Оболочка» нативная, «начинка» на веб-технологиях	3.c) Гибридное
Единая кодовая база для iOS и Android	4.d) Кроссплатформенное

7. **Назовите по одному основному преимуществу нативного приложения и веб-приложения.**
Ответ:
Нативное: *Максимальная функциональность и скорость работы / Полный доступ к функциям устройства (ПО) / Не требует интернета для работы*.
Веб-приложение: *Максимальная функциональность и скорость работы / Полный доступ к функциям устройства (ПО) / Не требует интернета для работы.*

8. **Какой тип приложения вы бы рекомендовали для быстрого и бюджетного запуска проекта, который должен быть доступен пользователям как iOS, так и Android через магазины приложений? Обоснуйте свой ответ, исходя из достоинств и недостатков.**
Ответ:

   Гибридное или **Кроссплатформенное** приложение.
   Обоснование: Эти подходы позволяют разработать одно приложение для двух платформ, что значительно дешевле и быстрее, чем создание двух отдельных нативных приложений. При этом, в отличие от веб-приложений, их можно публиковать в магазинах App Store и Google Play, что упрощает их распространение и поиск для пользователей.

9. **Почему для гибридных приложений, в отличие от веб-приложений, необходимо проходить процесс проверки в магазинах приложений (App Store, Google Play)?**
Ответ:

   Потому что гибридные приложения **загружаются и устанавливаются на устройство пользователя из магазина** как обычное приложение. Магазины проверяют все приложения, которые в них публикуются, на соответствие правилам безопасности, дизайна и контента. Веб-приложения работают в браузере и не устанавливаются через магазины, поэтому не проходят такую проверку.

Лекция 3

1. **Какой ключевой слово используется для объявления неизменяемой переменной в Kotlin?**

   a) var

   b) val

   c) const

   d) let

2. **Какой тип данных в Kotlin используется для хранения целых чисел?**

   a) Float

   b) Double

   c) Int

   d) String

3. **Какая функция используется для ввода данных с консоли в Kotlin?**

   a) readLine()

   b) input()

   c) scan()

   d) read()

4. **Что выведет следующий код?**

```
fun main() {
val x = 10
println(x / 3) }
```

a) 3

b) 3.333

c) 3.0

d) Ошибка компиляции

5. **Какой оператор используется для выхода из цикла досрочно?**

a) stop

b) exit

c) break

d) return

6. **Установите соответствие между типом данных и его описанием:**

1. Int
2. Boolean
3. String
4. Float

a) Логический тип
b) Число с плавающей точкой
c) Целое число
d) Строка символов

Ключ к тестам

1. b) val
2. c) Int
3. a) readLine()
4. a) 3
5. c) break
6. 1 → c) Целое число
   2 → a) Логический тип
   3 → d) Строка символов
   4 → b) Число с плавающей точкой

Лекция 4

1. **Какой компонент архитектуры Android отвечает за управление жизненным циклом приложения?**
   a) Диспетчер пакетов (Package Manager)
   b) Диспетчер активности (Activity Manager)

c) Диспетчер окон (Window Manager)
d) Диспетчер ресурсов (Resource Manager)

2. **На каком уровне архитектуры Android работают драйверы дисплея, камеры и клавиатуры?**
   a) Уровень приложений
   b) Уровень фреймворка приложений
   c) Уровень библиотек и среды выполнения
   d) Уровень ядра
3. **Какая библиотека отвечает за работу с реляционными базами данных в Android?**
   a) WebKit
   b) SQLite
   c) Bionic
   d) SurfaceManager
4. **Что такое API в контексте Android?**
   a) Аппаратный интерфейс устройства
   b) Набор правил и готового кода для взаимодействия между программами
   c) Язык программирования для разработки приложений
   d) Тип мобильного устройства
5. **Какой механизм позволяет приложениям получать доступ к данным других приложений?**
   a) Диспетчер телеметрии (Telephony Manager)
   b) Контент-провайдеры (Content Providers)
   c) Диспетчер уведомлений (Notification Manager)
   d) Система представлений (View System)

Часть 2: Вопросы на соответствие

6. **Установите соответствие между уровнями архитектуры Android и их компонентами:**

   1. Уровень ядра
   2. Уровень библиотек и среды выполнения
   3. Уровень фреймворка приложений
   4. Уровень приложений

   a) Activity Manager, Package Manager, Window Manager
   b) Драйверы дисплея, камеры, клавиатуры
   c) Встроенные приложения: почта, календарь, браузер
   d) Библиотеки WebKit, SQLite, SurfaceManager

Часть 3: Открытые вопросы

7. **Опишите назначение уровня фреймворка приложений. Приведите три примера диспетчеров, которые на нем работают, и их функции.**
8. **Почему в архитектуре Android используется модифицированное ядро Linux? Какие задачи оно решает?**
9. **Как Android обеспечивает безопасность данных приложений, учитывая открытость архитектуры?**

10. **В чем преимущество того, что все приложения в Android имеют равный статус? Приведите пример.**

Ключ к тестам

Часть 1:

1. b) Диспетчер активности (Activity Manager)
2. d) Уровень ядра
3. b) SQLite
4. b) Набор правил и готового кода для взаимодействия между программами
5. b) Контент-провайдеры (Content Providers)

Часть 2:

6. 1 → b) Драйверы дисплея, камеры, клавиатуры
   2 → d) Библиотеки WebKit, SQLite, SurfaceManager
   3 → a) Activity Manager, Package Manager, Window Manager
   4 → c) Встроенные приложения: почта, календарь, браузер

Часть 3:

7. **Уровень фреймворка приложений** предоставляет API для взаимодействия с системными сервисами. Примеры диспетчеров:
   - **Activity Manager**: управляет жизненным циклом приложений.
   - **Package Manager**: управляет установкой и удалением приложений.
   - **Notification Manager**: позволяет приложениям отображать уведомления.
8. **Модифицированное ядро Linux** используется как основа Android для управления аппаратными ресурсами, обеспечения безопасности и многозадачности. Оно адаптировано для мобильных устройств, поддерживает энергоэффективность и частые обновления для совместимости с новыми технологиями.
9. **Безопасность обеспечивается** через изоляцию процессов, управление разрешениями и закрытый доступ к данным по умолчанию. Приложения могут обмениваться данными только через разрешенные механизмы, такие как контент-провайдеры, с явным согласия пользователя.
10. **Равный статус приложений** позволяет пользователям и разработчикам заменять стандартные приложения (например, браузер, клавиатуру) на альтернативные. Это способствует кастомизации и конкуренции. Пример: установка браузера Chrome вместо стандартного.

Лекция 5

Часть 1: Вопросы с выбором ответа (один правильный)

1. **Что такое Android SDK?**

   a) Операционная система для мобильных устройств

   b) Инструментальные средства разработки мобильных приложений для ОС Android

c) Язык программирования для создания приложений
d) Аппаратная платформа для тестирования устройств

2. **Какова основная цель использования эмулятора Android в процессе разработки?**
   a) Заменить собой физическое устройство на постоянной основе
   b) Отлаживать и тестировать приложения в реальной среде выполнения без установки на физическое устройство
   c) Увеличить скорость работы приложения
   d) Обойти систему разрешений Android
3. **Что представляет собой «песочница» (sandbox) в контексте безопасности Android?**
   a) Виртуальная машина для запуска приложений
   b) Изолированная среда безопасности, в которой работает каждое приложение
   c) Специальный режим для тестирования игр
   d) Хранилище для системных файлов
4. **Для чего используется уникальный идентификатор (UID), присваиваемый каждому приложению?**
   a) Для идентификации приложения в Google Play Маркете
   b) Для отслеживания рейтинга приложения пользователями
   c) Для изоляции процессов и управления доступом к ресурсам на уровне системы
   d) Для шифрования данных приложения
5. **Когда пользователь предоставляет приложению разрешения на доступ к ресурсам (например, камере или контактам)?**
   a) Во время первого запуска приложения
   b) В любой момент через Настройки устройства
   c) Во время установки приложения
   d) Только если приложение разработано Google

Часть 2: Вопросы на соответствие

6. **Установите соответствие между понятием и его описанием:**

   1. Android SDK
   2. Эмулятор Android
   3. UID (User Identifier)
   4. «Песочница» (Sandbox)

   a) Утилита, позволяющая запускать и тестировать приложения в виртуальной среде.
   b) Набор инструментов для разработки приложений под Android.
   c) Изолированная среда выполнения, ограничивающая доступ приложения к системе.
   d) Уникальный идентификатор, назначаемый приложению системой для управления доступом.

Часть 3: Открытые вопросы

7. **Опишите принцип работы «песочницы» в Android. Как он способствует безопасности системы?**
8. **Каковы преимущества использования эмулятора на начальном этапе разработки по сравнению с физическим устройством?**
9. **При каком условии два разных приложения могут получать доступ к файлам друг друга? Опишите механизм.**
10. **Почему процесс приложения может быть уничтожен системой Android? Как это связано с управлением ресурсами?**

Ключ к тестам

Часть 1:

1. b) Инструментальные средства разработки мобильных приложений для ОС Android
2. b) Отлаживать и тестировать приложения в реальной среде выполнения без установки на физическое устройство
3. b) Изолированная среда безопасности, в которой работает каждое приложение
4. c) Для изоляции процессов и управления доступом к ресурсам на уровне системы
5. c) Во время установки приложения

Часть 2:

6. 1 → b) Набор инструментов для разработки приложений под Android.
   2 → a) Утилита, позволяющая запускать и тестировать приложения в виртуальной среде.
   3 → d) Уникальный идентификатор, назначаемый приложению системой для управления доступом.
   4 → c) Изолированная среда выполнения, ограничивающая доступ приложения к системе.

Часть 3:

7. **Принцип работы «песочницы»**: Каждое приложение запускается в изолированном процессе со своим уникальным UID. По умолчанию оно имеет доступ только к собственным компонентам и данным. Это предотвращает несанкционированный доступ одного приложения к данным или коду другого, обеспечивая базовый уровень безопасности.
8. **Преимущества эмулятора**:
   - Позволяет проводить первоначальную отладку и тестирование без риска повреждения физического устройства или его файловой системы.
   - Дает возможность тестировать приложение на разных версиях ОС и конфигурациях устройств, не имея их физически.
   - Ускоряет цикл разработки, так как развертывание на эмуляторе часто происходит быстрее.

9. **Доступ к файлам друг друга**: Два приложения могут получать доступ к файлам друг друга, если им присвоен **один и тот же UID**. Для этого оба приложения должны быть **подписаны одним и тем же сертификатом разработчика**. Это настраивается в манифестах приложений.
10. **Уничтожение процесса**: Система Android может уничтожить процесс приложения, когда он больше не нужен (например, пользователь закрыл приложение) или когда системе **требуется освободить оперативную память** для других, более приоритетных приложений (например, работающих в foreground). Это механизм управления ресурсами для обеспечения отзывчивости системы в условиях ограниченной памяти.

Лекция 6

1. **Что такое Android Studio?**
   a) Операционная система для смартфонов
   b) Язык программирования
   c) Интегрированная среда разработки (IDE) для Android
   d) Виртуальное устройство
2. **Какая папка является корневой в проекте Android Studio?**
   a) manifests и java
   b) app и Gradle Scripts
   c) res и layout
   d) drawable и values
3. **В какой папке размещается исходный код приложения (классы Java/Kotlin)?**
   a) manifests
   b) java
   c) res
   d) Gradle Scripts
4. **Какой файл содержит информацию о компонентах приложения, таких как Activity и Service?**
   a) build.gradle
   b) AndroidManifest.xml
   c) settings.gradle
   d) strings.xml
5. **Для чего предназначена папка mipmap?**
   a) Для хранения всех изображений приложения
   b) Для хранения XML-файлов макетов экрана
   c) Для хранения только значков (иконок) приложения
   d) Для хранения строк и цветов
6. **Какой файл описывает все модули в проекте?**
   a) build.gradle (Project)
   b) build.gradle (Module: app)
   c) settings.gradle
   d) AndroidManifest.xml

7. **Какой элемент интерфейса Android Studio показывает иерархию компонентов на экране в виде "дерева"?**
   a) Palette (Палитра)
   b) Design Editor (Визуальный редактор)
   c) Component Tree (Дерево компонентов)
   d) Attributes Panel (Инспектор)
8. **Какой режим в редакторе макетов отображает только контуры View-компонентов?**
   a) Design
   b) Blueprint
   c) Split
   d) Text
9. **Для чего предназначена папка manifests?**
   a) Для хранения изображений приложения
   b) Для хранения файлов конфигурации и манифеста приложения
   c) Для хранения исходного кода на Java/Kotlin
   d) Для хранения Gradle-скриптов
10. **В какой папке хранятся значки приложения (иконки)?**
    a) drawable
    b) layout
    c) mipmap
    d) values

Часть 2: Вопросы на соответствие

6. **Установите соответствие между папками и их содержимым:**

   1. res/layout
   2. res/values
   3. res/drawable
   4. manifests

   a) XML-файлы для построения пользовательского интерфейса
   b) Файлы с изображениями
   c) Файлы конфигурации приложения (AndroidManifest.xml)
   d) Файлы со строками, цветами, размерами

Часть 3: Открытые вопросы

7. **Опишите назначение и содержимое трех основных папок внутри папки app.**
8. **В чем разница между режимами Design и Text при работе с XML-файлами макета?**
9. **Какие возможности предоставляет визуальный редактор (Design Editor) при верстке с использованием View элементов?**
10. **Что такое Gradle Scripts и для чего они используются в проекте Android Studio?**

Ключ к тестам

Часть А: Вопросы с множественным выбором

1. c) Интегрированная среда разработки (IDE) для Android
2. b) app и Gradle Scripts
3. b) java
4. b) AndroidManifest.xml
5. c) Для хранения только значков (иконок) приложения
6. c) settings.gradle
7. c) Component Tree (Дерево компонентов)
8. b) Blueprint
9. b) Для хранения файлов конфигурации и манифеста приложения
10. c) mipmap

Часть 2:

6. 1 → a) XML-файлы для построения пользовательского интерфейса
   2 → d) Файлы со строками, цветами, размерами
   3 → b) Файлы с изображениями
   4 → c) Файлы конфигурации приложения (AndroidManifest.xml)

Часть 3:

7. **Основные папки внутри app:**
   - **manifests**: содержит AndroidManifest.xml с конфигурацией приложения
   - **java**: содержит исходный код на Java/Kotlin
   - **res**: содержит ресурсы приложения (изображения, макеты, строки)
8. **Разница между Design и Text:**
   - **Design** - графическое представление макета, визуальное редактирование
   - **Text** - текстовое представление XML-кода, прямое редактирование разметки
9. **Возможности визуального редактора:**
   - Palette - палитра элементов интерфейса
   - Component Tree - иерархия элементов
   - Design Editor - визуальное представление макета
   - Attributes Panel - настройки выбранного элемента
   - Возможность переключения между режимами Design/Blueprint
10. **Gradle Scripts** - это набор скриптов для автоматизации сборки проекта:
    - settings.gradle - описывает модули проекта
    - build.gradle - конфигурация сборки для проекта и модулей
    - Обеспечивают управление зависимостями, настройку сборки и т.д.

Лекция 7

Тест по теме: «Элементы управления и контейнеры ViewGroup»

Часть 1: Вопросы с единственным правильным ответом

1. **Какой класс является базовым для всех визуальных элементов в Android?**

    а) Widget
    б) ViewGroup
    в) View
    г) Object
2. **Какой элемент используется для отображения текста без возможности его редактирования?**

    а) EditText
    б) Button
    в) TextView
    г) Label
3. **От какого класса наследуется Button?**

    а) View
    б) CompoundButton
    в) TextView
    г) CheckBox
4. **Какой элемент позволяет пользователю выбрать только один вариант из нескольких?**

    а) CheckBox
    б) ToggleButton
    в) RadioButton
    г) Switch
5. **Какой атрибут задаёт уникальный идентификатор для элемента?**

    а) android:name
    б) android:id
    в) android:identity
    г) android:tag

Вопросы с множественным выбором

6. **Какие из перечисленных элементов наследуются от CompoundButton?**

    а) CheckBox
    б) TextView
    в) RadioButton
    г) ToggleButton
7. **Какие атрибуты отвечают за размеры View?**

    а) android:width
    б) android:layout_width

в) android:height
г) android:layout_height

*Часть 2:**Установите соответствие между элементом и его описанием:***

Элемент	Описание
1. ProgressBar	а) Позволяет выбрать значение рейтинга в виде звёздочек.
2. SeekBar	б) Показывает, что программа выполняет длительную операцию.
3. RatingBar	в) Выпадающий список с выбором одного значения.
4. Spinner	г) Слайдер для выбора значения путём перемещения ползунка.

Часть 3: Открытые вопросы

1. **Чем отличается атрибут** android:gravity от android:layout_gravity?
2. **Для чего используется атрибут** android:padding**?**
3. **Почему RadioButton обычно используется в группе?**

Ключ к тестам

Часть 1:

1. в) View
2. в) TextView
3. в) TextView
4. в) RadioButton
5. б) android:id
6. а) CheckBox, в) RadioButton, г) ToggleButton
7. б) android:layout_width, г) android:layout_height

Часть 2:

1. ProgressBar → б)
2. SeekBar → г)
3. RatingBar → а)
4. Spinner → в)

Часть 3:

1. android:gravity определяет выравнивание содержимого внутри самого View, а android:layout_gravity — положение самого View внутри родительского контейнера.
2. android:padding задаёт внутренние отступы между содержимым элемента и его границами.

3. **RadioButton** используется в группе, потому что позволяет выбрать только один вариант из нескольких предложенных.

Лекция 7 (Продолжение темы Контейнеры ViewGroup)

Тест по теме: «Контейнеры ViewGroup в Android»

1. Что такое ViewGroup?
а) Видимая кнопка на экране
б) Базовый класс для всех видимых элементов
в) Невидимый контейнер, организующий расположение дочерних элементов
г) Тип данных для хранения информации

2. Кто является родительским классом для FrameLayout, LinearLayout и ConstraintLayout?
а) View
б) ViewGroup
в) Adapter
г) LayoutManager

3. Какой контейнер размещает дочерние элементы друг на друге в виде стека?
а) LinearLayout
б) ConstraintLayout
в) FrameLayout
г) RecyclerView

4. С помощью какого атрибута во FrameLayout можно управлять положением дочернего элемента?
а) orientation
б) layout_gravity
в) weight
г) constraint

5. Для какого сценария НЕ рекомендуется использовать FrameLayout?
а) Создание оверлеев (наложений)
б) Отображение фонового изображения
в) Реализация сложного адаптивного списка с сотнями элементов
г) Попеременное отображение загрузочного индикатора и контента

6. В каком направлении LinearLayout НЕ может располагать дочерние элементы?
а) Вертикально (vertical)
б) Горизонтально (horizontal)
в) Диагонально
г) Оба варианта (а) и (б) верны, а (в) — нет

7. Какой атрибут в LinearLayout позволяет задать относительные размеры дочерних элементов?
а) layout_gravity
б) orientation

в) weight
г) layout_constraintDimensionRatio

8. Основной принцип работы ConstraintLayout — это использование...
а) ...весов (weight).
б) ...ограничений (constraints).
в) ...стека.
г) ...адаптеров (adapter).

9. Что такое «chain» в ConstraintLayout?
а) Новый тип анимации
б) Способ связывания нескольких элементов для их взаимного выравнивания и распределения пространства
в) Тип данных для адаптера
г) Метод оптимизации производительности

10. Какой контейнер рекомендуется использовать для замены устаревшего RelativeLayout?
а) FrameLayout
б) LinearLayout
в) ConstraintLayout
г) RecyclerView

11. Какой компонент RecyclerView отвечает за управление расположением и размерами элементов (например, делает список или сетку)?
а) ViewHolder
б) Adapter
в) LayoutManager
г) ItemDecorator

12. Какой компонент RecyclerView служит для оптимизации производительности, переиспользуя уже созданные View?
а) Adapter
б) LayoutManager
в) ViewHolder
г) SnapHelper

13. С помощью какого метода в адаптере RecyclerView можно реализовать отображение элементов разных типов?
а) onCreateViewHolder()
б) onBindViewHolder()
в) getItemViewType()
г) getItemCount()

14. RecyclerView может отображать данные в виде:
а) Только вертикального списка
б) Только сетки
в) Вертикального списка, сетки или карусели (горизонтального списка с прокруткой по страницам)
г) Только статичного набора виджетов

15. Для какого сценария использования RecyclerView НЕ подходит?

а) Отображение большого списка контактов

б) Отображение новостной ленты с разными типами контента (текст, фото, видео)

в) Создание простого экрана с одним изображением и двумя кнопками

г) Отображение каталога товаров в виде сетки

Ключ к тесту

1. **в)** Невидимый контейнер, организующий расположение дочерних элементов
2. **б)** ViewGroup
3. **в)** FrameLayout
4. **б)** layout_gravity
5. **в)** Реализация сложного адаптивного списка с сотнями элементов (для этого используется RecyclerView)
6. **в)** Диагонально
7. **в)** weight
8. **б)** ...ограничений (constraints)
9. **б)** Способ связывания нескольких элементов для их взаимного выравнивания и распределения пространства
10. **в)** ConstraintLayout
11. **в)** LayoutManager
12. **в)** ViewHolder
13. **в)** getItemViewType()
14. **в)** Вертикального списка, сетки или карусели (горизонтального списка с прокруткой по страницам)
15. **в)** Создание простого экрана с одним изображением и двумя кнопками (для этого проще использовать ConstraintLayout или LinearLayout)

Лекция 8

Тест по теме: «Компоненты приложения. Приоритеты приложений и состояния процессов»

1. Какой главный параметр Android-приложения позволяет использовать компоненты других приложений?

а) Собственный изолированный процесс

б) Возможность запроса на исполнение части кода другого приложения

в) Прямое копирование кода компонента

г) Наличие собственной реализации всех компонентов

2. Какой из перечисленных компонентов НЕ является одним из четырех основных типов компонентов в Android?

а) Activity
б) Service
в) Broadcast Receiver
г) Intent
д) Content Provider

3. В каком состоянии Activity полностью перекрыто другим Activity и не видно пользователю?
а) Active/Running
б) Paused
в) Stopped

4. Какой метод жизненного цикла Activity вызывается, когда оно снова становится видимым после состояния Stopped?
а) onCreate()
б) onStart()
в) onRestart()
г) onResume()

5. Процесс какого типа будет закрыт системой в самую последнюю очередь при нехватке памяти?
а) Процесс заднего фона
б) Видимый процесс
в) Служебный процесс
г) Процесс переднего плана

6. Для чего предназначен компонент Service?
а) Для предоставления пользовательского интерфейса
б) Для выполнения операций в фоновом режиме
в) Для реакции на широковещательные сообщения
г) Для предоставления доступа к данным

7. Почему Services рекомендуется запускать в отдельном потоке?
а) Потому что они не имеют методов жизненного цикла
б) Чтобы они могли взаимодействовать с Activity
в) Чтобы они не блокировали главный поток и пользовательский интерфейс
г) Потому что они не могут работать в главном потоке

8. Какой единственный метод жизненного цикла имеет Broadcast Receiver?
а) onCreate()
б) onStart()
в) onReceive()
г) onDestroy()

9. Что может сделать Broadcast Receiver при получении сообщения, чтобы привлечь внимание пользователя?
а) Запустить новый Service

б) Заблокировать главный поток
в) Использовать NotificationManager (вибрация, звук и т.д.)
г) Создать новое Activity

10. Что предоставляет компонент Content Provider?
а) Фоновое выполнение вычислений
б) Доступ к данным (чтение, обновление) для своего и других приложений
в) Рассылку широковещательных сообщений
г) Пользовательский интерфейс для настройки приложения

Ключ к тесту:

1. **б) Возможность запроса на исполнение части кода другого приложения**
2. **г) Intent** (Intent — это механизм для описания операции, а не отдельный тип компонента)
3. **в) Stopped**
4. **в) onRestart()**
5. **г) Процесс переднего плана**
6. **б) Для выполнения операций в фоновом режиме**
7. **в) Чтобы они не блокировали главный поток и пользовательский интерфейс**
8. **в) onReceive()**
9. **в) Использовать NotificationManager (вибрация, звук и т.д.)**
10. **б) Доступ к данным (чтение, обновление) для своего и других приложений**

Лекция 9

Тест: «Манифест Android-приложения»

1. Что такое AndroidManifest.xml?
а) Файл с исходным кодом приложения
б) Файл с пользовательским интерфейсом
в) Файл, определяющий структуру, функции и взаимодействие приложения с системой
г) Файл для хранения настроек пользователя

2. Какие два элемента являются обязательными в файле манифеста?
а) <activity> и <service>
б) <manifest> и <application>
в) <uses-permission> и <uses-sdk>
г) <provider> и <receiver>

3. Для чего в манифесте указывается имя пакета?
а) Для красоты
б) Чтобы указать версию приложения
в) Чтобы служить уникальным идентификатором приложения
г) Чтобы указать список библиотек

4. Какой префикс обычно имеют имена атрибутов в манифесте (за исключением некоторых атрибутов корневого элемента)?
а) app:

б) manifest:
в) android:
г) os:

5. Элемент <instrumentation>**...**
а) Всегда требуется для работы приложения
б) Используется только при разработке и удаляется при публикации
в) Определяет главное активность приложения
г) Указывает требуемые разрешения

6. Какой элемент манифеста используется для объявления разрешения, которое должно быть предоставлено приложению для доступа к камере?
а) <permission>
б) <uses-feature>
в) <uses-permission>
г) <uses-configuration>

7. Что из перечисленного НЕ является одной из ключевых функций манифеста?
а) Идентификация приложения
б) Описание компонентов приложения
в) Хранение пользовательских данных
г) Объявление требований и разрешений

8. Какое из следующих правил относительно элементов манифеста является НЕВЕРНЫМ?
а) Элементы <manifest> и <application> являются обязательными.
б) Элементы могут содержать символьные данные внутри себя.
в) Большинство элементов может присутствовать в нескольких экземплярах.
г) Элементы, как правило, могут быть неупорядочены.

9. Элемент <activity-alias>**...**
а) Должен всегда идти до элемента <activity>, для которого он создает псевдоним
б) Может находиться в любом месте файла манифеста
в) Должен всегда идти после элемента <activity>, для которого он создает псевдоним
г) Является обязательным элементом

10. Магазины приложений, такие как Google Play Store, используют манифест для...
а) Компиляции исходного кода приложения
б) Управления индексированием, листингом и распространением приложений
в) Отображения пользовательского интерфейса приложения
г) Хранения паролей пользователей

Ключи к тесту

1. **в)** Файл, определяющий структуру, функции и взаимодействие приложения с системой
2. **б)** <manifest> и <application>
3. **в)** Чтобы служить уникальным идентификатором приложения
4. **в)** android:

5. **б)** Используется только при разработке и удаляется при публикации
6. **в)** <uses-permission>
7. **в)** Хранение пользовательских данных. Эта функция не упоминается среди задач манифеста.
8. **б)** Элементы могут содержать символьные данные внутри себя. Это неверно, так как "все значения устанавливаются только с помощью атрибутов, а не как символьные данные внутри элемента".
9. **в)** Должен всегда идти после элемента <activity>, для которого он создает псевдоним. Это исключение из общего правила неупорядоченности элементов.
10. **б)** Управления индексированием, листингом и распространением приложений

1. Определите, на каком уровне архитектуры Android находится структура приложения (Application Framework)?
    a. **На втором**
    b. На первом
    c. На четвертом
    d. На третьем
2. Что означает аббревиатура Android SDK?
    a. Software Development Key
    b. Single Development Kit
    c. **Software Development Kit**
    d. Нет правильного ответа.
    e. нет ответа
3. Можно ли менять внешний вид интерфейса Android Studio?
    a. **Да**
    b. Нет
    c. Только при покупке полной версии продукта
    d. Нет правильного ответа
4. При запуске Android Studio по умолчанию открываются два файла. Какие?
    a. MainActivity.java, strings.xml
    b. style.xml, activity_main.xml
    c. **MainActivity.java, activity_main.xml**
    d. Нет правильного ответа
5. Сколько стоит аккаунт разработчика Android?
    a. **25$**
    b. 30$
    c. 0$
    d. 15$
6. Что означает аббревиатура IDE?
    a. **Integrated Development Environment**
    b. Integrated Development Environment Application
    c. Integrated Dev Electronics
    d. Integrated Driver Electronics

7. Что означает система с отрытым исходным кодом?
    a. Это значит, что система может видоизменяться по запросу в любое время.
    b. Это значит, что компания открыта для сотрудничества с любой желающий компанией или разработчиками.
    c. **Это значит, что любой разработчик может бесплатно скачать исходный код, изменить его и опубликовать свою версию**
    d. Нет правильного ответа
8. На каких языках программирования можно разрабатывать приложения для Android? Отметьте несколько правильных вариантов:
    a. PHP
    b. Ruby
    c. **C++**
    d. Pascal
    e. **Kotlin**
    f. **Java**
    g. CSS
    h. Bootstrap
    i. нет ответа
9. В какой папке находятся файлы, создаваемые системой в процессе компиляции?
    a. src
    b. build
    c. libs
    d. mipmap
10. Что содержит в себе папка values?
    a. Цвета, строки и стили
    b. Значки и иконки
    c. Файлы макета активити
    d. Изображения и картинки
11. Обязательным атрибутом макета LinearLayout является:
    a. text
    b. orientation
    c. gravity
    d. context
12. В каком файле хранятся строковые ресурсы приложения?
    a. style.xml
    b. strings.xml
    c. activity_main.xml
    d. colors.xml
13. Единица измерения для изменения размеров элементов:
    a. px
    b. sp
    c. dp
    d. pt
14. С помощью какого атрибута, можно выделить текст курсивом?
    a. textSize
    b. text

c. textAppearance
d. textStyle

15. Как группирует LinearLayout дочерние элементы?
    a. Относительно других элементов
    b. В табличном виде
    c. С привязкой друг к другу
    d. По горизонтали, по вертикали
16. Элемент Spinner представляет собой…
    a. Выпадающий список
    b. Плавающую кнопку
    c. Двухуровневый текст
    d. Картинку с кнопкой
17. В каком файле находятся все идентификаторы ресурсов?
    a. MainActivity.java
    b. A.java
    c. C.java
    d. R.java
18. Какой атрибут отвечает за установку внешних отступов?
    a. gravity
    b. padding
    c. margin
    d. bottom нет ответа

Лабораторные занятия № 1-2

1. Установка инструментария и настройка среды для разработки мобильных приложений
2. Установка среды разработки мобильных приложений с применением виртуальной машины

Тема 1.3. Структура и принципы функционирования мобильного приложения

Задание для устного опроса по темам

1. В какой папке в проекте определяется использование стилей?
2. Что содержит в себе папка values?
3. В каком элементе меняется цвет текста при создании собственной темы?
4. Что означает система с отрытым исходным кодом?
5. На каких языках программирования можно разрабатывать приложения для Android?
6. Можно ли менять внешний вид интерфейса Android Studio?
7. Для получения элементов по id в классе Activity используется метод?
8. Основные сведения, содержащие в объекте Intent?
9. Как называется набор средств программирования, который содержит инструменты, необходимые для создания, компиляции и сборки мобильного приложения?
10. С какой целью инструмент Intel* Graphics Performance Analyzers (Intel* GPA) System Analyzer используется в среде разработки Intel* Beacon Mountain?
11. Какой движок баз данных используется в ОС Android?
12. Какой класс является основным строительным блоком для компонентов пользовательского интерфейса (UI), определяет прямоугольную область экрана и

отвечает за прорисовку и обработку событий?

13. Какой слушатель используется для отслеживания события касания экрана устройства?
14. В какой папке необходимо разместить XML файлы, которые определяют все меню приложения?
15. Для получения элементов по id в классе Activity используется метод?
16. Обязательно ли делать описание (contentDescription) к картинке?
17. Можно ли выбрать несколько значений в переключателе Switch?
18. Что не характерно для неявных объектов Intent?
19. Какой метод запускает новый экземпляр компонента Service?
20. Как определяется тип передаваемых данных Uri?
21. Перечислить составляющие кроссплатформенной разработки?
22. Описать свойства современных IDE?
23. Что такое Элементы управления вводом?
24. Что такое Графический интерфейс пользователя?
25. Перечислить Атрибуты TextView
26. Перечислить ключевые атрибуты, которые можно задать у кнопок?
27. Что такое Всплывающие окна. Toast
28. Что является ключевым копнентом для создания визуального интерфейса в приложении Android?
29. Что необходимо для создания пользовательского интерфейса?
30. Для чего необходим Класс View?
31. Перечислить набора готовых Представлений?
32. Перечислить подклассы View?
33. Для чего используется класс R?
34. В каком каталоге проекта хранятся ресурсы приложения?
35. Из каких компонентов состоит идентификатор ресурса?

Тестовые вопросы

Тема 1.3. Структура и принципы функционирования мобильного приложения

1. Назовите атрибут у EditText, который автоматически исправляет вводимый текст
    a. textCapWords
    b. textAutoCorrect
    c. textCorrect
    d. textWordCorrect
2. Какова продолжительность показа константы Toast.LENGTH_LONG?
    a. 2 секунды
    b. 3,5 секунды
    c. 1,5 секунды
    d. 2,5 секунды
    e. 3 секунды
3. С помощью какого атрибута можно выравнивать дочерние элементы по центру?
    a. layout_gravity = “center”
    b. orientation = “center”
    c. layout_below = “center”
    d. gravity = “center”

4. Можно ли изменить Company Domain после создания проекта?
    a. Да
    b. Нет
    c. Зависит от домена
    d. Нет правильного ответа
5. Для получения элементов по id в классе Activity используется метод…
    a. src
    b. findId()
    c. findViewById()
    d. findViewById(id)
6. Короткое сообщение, содержащее информацию для пользователя, но не требует ни ввода, ни действий -
    a. Toast
    b. Message
    c. Text
    d. EditText
7. Атрибут элемента Spinner для отображения массива данных:
    a. entry
    b. array
    c. string-array
    d. entries
8. Можно ли выбрать несколько значений в переключателе Switch?
    a. Да, не больше двух
    b. Да, не больше трех
    c. Нет, нельзя выбрать несколько значений
    d. Зависит от исходных данных
9. Обязательно ли делать описание (contentDescription) к картинке?
    a. Нет, но желательно
    b. Да, иначе выдаст ошибку
    c. Нет правильного ответа
10. Что выведет на экран следующее выражение: android: text=”@string/hello”?
    a. Значение ресурсной строки с именем hello
    b. Значение” hello”
    c. Значение @string/hello
    d. Ничего не выведет
11. Явные объекты Intent -
    a. Указывают на компонент, который требуется запустить
    b. Не содержат имени конкретного компонента
    c. Объявляют действие, которое требуется выполнить
    d. Обрабатывают запросы приложения
12. Основные сведения, содержащие в объекте Intent: Выберите несколько правильных ответов.
    a. Данные
    b. Объекты Bundle
    c. Категория
    d. Неявный объект

e. Службы
f. Действие
g. Явный объект
h. Имя компонента

13. С помощью имени компонента объект Intent становится…
    a. Неявным
    b. Явным
    c. Действительным
    d. Недействительным
14. ACTION_SEND относится к …
    a. Действию объекта Intent
    b. Категории объекта Intent
    c. Данным объекта Intent
    d. Ни к чему не относится
15. Что не характерно для неявных объектов Intent?
    a. Указывают на компонент, который требуется запустить.
    b. Не содержат имени конкретного компонента.
    c. Объявляют действие, которое требуется выполнить.
    d. Обрабатывают запросы приложения.
16. Как определяется тип передаваемых данных Uri?
    a. Данными объекта Intent.
    b. Категорией объекта Intent.
    c. Действием объекта Intent.
    d. Определяется автоматически.
17. Объект Intent используется:
    a. Для запуска разметки.
    a. Для запуска Android Studio.
    b. Для запуска сообщений.
    c. Для запуска Activity.
18. CATEGORY_BROWSABLE относится к …
    a. Действию объекта Intent.
    b. Категории объекта Intent.
    c. Данным объекта Intent.
    d. Ни к чему не относится.
19. Какой метод запускает новый экземпляр компонента Service?
    a. startMainActivity();
    b. startService();
    c. startActivity();
    d. startIntent();
20. Тип данных MIME это…
    a. Спецификация объектов Intent.
    b. Спецификация объектов Uri, работающие со строками.
    c. Спецификация для передачи по сети файлов различного типа.
    d. Все ответы верны.

Лабораторные занятия № 3-7

1. Введение в платформу Android. Инструментарий среды разработки мобильных приложений.
2. Первое приложение. Структура типичного мобильного приложения
3. Создание элементов пользовательского интерфейса. Определение размеров. Ширина и высота элементов.
4. Верстка макетов. Основные элементы управления. ConstraintLayout. LinearLayout. RelativeLayout
5. Верстка макетов. Виджеты. Обработка нажатий. Список.

Тема 1.3.3. Создание пользовательского интерфейса

Задание для устного опроса по темам

1. Что такое манифест?
2. Перечислить основные типы компонентов архитектуры Android?
3. Что такое Активность (Activity)?
4. Что такое Сервисы (Services)?
5. Что такое Контент-провайдеры (Content Provider)?
6. Для чего служит Приемник широковещательных сообщений (Broadcast Receiver)?
7. Что такое Намерения (Intent), Виджеты, Уведомления (система пользовательских уведомлений)?
8. В какой папке необходимо разместить XML файлы, которые определяют все меню приложения?
9. Какого класса является приемник широковещательных сообщений?
10. В каком файле находятся все идентификаторы ресурсов?
11. Перечислите обязательные атрибуты для всех элементов?
12. Какой метод используется для получения элементов по id в классе Activity?
13. Какова продолжительность показа константы Toast.LENGTH_LONG?
14. Атрибут элемента Spinner для отображения массива каких данных?
15. Какой метод запускает новый экземпляр компонента Activity?
16. Остановка анимации или других текущих действий, которые могут потреблять ресурсы процессора происходят в методе?
17. Какой метод вызывается после того, как сработал onDestroy()?
18. Адаптер, предназначенный для работы с ListView, где данные представляются для списка через курсор, имеющий колонку с именем “id”?
19. С помощью данного метода, можно сохранить текущее состояние Activity (например, при повороте экрана)?
20. Перечислить несколько групп элементов пользовательского интерфейса?
21. Для чего необходим элемент LinearLayout?
22. Перечислить несколько стандартных типов разметок?
23. Описать Программный способ создания разметки?
24. Описать иерархия приоритетов, определяющая порядок, в котором прерывается работа приложений?
25. Охарактеризовать Активный процесс (Foreground Process)?
26. Какие существуют способы доступа к ресурсу?
27. Какие готовые *Представления* предоставляются разработчику системой?
28. Назовите наиболее часто используемые виды *Разметки*.
29. Для чего используется объект Drawable?

30. Назовите два способа определения и инициализации объекта Drawable.
31. Какие графические форматы поддерживаются в Android-приложениях?
32. С помощью каких методов создается мню?
33. Как осуществляется поиск пункта меню?

Тестовые вопросы

Создание пользовательского интерфейса

1. На каком уровне архитектуры Android находятся приложения (Applications)?
    a. На втором
    b. На четвертом
    c. На третьем
    d. На первом
2. Комбинация клавиш, с помощью которой можно запустить проект в Android Studio:
    a. Alt + F10
    b. Ctrl + F10
    c. Shift + F10
    d. Shift + Ctrl + F10
3. Атрибут для смены размера шрифта TextView:
    a. textAppear
    b. textSize
    c. text
    d. textStyle
4. В каком файле находятся все идентификаторы ресурсов?
    a. MainActivity.java
    b. A.java
    c. C.java
    d. R.java
5. GoogleUSBDriver представляет собой…
    a. драйвер для работы с сетью
    b. драйвер для создания виджетов
    c. драйвер для подключения камеры телефона
    d. драйвер для работы с данными телефона
6. В какой папке находится исходный код и ресурсы приложения?
    a. src
    b. build
    c. libs
    d. mipmap
7. Выберите обязательные атрибуты для всех элементов:
    a. match_parent, wrap_content
    b. wrap_content, fill_parent
    c. text, orientation
    d. text, gravity
8. Можно ли создать разметку с помощью кода Java?
    a. Нет
    b. Да

c. Нет правильного ответа

9. Какова продолжительность показа константы Toast.LENGTH_LONG?
    a. 2 секунды
    b. 3,5 секунды
    c. 1,5 секунды
    d. 2,5 секунды
    e. 3 секунды
10. С помощью какого атрибута можно выравнивать дочерние элементы по центру?
    a. layout_gravity = “center”
    b. orientation = “center”
    c. layout_below = “center”
    d. gravity = “center”
11. Для получения элементов по id в классе Activity используется метод…
    a. src
    b. findId()
    c. findViewById()
    d. findViewById(id)
12. Короткое сообщение, содержащее информацию для пользователя, но не требует ни ввода, ни действий -
    a. Toast
    b. Message
    c. Text
    d. EditText
13. Атрибут элемента Spinner для отображения массива данных:
    a. entry
    b. array
    c. string-array
    d. entries
14. Что выведет на экран следующее выражение: android: text=”@string/hello”?
    a. Значение ресурсной строки с именем hello
    b. Значение ”hello”
    c. Значение @string/hello
    d. Ничего не выведет
15. Назовите два вида Intent.
    a. Простые и сложные
    b. Простые и составные
    c. Явные и неявные
16. Можно ли передавать данные через Intent?
    a. Нет
    b. Да
    c. Нет правильного ответа
17. Можно ли посредством Intent перейти в другое приложение?
    a. Нет
    b. Да
    c. Нет правильного ответа
18. Какой метод запускает новый экземпляр компонента Activity?

a. startMainActivity()
b. startService()
c. startIntent()
d. startActivity()

19. Объект Intent используется:
    a. Для запуска разметки
    b. Для запуска Android Studio
    c. Для запуска сообщений
    d. Для запуска широковещательных сообщений
20. С помощью имени компонента объект Intent становится…
    a. Неявным
    b. Явным
    c. Действительным
    d. Недействительным
21. С помощью данного метода, можно сохранить текущее состояние Activity (например, при повороте экрана):
    a. onPause();
    b. Bundle();
    c. onSaveInstanceState();
    d. postDelayed();
22. Тип сохраняемых значений в Bundle:
    a. Имя и значение;
    b. Ключ и параметр;
23. Правильная реализация методов жизненного цикла Activity обеспечивает:
    a. Потребление ценных системных ресурсов, когда пользователь не использует приложение.
    b. Сохранение состояния приложения, если пользователь выходит из него и возвращается позднее.
    c. Закрывается с ошибкой и теряет данные пользователя при повороте экрана.
    d. Все перечисленное.
24. Объекты Handler позволяют:
    a. Вызывать Activity.
    b. Начинать отсчет времени при повороте экрана.
    c. Уничтожать Activity вместе с сохраненным данными.
    d. Планировать выполнение кода через некоторое время.
25. Перезапуск Activity сопровождается методом:
    a. onResume();
    b. onRestart();
    c. onCreate();
    d. onStart();
26. Тип сохраняемых значений в Bundle:
    a. Имя и значение;
    b. Ключ и параметр;
    c. Ключ и значение;
    d. Имя и параметр;
27. Какой метод вызывается после того, как сработал onDestroy()?

a. onStop();
b. onStart();
c. onPause();
d. Метод не вызывается.

28. Остановка анимации или других текущих действий, которые могут потреблять ресурсы процессора происходят в методе:
    a. onResume();
    b. onStart();
    c. onCreate();
    d. onPause();
29. С помощью данного метода, можно сохранить текущее состояние Activity (например, при повороте экрана):
    a. onPause();
    b. Bundle();
    c. onSaveInstanceState();
    d. postDelayed();
    e. нет ответа
30. Приостановка Activity сопровождается методом:
    a. onResume();
    b. onStart();
    c. onCreate();
    d. onPause();
31. Какой элемент предоставляет наиболее простой вариант разметки. Он может быть использован для абсолютного позиционирования элементов с использованием координат.
    a. Grid
    b. Pivot
    c. Panorama
    d. Canvas (+)
32. На базе какого ядра сделано ядро Android?
    a. Windows NT
    b. Linux
    c. FreeBSD
    d. VxWorks
33. Какое название носит виртуальная Java-машина Android?
    a. Dalvik
    b. B ionic
    c. HotSpot
34. Как изменился набор консольных утилит Linux в Android?
    a. Существенно уменьшился
    b. Остался без изменений
    c. Существенно увеличился
    d. В Android нет консольных утилит
35. Является ли logcat полным аналогом gdb?
    a. Да
    b. logcat — полный аналог gdbserver

c. Нет

36. Из-за чего glibc была заменена на Bionic? Смена бренда
    a. Несовместимость glibc со свободными лицензиями
    b. glibc невозможно скомпилировать для ARM
    c. Уменьшение размеров объектного кода
37. Какой байт-код используется в Dalvik?
    a. Обычный
    b. Исключительно свой собственный
    c. Свой собственный, но обычный, может быть, в него сконвертирован
38. Из чего состоит Java API в Android?
    a. Часть стандартных библиотек Java
    b. Стандартные библиотеки Java и пакет javax.
    c. Набор стандартных, популярных и собственных пакетов
    d. Исключительно собственные библиотеки Android
39. Какую функциональность обеспечивает пакет android.view?
    a. Реализация GUI
    b. Работа с камерой
    c. Темы рабочего стола

Лабораторные занятия № 8-9

1. Элементы управления и контейнеры. Класс Activity и ресурсы.
2. Создание графического приложения. Запуск второй Activity

Тема 1.4 Работа с мультимедиа

Задание для устного опроса по темам

1. Перечислить библиотеки для обработки мультимедиа?
2. Перечислить способы аудиозаписи и воспроизведения?
3. Какой самый простой способ инициализации плеера для воспроизведения?
4. Что необходимо для воспроизведения звука? Средствами AudioTrack?
5. Перечислить основной способ воспроизведения звука?
6. Для чего необходима встроенная камера?
7. Каким методом запускается воспроизведение видео?
8. Охарактеризовать Видимый процесс (Visible Process)?
9. Сервисный процесс (Service Process)?
10. Фоновый процесс (Background Process)?
11. Охарактеризовать Пустой процесс (Empty Process)?
12. В каком каталоге храниться файл манифеста?
13. Перечислить основные элементы файла AndroidManifest.xml?
14. Сколько в манифесте может присутствовать экземпляр тега <application>?
15. Как называется система передачи сообщений меду приложениями?
16. Экран, который приложение может показывать пользователям называется?
17. Какой класс наследуется для создания новой Активности. Перечислить состояние Активности?
18. Описать состояние Активное (active или running)?
19. Описать состояние Приостановленное (paused)?
20. Описать состояние Остановленное (stopped)?

21. Описать состояние Неактивное (inactive)?
22. Для чего предназначен Класс Application?
23. Какую возможность дает создание собственной реализации класса Application?
24. Где должена быть зарегистрирована Реализация класса Application?
25. Перечислить основные методы класса Activity?
26. В чем особенность Сборка программных пакетов из исходного кода?

Тестовые вопросы

Тема 1.4 Работа с мультимедиа

1. Для упрощения работы с элементами списка существует специальный класс, под названием:
    a. ListCategory.
    b. ListAdapter.
    c. ArrayAdapter.
    d. ListActivity.
2. ListView представляет собой:
    a. Адаптер, позволяющий заполнить данными список более сложной структуры.
    b. Простейший адаптер, который связывает массив данных.
    c. Прокручиваемый список элементов.
    d. Адаптер, дополняющий BaseAdapter.
3. Адаптер, предназначенный для работы с ListView, где данные представляются для списка через курсор, имеющий колонку с именем “id”:
    a. ArrayAdapter.
    b. SimpleAdapter.
    c. SimpleCursorAdapter.
    d. CursorAdapter.
4. Активити категорий:
    a. Выводят данные, принадлежащие конкретной категории, в частности в виде списка.
    b. Представляют собой операции, наиболее важные для пользователя и простые средства для навигации к ним.
    c. Выводят подробную информацию по конкретной записи, предоставляют пользователю возможность редактирования существующих записей или ввода новых записей.
    d. Нет правильного ответа.
5. Адаптер, дополняющий ResourceCursorAdapter, а также создает компоненты TextView, ImageView из столбцов, содержащихся в курсоре.
    a. ArrayAdapter.
    b. SimpleAdapter.
    c. SimpleCursorAdapter.
    d. CursorAdapter.
6. Для чего существуют адаптеры?
    a. Для сохранения состояния приложения при повороте экрана.
    b. Для упрощения связывания данных с элементами управления.
    c. Для прокручивания списка элементов.
    d. Нет правильного ответа.

e. нет ответа

7. К минусам ListView не относится:
    a. Тяжело реализовывать анимации.
    b. Нет поддержки просмотра изображений.
    c. Не подгружает элементы списка в текстовом виде.
    d. Гораздо менее гибкий виджет, чем необходимо.
    e. нет ответа
8. RecyclerView предназначен для:
    a. Связывания данных.
    b. Обработки данных.
    c. Замены устаревших представлений.
    d. Оптимизации работы со списками.
9. ViewHolder:
    a. Оборачивает набор данных и создает представления для отдельных элементов.
    b. Удерживает все зависимые представления, которые зависят от данных текущего элемента.
    c. Помещает объекты в пределах доступной области.
    d. Определяет отступы вокруг или сверху каждого элемента.
    e. Анимация элементов, во время добавления, удаления или переупорядочивания.
10. К внутренним классам RecyclerView не относится:
    a. ViewHolder
    b. LayoutManager
    c. ItemDecoration
    d. ListView
11. Класс LayoutManager в RecyclerView отвечает за:
    a. Реализацию анимации в макете.
    b. Обработку массива данных.
    c. Размер и позиционирование элементов.
    d. Кэширование элементов ListView.
12. К методам реализации RecyclerView.Adapter относится:
    a. onCreateView ();
    b. getItemDecoration();
    c. onBindViewHolder();
    d. getIntItemCount();
13. ItemAnimator:
    a. Оборачивает набор данных и создает представления для отдельных элементов.
    b. Удерживает все зависимые представления, которые зависят от данных текущего элемента.
    c. Помещает объекты в пределах доступной области.
    d. Определяет отступы вокруг или сверху каждого элемента.
    e. Анимация элементов, во время добавления, удаления или переупорядочивания.
14. С помощью ItemDecoration можно:
    a. Добавить реализацию стандартного списка прокрутки по вертикали и горизонтали.
    b. Добавить отступ к каждому элементу и изменить элемент.
    c. Заменить старый интерфейс адаптера новым базовым классом.

d. Реализовать отображение элементов, которые подстраиваются под произвольный размер.

15. Библиотека загрузки и кэширования изображений для Android:
    a. Square
    b. Recycler
    c. Picasso
    d. Holder
16. В каком файле проекта можно написать разрешение для подключения приложения к интернету?
    a. AndroidManifest.xml
    b. MainActivity.java
    c. Service.java
    d. Class.java
17. RecyclerView относится к семейству:
    a. Spinner
    b. ListView
    c. ViewGroup
    d. Gallery
    e. Adapter
18. Зависит ли Fragment от Activity?
    a. Да.
    b. Нет.
19. Создание Fragment происходит в методе:
    a. onResume();
    b. onRestart();
    c. onCreate();
    d. onStart();
20. Запись несохраненных изменений в Fragment (например, черновик электронного письма) происходит в методе:
    a. onResume();
    b. onPause();
    c. onCreate();
    d. onStart();
21. ListFragment:
    a. Отображает иерархию объектов.
    b. Отображает список элементов.
    c. Отображает диалоговое окно.
    d. Все перечисленное.
22. Метод inflate() принимает следующий аргумент:
    a. Ключ и параметр.
    b. Массив данных.
    c. Объект класса ViewGrop.
    d. Объект Handler.
23. Нужно ли объявлять фрагменты в AndroidManifest.xml?
    a. Да.
    b. Нет.

24. Объекты Handler позволяют:
    a. Вызывать Activity.
    b. Начинать отсчет времени при повороте экрана.
    c. Уничтожать Activity вместе с сохраненным данными.
    d. Планировать выполнение кода через некоторое время.
25. Перезапуск Activity сопровождается методом:
    a. onResume();
    b. onRestart();
    c. onCreate();
    d. onStart();
26. Правильная реализация методов жизненного цикла Activity обеспечивает:
    a. Потребление ценных системных ресурсов, когда пользователь не использует приложение.
    b. Сохранение состояния приложения, если пользователь выходит из него и возвращается позднее.
    c. Закрывается с ошибкой и теряет данные пользователя при повороте экрана.
    d. Все перечисленное.
27. Какой метод вызывается после того, как сработал onDestroy()?
    a. onStop();
    b. onStart();
    c. onPause();
    d. Метод не вызывается.
28. Возобновление Activity сопровождается методом:
    a. onResume();
    b. onStart();
    c. onCreate();
    d. onActivity();
29. Адаптер, позволяющий заполнить данными список более сложной структуры (например, два текста в одной строке списка):
    a. ArrayAdapter.
    b. SimpleAdapter.
    c. SimpleCursorAdapter.
    d. CursorAdapter.
30. К классификации активити в приложении не относится:
    a. Активити верхнего уровня.
    b. Активити данных.
    c. Активити категорий.
    d. Активити детализации.
31. Для чего существуют адаптеры?
    a. Для сохранения состояния приложения при повороте экрана.
    b. Для упрощения связывания данных с элементами управления.
    c. Для прокручивания списка элементов.
32. К минусам ListView не относится:
    a. Тяжело реализовывать анимации.
    b. Нет поддержки просмотра изображений.
    c. Не подгружает элементы списка в текстовом виде.

d. Гораздо менее гибкий виджет, чем необходимо.

33. RecyclerView использует следующий подход:
    a. Помещает объекты в пределах доступной области.
    b. Заменяет устаревшие представления.
    c. Иллюстрирует работу ListView.
    d. Утилизирует и повторно использует элементы.
34. Класс LayoutManager в RecyclerView отвечает за:
    a. Обработку массива данных.
    b. Кэширование элементов ListView.
    c. Расположение элементов на макете.
    d. Реализацию анимации в макете.
35. ItemDecoration:
    a. Оборачивает набор данных и создает представления для отдельных элементов.
    b. Удерживает все зависимые представления, которые зависят от данных текущего элемента.
    c. Помещает объекты в пределах доступной области.
    d. Определяет отступы вокруг или сверху каждого элемента.
    e. Анимация элементов, во время добавления, удаления или переупорядочивания.
36. Объект ViewHolder:
    a. Отвечает за размер и позиционирование элементов в RecyclerView, а также за определение переработки элементов.
    b. Хранит каждый из представлений компонентов внутри поля тега макета, поэтому можно сразу получить к ним доступ.
    c. Оборачивает набор данных и создает представления для отдельных элементов.
    d. Хранит каждый из представлений компонентов снаружи поля тега макета, поэтому можно сразу получить к ним доступ.
37. Класс Adapter в RecyclerView отвечает за:
    a. Реализацию анимации в макете.
    b. Расположение элементов на макете.
    c. Размер и позиционирование элементов.
    d. Создание правильной компоновки для отдельных элементов.
38. Метод onBindViewHolder:
    a. Выполняет привязку объекта ViewHolder.
    b. Определяет метод onCreateViewHolder().
    c. Хранит данные по одному объекту из класса.
    d. Возвращает общее количество элементов в списке.
39. Класс Adapter в RecyclerView отвечает за -
    a. Реализацию анимации в макете.
    b. Обработку массива данных.
    c. Кэширование элементов ListView.
    d. Обеспечивает доступ к базовому набору данных.
40. К методам реализации RecyclerView.Adapter относится:
    a. onCreateViewHolder();
    b. getItemDecoration();
    c. onBindView();
    d. getIntItemCount();

41. Нужно ли объявлять фрагменты в AndroidManifest.xml?
    a. Да.
    b. Нет.
42. При первом отображении пользовательского интерфейса в Fragment вызывается метод:
    a. onResume();
    b. onCreateView();
    c. onCreate();
    d. onStart();
43. DialogFragment:
    a. Отображает иерархию объектов.
    b. Отображает список элементов.
    c. Отображает диалоговое окно.
    d. Все перечисленное.
44. Какие методы отражают жизненный цикл Activity:
    a. onCreate() (+)
    b. onRestart()(+)
    c. onResume()(+)
    d. onRunning()
    e. onClose()
45. Подробная информация о приложении содержится в файле:
    a. AndroidManifest.xml (+)
    b. main.xml
    c. R.j ava
    d. default.properties
46. Для создания всплывающего уведомления необходимо инициализировать объект:
    a. Toast (+)
    b. Messag
    c. TextView
    d. MessageBox
47. Единица измерения dp или dip - это 1/72 дюйма
    a. определяется по физическому размеру экрана дюйм
    b. определяется по физическому размеру экрана абстрактная ЕИ
    c. позволяющая приложениям выглядеть одинаково на различных экранах и разрешениях (+)
48. Тип верстки, при котором позиционирование элементов происходит относительно друг друга и относительно главного контейнера
    a. AbsoluteLayout
    b. FrameLayout
    c. RelativeLayout (+)
49. Какой класс можно использовать для перехода между Activity?
    a. ActivityChanger
    b. Activity Intent (+)
    c. Switcher
50. От какого класса наследуются все элементы управления?
    a. Control

b. Controls
c. Element
d. View (+)

51. Для чего используется класс R?
    a. Класс ресурсов (+)
    b. Класс для управления элементами управления
    c. Класс для работы с потоками
    d. Класс, предназначенный только для доступа к идентификаторам элементов
52. Как программно получить доступ к элементам управления через идентификатор?
    a. getViewById(id)
    b. findViewById(id) (+)
    c. findView(id)
53. В какой библиотеке находится класс Activity?
    a. android
    b. android.Activities
    c. android.app (+)
    d. java.app
54. На основе какой операционной системы была разработана система Android:
    a. Linux (+)
    b. MiniX
    c. Windows
55. Как называется режим использования Android-системы с максимальными правами (аналог администратора Windows):
    a. Суперпользователь (+)
    b. Root-Home пользователь
    c. Android user
    d. Up-User
56. Как называется событие-клик объекта-кнопки:
    a. OnClick (+)
    b. OnClicked
    c. Click
    d. ClickEvent
57. Для добавления в приложение более одного экрана взаимодействия необходимо создание:
    a. Activity (+)
    b. View
    c. Layout
    d. XML-файла
58. Как называется язык разметки Window Phone приложений:
    a. XAML(+)
    b. XML
    c. Java
    d. C#
59. С помощью какого класса можно вывести сообщение в Window Phone приложении:
    a. Toast
    b. MessageBox(+)

c. Message
d. ShowClass

60. Какой элемент предоставляет вариант разметки, элементы, помещенные в который, располагаются в стек горизонтально или вертикально:
    a. StackPanel (+)
    b. Pivot
    c. Panorama
    d. Canvas
61. Какой элемент позволят позиционировать элементы внутри себя максимально гибко предоставляя возможность размещать элементы, используя строки и столбцы.
    a. Grid (+)
    b. Pivot
    c. Panorama
    d. Canvas

Лабораторные занятия № 10-12

1. Всплывающие окна Toast. ToggleButton. RadioButton. Ползунок SeekBar
2. Работа с изображениями. ImageView
3. Google Maps. Создание и настройка проекта. Карта, камера, события

Тема 1.1 Язык Swift и платформа iOS

Задание для устного опроса по темам

1. Что такое Swift? История его создания и развития?
2. Перечислить возможности и особенности Swift
3. Преимущества и недостатки Swift
4. MVC-архитектура.
5. IDE (интегрированная среда разработки) для iOS.
6. Основные компоненты архитектуры. Алгоритм работы архитектуры.
7. Основы Swift?
8. Что такое перечисления (enumerations) в Swift?
9. Каковы три основных типа коллекций в Swift?
10. Объясните архитектуру MVC?
11. Какие функции есть в Swift?
12. Что такое dictionary (словарь) в Swift?
13. В чем разница между операторами == и ===?
14. Что такое расширения?
15. Что такое вложенная функция?
16. Что делает ключевое слово Mutating?
17. В чем разница между функциями и методами?
18. Что такое (autoclosure) автозамыкание в Swift?
19. Как и когда следует использовать(autoclosure) автозамыкание?
20. Как можно улучшить читаемость кода?
21. Можно ли поменять местами две переменные без третьей переменной-помощника?
22. Что такое Force Unwrapping (принудительное разворачивание)?
23. Когда следует использовать Force Unwrapping (принудительное разворачивание)?
24. Перечислите преимущества функций высшего порядка.

25. Для чего используется Guard?
26. Как создать базовый класс в Swift?
27. Для чего нужен оператор вида «??»?
28. Как тестировать приложение без физического устройства?
29. Что такое plist?
30. Для чего нужны Protocols в Swift? Приведите пример.
31. Что делает init() в Swift?
32. Зачем нужен completion handler в Swift?
33. В чем разница между структурами и классами?
34. Что такое Optional в Swift?
35. Как создать Optional в Swift?
36. Что такое необязательная цепочка (Optional Chaining)?
37. Назовите 5 утверждений передачи управления (Control Transfer Statements) и опишите, как их использовать.
38. Что такое typealias в Swift?
39. Как можно создать typealias в Swift?
40. Что такое опциональное связывание (optional binding)

Тестовые вопросы

Тема 1.1 Язык Swift и платформа iOS

1. Что такое бандл iOS приложения file.app?
   A. Папка, в которой содержатся бинарный исполняемый файл и ресурсы
   B. Бинарный исполняемый файл
   C. Ашифрованный zip-архив
   D. Папка с ресурсами приложения
2. В какой папке нужно хранить созданные пользователем критичные данные?
   A. Library/Caches
   B. tmp
   C. Documents
   D. Library/Application Support
3. В какой папке нужно хранить данные, которые в любой момент можно создать или загрузить заново?
   A. Library/Caches
   B. Library/Application Support
   C. tmp
   D. Documents
4. В какой папке нужно создавать файлы, которые нужны только во время одного запуска приложения?
   A. Documents
   B. Library/Application Support
   C. tmp
   D. Library/Caches
5. С помощью какого объекта можно сохранять небольшие данные между запусками приложения?
   A. NSCoreData

B. NSFileManager
C. NSManagedObject
D. NSUserDefaults

6. Как сохранить произвольный класс User в файл?
   A. Реализовать протокол NSCoding, преобразовать объект в NSString и сохранить ее в файл
   B. Реализовать протокол NSJson, преобразовать объект в NSData и сохранить ее в файл
   C. Реализовать протокол NSJson, преобразовать объект в NSString и сохранить ее в файл
   D. Реализовать протокол NSCoding, преобразовать объект в NSData и сохранить ее в файл
7. Какая база данных чаще всего используется в мобильных приложениях?
   A. SQLite
   B. Mongogb
   C. PostgreSQL
   D. MySQL
8. Какой объект нужно создать, чтобы работать с Core Data?
   A. NSManagedObjectModel
   B. NSPersistentStoreCoordinator
   C. NSManagedObjectContext
9. В какой среде разработки происходит программирование для iOS на языке Swift?
   A. Xcode
   B. Visual Studio
   C. Eclipse
   D. Netbeans
10. Где можно запустить Xcode?
    A. На Mac OS и Windows
    B. Только на Mac OS
    C. На Mac OS, Linux, Windows
    D. Только на Windows
11. Что такое MVC?
    A. Model-View-Container
    B. Mapping-View-Container
    C. Many-Various-Containers
    D. Model-View-Controller
12. Какая часть MVC не имеет отношения к интерфейсу приложения?
    A. Model
    B. View
    C. Controller
    D. Все имеют
13. Какая часть MVC связывает данные и интерфейс?
    A. View
    B. Model
    C. Controller
    D. ViewModel
14. Что такое полиморфизм?

A. Ограничение доступа к составляющим объект компонентам
B. Единообразная обработка разнотипных данных
C. Описание новых классов на основе существующих базовых классов
D. Взаимно-однозначное отображение линейных пространств

15. Что такое инкапсуляция?
A. Ограничение доступа к составляющим объект компонентам
B. Единообразная обработка разнотипных данных
C. Описание новых классов на основе существующих базовых классов
D. Повышение общего уровня цен на товары и услуги

16. Что такое наследование?
A. Ограничение доступа к составляющим объект компонентам
B. Единообразная обработка разнотипных данных
C. Описание новых классов на основе существующих базовых классов

Лабораторные занятия № 13 - 16

1. Установка среды разработки Xcode. Создание и настройка проекта. Тестирование кода в Playground.
2. Создание первого кода. Переменные. Константы. Типы данных. Преобразование типов данных.
3. Операции с числами. Логические операторы.
4. Операции со строками. Интерполяция. Конкатинация.

Тема 1.2 Основы Swift. Операторы управления

Задание для устного опроса по темам

1. Как создаются объекты в Swift?
1. Что значит минус перед определением метода?
2. Типы данных. Предназначение типов данных. Числовые типы данных.
3. Операции с числами. Преобразование числовых данных.
4. Контейнерные типы данных. Кортеж.
5. Доступ к элементам кортежа.
6. Редактирование кортежа. Сравнение кортежей.
7. Контейнерные типы данных. Диапазоны.
8. Способы создания диапазонов.
9. Базовые свойства и методы диапазонов
10. Контейнерные типы данных.
11. Массивы. Способы создания массива.
12. Доступ к элементам массива. Тип данных массива.
13. Пустой и многомерный массивы. Операции с массивами.
14. Срезы массивов (ArraySlice).
15. Операции со множествами.
16. Синтаксис множеств. Пустое множество.
17. Что означает модификатор weak на свойстве в Swift?
18. Из какого языка в Swift доступны структуры?
19. Какой метод вызывается в контроллере, когда загрузилось его View?
20. Как показать модальный контроллер vc, который выедет снизу вверх поверх нашего контроллера?

21. Базовые свойства и методы словарей.
22. В каком методе контроллера правильнее всего заполнять его интерфейс?
23. Как изменить иконку этого контроллера, которая отображается в tabbar-e?
24. Как отразить контроллер в AppDelegate?

Тестовые вопросы

Тема 1.2 Основы Swift. Операторы управления

1. За что отвечает View в MVC?
   A. Работа с файлами
   B. Работа с интернетом
   C. Хранение в БД
   D. Интерфейс
2. В какой части MVC можно отнести класс "Кнопка"?
   A. Model
   B. View
   C. Controller
   D. Ни к какому
3. В какой части MVC можно отнести класс "Магазин"?
   A. View
   B. Controller
   C. Model
   D. Ни к какому
4. С каким символом определяются переменные-объекты в Swift
   A. NSString $a
   B. NSString #a
   C. NSString *a
   D. NSString .a
5. Как пишутся булевы "правда" и "ложь" в Swift?
   A. YES и NO
   B. yes и no
   C. true и fals
   D. True и False
6. Как задается массив строк NSArray в Swift?
   A. @{ @"a", @"b", @"c" }
   B. @< @"a", @"b", @"c" >
   C. @[@"a", @"b", @"c"]
   D. @(@"a", @"b", @"c")
7. Как задается словарь NSDictionary в Swift?
   A. @{ @"key1" : @"value1", @"key2" : @"value2" }
   B. @[@"key1" : @"value1", @"key2" : @"value2"]
   C. @(@"key1" => @"value1", @"key2" => @"value2")
   D. @[@"key1", @"value1", @"key2" , @"value2"]
8. Как у объекта person вызвать метод save без параметров
   A. person.save()
   B. person->save

C. person.save
D. [person save]

9. Как создаются объекты в Swift?
   A. Person.new()
   B. [[Person alloc] init]
   C. New Person()
   D. [Person create]
10. Какой класс в Swift представляет строку?
   A. Char
   B. UIStrin
   C. Strin
   D. NSString
11. Как добавить число 3 в NSMutableArray *arr?
   A. [arr addObject:@]
   B. [arr addObject:3]
   C. arr[] = 3
   D. arr <= 3
12. От какого класса наследуются все объекты в Swift?
   A. NSString
   B. NSArray
   C. ob
   D. NSObject
13. Для чего нужны селекторы
   A. Указывать на метод объекта
   B. Указывать на тип объекта
   C. Указывать на значение свойства
   D. Указывать на количество памяти
14. Что значит минус перед определением метода?
   A. Это метод класса
   B. Это статический метод
   C. Это абстрактный метод
   D. Это метод экземпляра
15. Что такое протокол в Swift?
   A. Общее название для группы методов
   B. Базовый класс
   C. Имя одного метода
   D. Тип объекта
16. Что означает модификатор weak на свойстве в Swift?
   A. Эта ссылка учитывается в последнюю очередь при подсчете ссылок на объект
   B. Эта ссылка не учитывается при подсчете ссылок на объект
   C. Эта свойство доступно только на чтение
   D. В этой свойство можно положить только слабые объекты
17. Из какого языка в Swift доступны структуры?
   A. Си
   B. C++
   C. Java

D. Pascal

18. Какими объектами представляются все видимые на экране iPhone прямоугольники?
    A. UIViewController
    B. NSView
    C. UIView
    D. NSString
19. Как задается положение UIView на экране
    A. 4-мя точками x1, x2, y1, y2
    B. Левым верхним и правым нижним углами corner1 и corner2
    C. Центром center и размером size
    D. Прямоугольником CGRect
20. Какой объект отвечает за отображение простых надписей на экране?
    A. UIButton
    B. UIProgress
    C. UIImageView
    D. UILabel
21. Как задать красный цвет текста в UILabel *label
    A. label.color = @"red"
    B. label.color = @"#ff0000"
    C. label.textColor = "red"
    D. label.textColor = [UIColor redColor]
22. Что такое target и action, которые часто используются в Swift?
    A. Имя реагирующего класса и имя метода
    B. Объект, у которого нужно вызвать метод и сам метод, который нужно вызвать
    C. Объект, который реализует протокол и название протокола
    D. Делегат и объект делегата
23. Что такое UITextFieldDelegate?
    A. Имя протокола
    B. Имя делегата
    C. Тип объекта
    D. Объект для отображения ввода текста
24. Через какой объект можно послать сообщение, на которое смогут подписаться многие объекты?
    A. NSPostManager
    B. NSLocalNotifications
    C. NSNotificationCenter
    D. NSMessages
25. Какой метод вызывается в контроллере, когда загрузилось его View?
    A. viewWillAppear
    B. viewWillLoad
    C. viewDidLoad
    D. viewDidAppear
26. Какой контроллер отвечает за навигацию в iOS-приложениях?
    A. UITableViewController
    B. UINavigationController
    C. UITabBarController

D. UIViewController

27. Какой контроллер умеет отображать несколько табов?
    A. UITableViewController
    B. UINavigationController
    C. UITabBarController
    D. UIViewController
28. Как показать модальный контроллер vc, который выедет снизу вверх поверх нашего контроллера?
    A. [self presentViewController:vc animated:YES completion:nil]
    B. [self showModalViewController:vc]
    C. [self presentModalViewController:vc]
    D. [vc presentViewController: self]
29. Как отобразить контроллер в AppDelegate?
    A. Вызвать у self.window метод presentViewControlle
    B. Вызвать у этого контроллера метод presentViewController
    C. Добавить этот контроллер в window через addSubview
    D. Установить self.window.rootViewController на него
30. Как показать контроллер vc в UINavigationController при нажатии на кнопку?
    A. [self.navigationController pushViewController:vc animated:YES]
    B. [self.navigationController popViewController:vc animated:YES]
    C. [vc presentViewController:self.navigationController animated:YES]
    D. [self.view addSubview:vc.view]
31. В каком методе контроллера правильнее всего заполнять его интерфейс?
    A. viewDidUnload
    B. viewDidLoad
    C. viewWillAppear
    D. viewWillLayoutSubviews
32. Как поставить кнопку в левой части navigation bar-а, если контроллер лежит внутри UINavigationController?
    A. self.leftBarButtonItem = ...
    B. self.navigationItem.leftBarButtonItem = ...
    C. self.navigationController.navigationBar.leftBarButtonItem = ...
    D. [self.navigationBar addSubview:...
33. Как изменить иконку этого контроллера, которая отображается в tab bar-е?
    A. self.tabBarItem.image = ..
    B. self.tabBarImage = ...
    C. self.tabBarController.image = ...
    D. self.tabBarController.tabBar.image = ...
34. Что означает стрелочка со свободным левым концом в Storyboard, указывающая на контроллер?
    A. Это невидимый контроллер
    B. Этот контроллер отобразится при запуске приложения
    C. Это главный контроллер приложения
    D. Любое нажатие на кнопки в приложении отобразит этот контроллер
35. На каком сайте мы задаем название приложения, иконку, скриншоты?
    A. developer.apple.com

B. itunesconnect.apple.com
C. appstore.apple.com
D. itunes.apple.com

Лабораторные занятия № 17-19

1. Опциональные типы данных. Контейнерные типы данных
2. Оператор условия if. Оператор ветвления switch.
3. Операторы повторения while и repeat while. Оператор повторения for.

Тема 1.3. Объектные типы данных

Задание для устного опроса по темам

1. Операторы управления.
2. Виды операторов управления.
3. Глобальная функция утверждений.
4. Виды операторов управления.
5. Оператор условия. Сокращенный и стандартный синтаксис оператора условия.
6. Виды операторов управления.
7. Расширенный синтаксис оператора условия.
8. Тернарный оператор условия.
9. Операторы управления. Виды операторов управления.
10. Оператор ветвления. Дополнительные операторы в операторе ветвления.
11. Виды операторов управления
12. Операторы повторения с предварительной проверкой условия и постпроверкой условия.
13. Использование дополнительных операторов в операторах повторения.
14. Виды операторов управления.
15. Оператор повторения для элементов последовательности.
16. Функции. Характеристика функций.
17. Синтаксис и алгоритм выполнения функции.
18. Сквозные параметры функции.
19. Функциональные типы.
20. Возможности функций.
21. Замыкания. Виды замыканий. Замыкающие выражения. Метод sorted.
22. Возможности замыкания. Замыкания как ссылочный тип.
23. Замыкания.
24. Автозамыкания.
25. Реализация автозамыканий

Тестовые вопросы

Тема 1.3.8 Объектные типы данных

1. Где нужно задавать self.title контроллера, чтобы он отобразился в tab bar-e?
   A. В инициализаторе init
   B. Во viewDidLoad
   C. Во viewWillAppear
   D. В любом месте
2. Если мы хотим создать tab’ы одновременно с навигацией, как правильнее разместить

контроллеры?

   A. Корневой UINavigationController и в нем один UITabBarControlle
   B. Корневой UITabBarController и в каждом tab’е по UINavigationController
   C. Корневой UINavigationController и в нем по одному UITabBarController на каждый tab
   D. На каждый tab по одному UINavigationController и в каждом из них свой UITabBarController

3. Какой класс позволяет отображать список ячеек с прокруткой?
   A. UIScrollView
   B. UIView
   C. UITableView
   D. UICellView
4. Какой протокол нужно реализовать, чтобы сообщить UITableView, какие строки нужно показывать?
   A. UITableViewDataSource
   B. UITableViewCells
   C. UITableViewRows
   D. UITableViewDataCells
5. Как задается количество отображаемых в UITableView строк?
   A. Нужно задать свойство tableView.rowsCount
   B. Нужно вернуть число в методе tableView:numberOfRowsInSection
   C. Нужно заполнить массив tableView.rows
   D. Нужно заполнить массив tableView.sections
6. Для чего нужен reuseIdentifier у ячейки UITableViewCell
   A. Чтобы отличать ячейки друг от друга
   B. Чтобы задавать содержимое ячеек
   C. Чтобы сообщить таблице число ячеек
   D. Чтобы переиспользовать одни и те же ячейки при прокрутке по возможности несколько раз
7. Какой тип ячейки позволяет отобразить строку крупного текста и под ней строку более мелкого текста?
   A. UITableViewCellStyleTwoRows
   B. UITableViewCellStyleSubtitle
   C. UITableViewCellStyleValue1
   D. UITableViewCellStyleDefault
8. Как реагировать на нажатие ячейки в таблице?
   A. Нужно задать у tableView target и action
   B. Нужно подписаться на уведомление UITableViewDidSelectRow
   C. Нужно установить tableView.delegate и реализовать метод tableView:didSelectRowAtIndexPath
   D. Нужно добавить в каждую ячейку по кнопке и реагировать на их нажатие
9. Какой факт не верен про UITableViewController?
   A. Его корневое view это UITableView
   B. Он по умолчанию реализует протокол UITableViewDelegate
   C. Он по умолчанию реализует протокол UITableViewDataSource
   D. В нем можно задать количество ячеек через свойство self.rowsCount

10. Какое view используется для отображения интерфейса с прокруткой?
    A. UILargeView
    B. UIContentView
    C. UILabe
    D. UIScrollView
11. Как задать какой прокручиваемый размер должен быть у UIScrollView?
    A. Переопределить метод contentSizeForScrollView: и вернуть там нужный размер
    B. Установить нужный frame
    C. Добавить addSubview: нужного размера
    D. Установить свойство contentSize
12. Почему UIScrollView удобно отображать внутри UINavigationController-а?
    A. Он умеет автоматически задавать contentSize для UIScrollView
    B. Он запоминает положение прокрутки
    C. Он автоматически сдвигает содержимое под статус и навигейшн бары
    D. По-другому UIScrollView отобразить нельзя
13. Какое view используется для отображения html?
    A. UISiteView
    B. UISafariView
    C. UIWebView
    D. UIScrollView
14. Как называется файл, в котором через Interface Builder можно задать все экраны приложения?
    A. storyboard-файл
    B. xib-файл
    C. nib-файл
    D. ib-файл
15. Что означает стрелочка со свободным левым концом в Storyboard, указывающая на контроллер?
    E. Это невидимый контроллер
    F. Этот контроллер отобразится при запуске приложения
    G. Это главный контроллер приложения
    H. Любое нажатие на кнопки в приложении отобразит этот контроллер
16. Как в коде контроллера получить ссылку на view, созданную через Interface Builder?
    A. Создать IBOutlet свойство для этой view
    B. Создать IBAction для этой view
    C. Это невозможно
    D. Добавить ее в коде через addSubview
17. Как сделать так, чтобы по нажатию кнопки, заданной через Interface Builder, выполнялся определенный метод в контроллере?
    A. Создать IBOutlet на эту кнопку
    B. Это невозможно
    C. Создать IBAction метод
    D. Создать делегат с этим методом
18. Как сделать так, чтобы по нажатию кнопки, заданной через Interface Builder, в UINavigationController пушился другой контроллер?

A. Перетащить левой кнопкой мыши от этой кнопки до контроллера и выбрать «push»
B. Перетащить правой кнопкой мыши от этой кнопки до контроллера и выбрать «push»
C. Перетащить правой кнопкой мыши от контроллера до этой кнопки и выбрать «push»
D. Перетащить левой кнопкой мыши от контроллера до этой кнопки и выбрать «push»

19. Как сделать так, чтобы контроллер в InterfaceBuilder имел тип созданного вами класса CustomViewController?
    A. В коде CustomViewController указать идентификатор нужного контроллера
    B. Задать CustomViewController в поле identifier в InterfaceBuilder для этого контроллера
    C. Не нужно ничего сделать, Xcode понимает это автоматически
    D. Задать CustomViewController для этого контроллера в InterfaceBuilder в поле Custom Class
20. Как создать IBOutlet на созданный в Interface Builder label?
    A. Перетащить левой кнопкой мыши от этого label в место в коде, где создаются свойства заполнить нужные параметры
    B. Перетащить левой кнопкой мыши от нужного свойства в коде на этот label и заполнить нужные параметры
    C. Перетащить правой кнопкой мыши от нужного свойства в коде на этот label и заполнить нужные параметры
    D. Перетащить правой кнопкой мыши от этого label в место в коде, где создаются свойства и заполнить нужные параметры
21. Как создать IBAction для созданной в Interface Builder кнопки?
    A. Перетащить левой кнопкой мыши от этой кнопки в место в коде, где реализуются методы и заполнить нужные параметры
    B. Перетащить правой кнопкой мыши от этой кнопки в место в коде, где реализуются методы и заполнить нужные параметры
    C. Перетащить правой кнопкой мыши от нужного метода к этой кнопке и заполнить нужные параметры
    D. Перетащить левой кнопкой мыши от нужного метода к этой кнопке и заполнить нужные параметры
22. Какой тип таблицы надо выбрать, чтобы можно было прямо в Interface Builder заполнять ячейки данными?
    A. Dynamic cells
    B. Interface builder cells
    C. Manual cells
    D. Static cells
23. Можно ли в Interface Builder задать цвет текста для UILabel?
    A. нет
    B. да
24. Что такое CocoaPods?
    A. Система для подключения библиотек к проекту
    B. Система для юнит-тестирования

C. Библиотека для работы с сетью
D. Библиотека для Dependency Injection

25. Как называется самая популярная библиотека для отправки GET и POST запросов?
    A. CocoaPods
    B. AFNetworking
    C. UIKit
    D. NSURL
26. Как можно загрузить содержимое текстового файла по ссылке на сервере?
    A. Вызвать метод loadString у NSURL, указывающего на этот файл
    B. Создать строку через метод initWithData:encoding: и передать туда ссылку на файл
    C. Создать строку через метод initWithConnection: и передать туда соединение
    D. Создать строку через метод initWithContentsOfURL:encoding:error: и передать туда ссылку на файл
27. С помощью какой конструкции языка Swift пишутся асинхронные вызовы при работе сетью через AFNetworking?
    A. Делегаты
    B. Протоколы
    C. Блоки
    D. target/action
28. Что делает вызов dispatch_async(dispatch_get_main_queue(), ^{ }) ?
    A. Выполняет блок в главном потоке
    B. Выполняет блок в фоновом потоке
    C. Выполняет блок отложено через секунду
    D. Сохраняет блок в файл
29. С помощью какого класса можно распарсить JSON?
    A. NSJSONSerialization
    B. NSCoding
    C. NSData
    D. NSKeyedUnarchiver
30. Как чаще всего реализуют авторизацию мобильных приложений на сервере через JSON?
    A. Через cookie
    B. Через basic-авторизацию
    C. Отсылая логин и пароль в каждом запросе
    D. Через токен
31. Какой получится объект, если распарсить следующий JSON: {"result" : "success"} ?
    A. NSDictionary
    B. NSArray
    C. NSString
    D. NSData
32. Какое свойство UIView нельзя анимировать?
    A. backgroundColor
    B. alpha
    C. frame
    D. hidden
33. С помощью какого объекта можно задавать свойства для всех view данного типа?

A. UIStyle
B. UITemplate
C. UICss
D. UIAppearance

34. Какое свойство UIWindow задает цвет активных элементов в кнопках, слайдерах, прогресс-барах и т.д.?
    A. tintColor
    B. backgroundColor
    C. buttonColor
    D. activeColor
35. Через какой объект мы можем задать границу, скругленные углы или тень для UIView?
    A. view.shadows
    B. view.props
    C. view.layer
    D. view.graphics
36. Какие файлы удобнее всего получать программисту от дизайнера для реализации интерфейса?
    A. ppt презентация с анимациями png
    B. psd всего интерфейса
    C. png с общим видом экранов
    D. для каждого отдельного элемента
37. Как задается цвет текста в UILabel?
    A. label.textColor = [UIColor redColor];
    B. label.color = @"red";
    C. label.color = @"#ff0000";
    D. label.backgroundColor = [UIColor redColor];
38. Что делает следующий код [UIView animateWithDuration:1 animations:^{ self.smallView.alpha = 0; }]; ?
    A. Анимированно показывает smallView
    B. Анимированно скрывает smallView
    C. Анимированно сдвигает smallView
    D. Анимированно поворачивает smallView
39. Что такое "универсальное приложение" для iOS?
    A. Приложение, которое поддерживает и iPhone, и iPad
    B. Приложение, которое работает на всех старых iPhone
    C. Приложение, которое поддерживает все версии iOS
    D. Приложение, которое работает на iPhone 11 и iPhone 12
40. Что делает следующий код if (UI_USER_INTERFACE_IDIOM() == UIUserInterfaceIdiomPad) { ... } ?
    A. Выполняет код, если приложение запущено на iPhone
    B. Выполняет код, если приложение скомпилировано для iPad
    C. Выполняет код, если приложение запущено на iPad
    D. Выполняет код, если приложение универсальное
41. Какие существуют два способа для автоматического изменения интерфейса при изменении размера экрана на iOS?
    A. autofixing и autoswitching

B. autoresizing и autorebooting
C. layouting и resizing
D. autosizing и autolayout

42. Что на самом деле задают constraint-ы в autolayout?
A. Формулу энтропии
B. Дифференциальные уравнения второго порядка
C. Систему линейных уравнений и неравенств
D. Систему квадратных уравнений
43. Какой контроллер доступен только на iPad?
A. UIPageViewController
B. UINavigationController
C. UITabBarController
D. UISplitViewController
44. Сколько максимум может быть Storyboard в одном приложении?
A. Не ограничено
B. Одна
C. Две
D. Три
45. Обязательно ли создавать Storyboard отдельно для iPhone, отдельно для iPad?
A. Нет
B. Да
46. Можно ли разрешить приложению на iPhone только портретную ориентацию, а на iPad только ландшафтную?
A. Нет
B. Да
47. Может ли iPad интерфейс загружаться из Storyboard для iPhone?
A. Да
B. Нет
48. Какого датчика нет в iPhone?
A. Акселерометр
B. Гироскоп
C. Барометр
D. Компас
49. Что измеряет акселерометр?
A. Силу
B. Скорость вращения
C. Линейную скорость
D. Ускорение
50. Что измеряет гироскоп?
A. Силу
B. Линейную скорость
C. Ускорение
D. Скорость вращения
51. Как указать CLLocationManager, надо ли включать GPS или нет при определении геопозиции?
A. Вызвать метод stopUpdatingLocation

B. Установить свойство gpsEnabled
C. Установить точность работы через desiredAccuracy
D. Установить свойство satelliteEnabled

52. Какие данные содержит класс CLHeading?
A. Широта и долгота
B. Направление относительно севера
C. Высота относительно уровня моря
D. Часовой пояс

53. С помощью какого контроллера проще всего воспроизвести видео?
A. UIVideoView
B. AVPlayer
C. UIVideoViewController
D. MPMoviePlayerViewController

54. Для чего нужен объект UIImagePickerController?
A. Отображение фотогалереи
B. Получение фотографий с камеры и из фотоальбома
C. Получение фотографий из интернета
D. Воспроизведение слайд-шоу

55. Наследники какого объекта используются для распознавания жестов?
A. UIGesturePicker
B. UIGestureResolver
C. UIGestureRecognizer
D. UIGesturesManager

56. Какой жест обрабатывает UIPanGestureRecognizer?
A. Нажатие двумя пальцами на экран
B. Нажатие пальцем на экран
C. Сдвиг пальца по экрану
D. Поворот двух пальцев после нажатия

57. Какой gesture recognizer нужно использовать, чтобы реализовать уменьшение/увеличение фотографии двумя пальцами?
A. UISwipeGestureRecognizer
B. UIPanGestureRecognizer
C. UIRotationGestureRecognizer
D. UIPinchGestureRecognizer

58. На каком сайте мы задаем название приложения, иконку, скриншоты и другую подобную информацию?
A. developer.apple.com
B. itunesconnect.apple.com
C. appstore.apple.com
D. itunes.apple.com

59. Какой сервис используется для распространения приложений для тестировщиков до публикации в App Store?
A. AppBot
B. AppAnnie
C. Flurry
D. TestFlight

60. Какой сервис используется для отслеживания количества скачиваний и покупок приложения?
    A. Flurry
    B. TestFlight
    C. AppAnnie
    D. AppBot
61. Какой сервис используется для отслеживания отзывов к приложению?
    A. AppAnnie
    B. AppBot
    C. TestFlight
    D. Flurry
62. Какой сервис используется для отслеживания поведения пользователей внутри вашего приложения?
    A. Flurry
    B. AppBot
    C. AppAnnie
    D. TestFlight
63. Какой сервис используется для отслеживания падений вашего приложения после его публикации в App Store?
    A. iTunes Connect
    B. AppBot
    C. AppAnnie
    D. Crashlytics
64. Что такое in-app purchase
    A. Платное приложение
    B. Покупка внутри приложениz
    C. Плата за отключение рекламы
    D. Реклама внутри приложения
65. Для чего используется библиотека ShareKit?
    A. Отправка сообщений в социальные сети
    B. Локализация приложений
    C. Обработка in-app purchases
    D. Отслеживание ошибок в приложении
66. Какие существуют два способа для автоматического изменения интерфейса при изменении размера экрана на iOS?
    A. autofixing и autoswitching
    B. autoresizing и autorebooting
    C. layouting и resizing
    D. autosizing и autolayout
67. Какой сервис используется для распространения приложений для тестировщиков до публикации в App Store?
    A. AppBot
    B. AppAnnie
    C. Flurry
    D. TestFlight
68. Какой сервис используется для отслеживания количества скачиваний и покупок

приложения?

A. Flurry
B. TestFlight
C. AppAnnie
D. AppBot

Лабораторные занятия № 20-28

1. Консольное приложение "Сумма двух чисел"
2. Функции. Функциональные типы.
3. Замыкания. Возможности замыканий.
4. Создание собственного типа данных. Перечисления.
5. Структуры. Свойства в структурах.
6. Классы. Наследования классов
7. Свойства. Сабскрипты. Протоколы.
8. Знакомство с основами SwiftIU. Оптимизация мобильного приложения. Паттерны проектирования.

Практическое задание № 1

Необходимо создать приложение Заметки на языках программирования, который вы больше предпочитаете. Приложение может быть сделано как в консольном виде, так и в браузере или в виде приложения на Android и IOS.

Строгих требований к реализации функциональности нет, будьте творческими!

Обязательные требования:

1. Создание одной простейшей заметки только с текстом
2. Редактирование заметки в окне собственного приложения
3. Сохранение заметки между сеансами приложения, в любом формате
4. При первом запуске, приложение должно иметь одну заметку с текстом

Желательно:

1. Создание нескольких заметок в приложении
2. Выводить список существующих заметок
3. Возможность редактирования любой заметки из списка
4. Также сохранять все заметки между сеансами

Идеи для улучшения

1. Возможность выделять текст курсивом, жирным и т. д.
2. Менять шрифт и размер текста
3. Вставка картинок

Практическое задание № 2

Разработать и протестировать приложение для iPhone с использованием среды разработки XCode. Приложение для iPhone, представляющее из себя мини игру.

Загадывающий игрок загадывает любое целочисленное число. Отгадывающий должен за наименьшее количество ходов отдать число. Загадывающий может отвечать только «больше», «меньше» или «равно».

Приложение может состоять из нескольких экранов:

- Экран старта игры
- Экран загадывания числа
- Экран отгадывания числа компьютером
- Экран отгадывания числа игроком
- Экран с результатом игры

Победителем считается игрок, угадавший число за наименьшее количество шагов.

Технические требования к приложению:

- Для реализации интерфейса пользователя необходимо использовать стандартные элементы UIKit;
- Приложение должно поддерживать, как горизонтальную, так и вертикальную ориентацию экрана. При автоповороте UI должен обновляться;
- Открытие экрана игрового процесса должно осуществляться модально;
- Необходимо использование одного из архитектурных подходов: MVP, MVVM, VIPER с обоснованием выбора. Обоснование прикрепить в Readme проекта;
- Всю разработку необходимо вести с применением системы контроля версий GIT. Для размещения выбрать площадку на свое усмотрение: GitHub, GitLab, BitBucker. Макет экранов ниже:

Практическое задание № 3

Разработать и протестировать демо-приложение для iPhone с использованием среды разработки XCode. Реализовать демо-приложение для поиска фильмов и отображению детальной информации об отдельном фильме.

Приложение может состоять из нескольких экранов:

- Главный экран, который должен содержать в себе следующие элементы:
  - Строку поиска.
  - Ячейка результатов поиска. Ячейка должна содержать постер фильма, название, год выхода и краткое описание.
  - Блок «Популярные». Данный блок – это отдельный MVVM модуль, представляющий из себя горизонтальный список с постерами популярных фильмов.
  - Список результатов поиска. Должен поддерживать постраничную погрузку.
- Отдельный экран подробной информации.
  - После тапа на постер популярного фильма или на ячейку результата поиска, открывается экран подробной информации о фильме. На нем должен отображаться постер фильма, название, подробное описание, год выхода и рейтинг фильма.

Технические требования к приложению:

- Язык Swift.
- Архитектура – MVVM
- Обязательно использование FRP фреймворка RxSwift
- Routing должен быть выведен в отдельный слой
- Подготовка моделей и мью-моделей должна происходить в фоновом потоке
- Обработка ошибок и отображение их пользователю (в виде обычного UIAlertView)
- Мониторинг интернет-соединения и повторение запроса в случае, если

соединение с интернетом восстановлено;

- View-компонент в модуле MVVM может быть как UIView, так и UIViewController.
- Макет экрана подробной информации – составляется по вашему усмотрению. Макет главного экрана представлен ниже:



Практическое задание № 4

Разработать и протестировать приложение для iPhone с использованием среды разработки XCode. Приложение для просмотра фотографий.

Приложение может состоять из нескольких экранов:

- Главный экран. Представляет собой бесконечную ленту-галерею фотографий следующим образом: каждый элемент – фотография + мета-информация об этой фотографии.
- Экран просмотра фотографии. Открывается при клике по фотографии. Экран содержит фотографию в полном размере и дату её загрузки.

Технические требования к приложению:

- Источник картинок – любое публичное API:
- Реализация оффлайн-режима: сохранение максимальной функциональности без сети (или в случае её нестабильной работы);
- Для реализации интерфейса пользователя необходимо использовать стандартные элементы UIKit;
- Приложение должно поддерживать, как горизонтальную, так и вертикальную ориентацию экрана. При автоповороте UI должен обновляться;
- Необходимо использование одного из архитектурных подходов: MVP, MVVM, VIPER с обоснованием выбора. Обоснование прикрепить в Readme проекта;
- Всю разработку необходимо вести с применением системы контроля версий GIT. Для размещения выбрать площадку на свое усмотрение: GitHub, GitLab, BitBucker.

Практическое задание № 5

Разработать и протестировать приложение для iPhone с использованием среды разработки XCode. Приложение для просмотра и публикации фотографий.

Приложение состоит из двух экранов:

- Гланый экран – представляет собой таббар с двумя вкладками:

- На первой вкладке – коллекция случайных фотографий с Unsplash. Вверху строка поиска по фотографиям с Unsplash. При нажатии на ячейку пользователь попадает на экран подробной информации.
- На второй вкладке – таблица со списком любимых фотографий, в ячейке миниатюра фотографии и имя автора. При нажатии на ячейку – переход на экран подробной информации.

– Экран подробной информации. Содержит в себе фотографию, имя автора, дату создания, местоположение и количество скачиваний. Также содержит кнопку, нажатие на которую может добавить фотографию в список любимых фотографий и удалить из него. Можно сделать этот список редактируемым (желательно).

Технические требование к мобильному приложению:

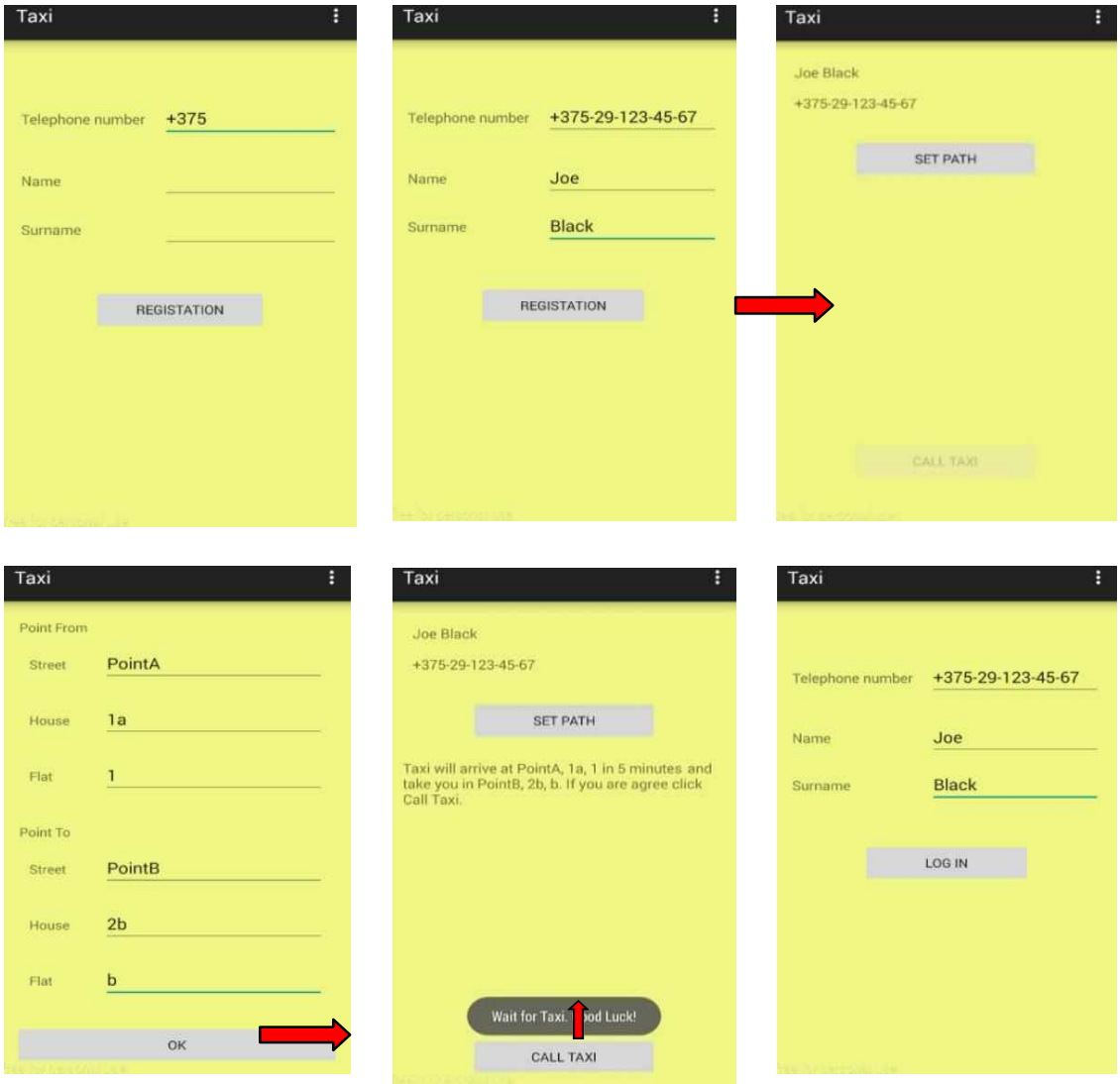
– Вёрстка приложения должна быть сделана кодом, без использования Storyboard;
– Вёрстка должна производиться только через Auto Layot;
– Использование SwiftUI – запрещено;
– Необходимо использовать как минимум одну стороннюю библиотеку Cocoa Pods (кроме Usplash Photo Picker for iOS и пр.);
– Необходимо использование как минимум одного UIAlertView;
– Ссылка на API – https://unsplash.com/documentation.

Самостоятельная работа обучающихся – Разработать мобильное приложение «Taxi»

1. Разработать мобильное приложение Taxi, состоящее из трех Activity (рисунок).
2. В первом Activity создать три редактируемых текстовых поля (EditText) для ввода пользователем регистрационных данных (телефон, имя и фамилия) и кнопку Registration для запуска второго Activity.
3. При нажатии кнопки Registration выполнить явный вызов второго Activity с передачей данных о пользователе (телефон, имя и фамилия).
4. Во втором Activity создать два текстовых поля (TextView) для вывода переданной информации о пользователе (имя и 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).
5. При нажатии кнопки Set path выполнить неявный вызов третьего Activity с помощью метода startActivityForResult.
6. В третьем Activity создать шесть редактируемых текстовых полей (EditText) для ввода параметров маршрута движения, кнопку ОК для возврата во второе Activity.
7. При нажатии кнопки ОК реализовать возврат во второе Activity с передачей в качестве результата параметров маршрута движения.
8. После возврата во второе Activity в текстовое поле вывести информацию о маршруте движения и предложение вызвать такси, кнопку вызова такси Call taxi сделать доступной.
9. При нажатии кнопки Call Taxi вывести всплывающее сообщение об успешной отправке такси.
10. 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.

11. Вывести в лог очередность вызовов методов жизненного цикла первого, торого и третьего Activity.
12. Сделать вывод на основании логов о жизненном цикле Activity
13. Продемонстрировать работу приложения Taxi на эмуляторе или реальном устройстве.
14. Дополнительное задание, предполагающее самостоятельное углубленное освоение материала: реализовать функционал geo-локации для автоматического определения исходной точки маршрута движения и/или Google-карт для обозначения конечной точки маршрута движения пользователя.



а- исходное Activity; *б* – исходное Activity с введенными данными; *в* – второеActivity с выводом данных; *г* – третье Activity для ввода маршрута движения; *д* – возврат во второе Activity; *е* – исходное Activity при повторных запусках

Рисунок – Пример работы приложения «Taxi»

5.2. Критерии оценивания

5.2.1. Критерии оценивания устного ответа

При оценке устного ответа, обучающегося учитывается:

1) полнота и правильность ответа;
2) степень осознанности, понимания изученного;

Отметка «5»: ответ правильный, полный в соответствии с изученным материалом; материал изложен в определенной логическойпоследовательности, литературным языком.

Отметка «4»: ответ правильный, полный в соответствии с изученным материалом; материал изложен в определенной логической последовательности; возможны отдельные затруднения в формулировке выводов.

Отметка «3»: ответ, в котором в основном правильно, но схематично или с отклонениями от последовательности изложения раскрыт материал или неполный, несвязный ответ, изложенный нелогично

Отметка «2»: при ответе обнаружено непонимание обучающимся основного содержания учебного материала, неумение его анализировать допущены существенные ошибки, которые обучающийся не смог исправить при наводящих вопросах преподавателя, отсутствует логика в изложении материала, нет необходимых обобщений и самостоятельной оценки фактов; недостаточно сформированы навыки устной речи.

5.2.2. Критерии оценивания выполнения заданий на лабораторных и практических занятиях

- **Отметка «5»**: работа выполнена полностью и правильно; сделаны правильные выводы.
- **Отметка «4»**: работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя.
- **Отметка «3»**: работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки.
- **Отметка «2»**: допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя.

Задания к лабораторным занятиям представлены в методических указаниях к лабораторным занятиям по МДК.02.07 Разработка приложений для мобильных платформ

5.2.3. Критерии оценивания тестовых заданий

Оценка в баллах	Степень выполнения задания
Неуд.	Выполнено от 0 до 49,9 % предложенных заданий
Удов.	Выполнено от 50 до 69,9% предложенных заданий
Хор.	Выполнено от 70 до 89,9% предложенных заданий
Отл.	Выполнено от 90 до 100% предложенных заданий

5.2.4. Общая классификация ошибок

При оценке знаний и умений учитываются ошибки и недочёты в работе.

Грубыми считаются ошибки:

- незнание определения основных понятий, законов, общепринятых символов обозначений величин;
- неумение выделить в ответе главное; обобщить результаты изучения;
- неумение применить знания для решения задач;
- неумение использовать полученные данные для выводов;
- неумение пользоваться первоисточниками, учебником, справочником;
- нарушение техники безопасности, небрежное отношение к оборудованию

Негрубыми считаются ошибки:

- неточность формулировок, определений, понятий, законов, вызванная неполнотой охвата основных признаков определяемого понятия или заменой 1-3 из этих признаков второстепенными;
- ошибки, вызванные несоблюдением условий проведения лабораторных занятий;
- недостаточно продуманный план устного ответа (нарушение логики изложения, подмена отдельных основных вопросов второстепенными);
- нерациональные методы работы со справочной литературой;
- неумение выполнять лабораторные задания в общем виде.

6. ОЦЕНОЧНЫЕ СРЕДСТВА ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО МЕЖДИСЦИПЛИНАРНОМУ КУРСУ

МДК.02.07 Разработка приложений для мобильных платформ

Промежуточная аттестация по междисциплинарному курсу МДК.02.07 Разработка приложений для мобильных платформ проводится в форме экзамена.

6.1. Вопросы к экзамену по МДК.02.07 за I семестр

1. Основные платформы мобильных приложений
2. Сравнительный анализ мобильных платформ ОС Android, iOS
3. Нативные приложения, веб-приложения, их области применения
4. Гибридные и кроссплатформенные приложения, их области применения
5. Основные языки для разработки мобильных приложений (Kotlin)
6. Понятие мобильной операционной системы. Архитектура Android (Уровень ядра; уровень библиотек и среды выполнения)
7. Понятие мобильной операционной системы. Архитектура Android (уровень фреймворка приложений, уровень приложений)
8. Среда выполнения и система безопасности
9. Структура типичного мобильного приложения (Основные компоненты Activity)
10. Структура типичного мобильного приложения (Service, Broadcast Receiver и Content Provider)
11. Структура проекта Android Studio (Корневая папка – App, Рабочая область для XML-файлов в Android Studio)
12. Структура проекта Android Studio (Параметры разметки, единицы измерения**)**
13. Элементы управления и контейнеры ViewGroup (Контейнер LinearLayout)
14. Элементы управления и контейнеры ViewGroup (Контейнер TableLayout)
15. Элементы управления и контейнеры ViewGroup (Контейнер ReiativeLayout)
16. Элементы управления и контейнеры ViewGroup (Контейнер ConstraintLayout)
17. Элементы управления (TextView)
18. Элементы управления (Button). Обработка событий (нажатий) на кнопку
19. Всплывающие окна. Toast
20. Взаимодействие Activity. Понятие Intent

6.2. Вопросы к экзамену по МДК.02.07 за II семестр

1. Операционная система. Достоинства и недостатки платформы Mac OS. Сравнение iOS с другими платформами. Достоинства и недостатки iOS.
2. Уровни iOS-архитектуры. Библиотеки и пакеты.
3. MVC-архитектура. Основные компоненты архитектуры. Алгоритм работы архитектуры.
4. Основы Swift.
5. IDE (интегрированная среда разработки) для iOS.
6. Контейнерные типы данных. Кортеж. Литерал кортежа. Тип данных кортежа. Взаимодействие с элементами кортежа.
7. Контейнерные типы данных. Кортеж. Доступ к элементам кортежа. Редактирование кортежа. Сравнение кортежей.
8. Контейнерные типы данных. Диапазоны. Способы создания диапазонов. Базовые свойства и методы диапазонов
9. Контейнерные типы данных. Массивы. Способы создания массива. Доступ к элементам массива. Тип данных массива.
10. Контейнерные типы данных. Массивы. Массивы как Value Type (значимый тип). Отличие значимого типа от ссылочного (reference type).
11. Контейнерные типы данных. Массивы. Пустой и многомерный массивы. Операции с массивами.
12. Контейнерные типы данных. Массивы. Базовые свойства и методы массивов. Срезы массивов (ArraySlice).
13. Контейнерные типы данных. Множества. Отличие множеств от массивов. Синтаксис множеств. Пустое множество.
14. Контейнерные типы данных. Множества. Базовые свойства и методы множеств. Операции со множествами.
15. Контейнерные типы данных. Словари. Способы создания словаря. Тип данных словаря. Взаимодействие с элементами словаря.
16. Контейнерные типы данных. Словари. Пустой словарь. Базовые свойства и методы словарей.
17. Типы данных. Предназначение типов данных. Числовые типы данных. Операции с числами. Преобразование числовых данных.
18. Типы данных. Строковые типы данных. Тип данных Character и String.
19. Типы данных. Пустые и многострочные строковые литералы. Операции со строками. Приведение к строковому типу данных.
20. Типы данных. Типы данных. Способы записи числовых значений в популярных системах счисления.
21. Операторы управления. Виды операторов управления. Глобальная функция утверждений.
22. Операторы управления. Виды операторов управления. Оператор условия. Сокращенный и стандартный синтаксис оператора условия.
23. Операторы управления. Виды операторов управления. Расширенный синтаксис оператора условия. Тернарный оператор условия.
24. Операторы управления. Виды операторов управления. Оператор ветвления. Дополнительные операторы в операторе ветвления.

25. Операторы управления. Виды операторов управления. Операторы повторения с предварительной проверкой условия и постпроверкой условия. Использование дополнительных операторов в операторах повторения.
26. Операторы управления. Виды операторов управления. Оператор повторения для элементов последовательности.
27. Функции. Характеристика функций. Синтаксис и алгоритм выполнения функции. Сквозные параметры функции.
28. Функции. Функциональные типы. Возможности функций.
29. Замыкания. Виды замыканий. Замыкающие выражения. Метод sorted.
30. Замыкания. Возможности замыкания. Замыкания как ссылочный тип.
31. Замыкания. Автозамыкания. Реализация автозамыканий.
32. Опциональные типы данных. Виды извлечения опционального значения. Проверка наличия значения в опционале. Опциональное связывание.
33. Основные объекты ООП и ПОП. Основные принципы ООП и ПОП.
34. Перечисления. Синтаксис, применение и алгоритм перечислений. Присвоение значения перечисления параметру. Свойства перечислений.
35. Перечисления. Синтаксис и алгоритм перечислений. Инициализатор в перечислениях. Свойства и методы в перечислениях.
36. Структуры. Синтаксис структур. Свойства в структурах. Методы в структурах.
37. Классы. Свойства классов. Общие черты структуры и класса.
38. Классы. Наследование. Синтаксис наследования.
39. Классы. Переопределение наследуемых типов. Правила переопределения элементов.
40. Классы. Приведение типов. Операции при помощи приведения типов

6.3. Критерии оценивания ответов на экзамене

– оценка **«отлично»**, если студент обладает глубокими и прочными знаниями программного материала; при ответе на вопросы продемонстрировал исчерпывающее, последовательное и логически стройное изложение; правильно сформулировал понятия и закономерности по вопросам; сделал вывод по излагаемому материалу;

– оценка **«хорошо»**, если студент обладает достаточно полным знанием программного материала; его ответ представляет грамотное изложение учебного материала; но имеются существенные неточности в формулировании понятий и закономерностей по вопросам; не полностью сделаны выводы по излагаемому материалу;

– оценка **«удовлетворительно»**, если студент имеет общие знания основного материала без усвоения некоторых существенных положений; формулирует основные понятия с некоторой неточностью; затрудняется в приведении примеров, подтверждающих теоретические положения;

– оценка **«неудовлетворительно»**, если студент не знает значительную часть программного материала; допустил существенные ошибки в процессе изложения; не умеет выделить главное и сделать вывод; приводит ошибочные определения; ни один вопрос не рассмотрен до конца, наводящие вопросы не помогают.

7. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ

Основные печатные издания

1. Федотенко М.А. Разработка мобильных приложений: учебник СПО. - М.: Образовательно-издательский центр «Академия», 2024 г
2. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности: учебное пособие / Г.Н. Федорова. — Москва: КУРС: ИНФРА-М, 2023. — 336 с. — (Среднее профессиональное образование). - ISBN 978-5-906818-41-6. - Текст: электронный. - URL: https://znanium.ru/catalog/product/1896457
3. Чистов. Д. В. Проектирование информационных систем: учебник и практикум для среднего профессионального образования / Д. В. Чистов, П. П. Мельников, А. В. Золотарюк, Н. Б. Ничепорук. — 2-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2024. — 293 с. — (Профессиональное образование). — ISBN 978-5-534-16217-2. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/538370

Дополнительные источники

1. 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: 13.12.2021).
2. Белугина С.В. Разработка программных модулей программного обеспечения для компьютерных систем, Прикладное программирование. – Санкт-Петербург: Лань, 2021 – 312 с.
3. Во Х., Оптимизация производительности приложений для iOS/Во Х. - Москва: ДМК Пресс, 2022. - 320 с.

Интернет-источники

- http://programm.ws/index.php - учебники по программированию
- https://metanit.com - сайт о программировании METANIT.COM
- https://android-developers.blogspot.ru/ - сайт о программировании
- https://android-arsenal.com/ - сайт о программировании