

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ярославский государственный университет им. П.Г. Демидова

Кафедра компьютерных сетей

УТВЕРЖДАЮ

Декан факультета ИВТ

 Д.Ю. Чалый

« 24 » мая 2022 г.

Рабочая программа дисциплины

«Основы мобильной разработки»

Направление подготовки

09.03.03 Прикладная информатика

Профиль

«Прикладная информатика в экономике»

Квалификация выпускника

Бакалавр

Форма обучения

очная

Программа рассмотрена
на заседании кафедры
от 22 марта 2022 г.,
протокол № 7

Программа одобрена НМК
факультета ИВТ
протокол № 6 от
18 апреля 2022 г.

Ярославль

1. Цели освоения дисциплины

Целью дисциплины «Основы мобильной разработки» являются изучение современных платформ для разработки мобильных приложений; изучение основных принципов построения пользовательских интерфейсов приложений для мобильных устройств; изучение основных принципов построения и особенностей современных инновационных мобильных сервисов; формирование представления о современном состоянии и проблемах построения мобильных сервисов; формирование способности к восприятию новых научных фактов и гипотез и использованию полученных знаний в процессе образования; формирование умения ориентироваться в методологических подходах и видеть их в контексте существующих парадигм разработки программного обеспечения и информационных технологий.

2. Место дисциплины в структуре ОП бакалавриата

Дисциплина «Основы мобильной разработки» относится к факультативной части ОП бакалавриата.

В курсе рассматриваются вопросы разработки приложений и сервисов для мобильных устройств с учётом их специфических особенностей. Содержание курса тесно связано фактически со всеми дисциплинами, которые изучались студентами. Освоению данной программы предшествуют учебные курсы по программированию и современным информационным технологиям.

Дисциплина «Основы мобильной разработки» обеспечивает закрепление и углубление теоретических знаний и практических навыков по основным дисциплинам ИТ-цикла. Дисциплина способствует профессиональному росту студентов, повышению их общеметодологического уровня, а также дальнейшему развитию навыков научно-исследовательской деятельности.

3. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОП бакалавриата

Процесс изучения дисциплины направлен на формирование следующих элементов компетенций в соответствии с ФГОС ВО, ОП ВО и приобретения следующих знаний, умений, навыков и (или) опыта деятельности:

Формируемая компетенция (код и формулировка)	Индикатор достижения компетенции (код и формулировка)	Перечень планируемых результатов обучения
Профессиональные компетенции		
ПК-2 Способность разрабатывать и адаптировать прикладное программное обеспечение	ПК-2.1. Знает современные методы разработки и реализации алгоритмов математических моделей на базе языков и пакетов прикладных программ моделирования. ПК-2.2. Умеет разрабатывать и реализовывать алгоритмы математических моделей на базе языков и пакетов прикладных программ моделирования. ПК-2.3. Имеет практический опыт разработки и реализации алгоритмов их на базе языков и пакетов прикладных программ моделирования	Знать: общие принципы построения приложений под мобильные платформы; Уметь: проектировать мобильные приложения в соответствии с требованиями пользователя и конкретных программных платформ. Владеть навыками: работы с инструментарием разработки приложений под различные мобильные платформы.

4. Объем, структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 2 зач. ед., 72 акад. час.

№ п/п	Темы (разделы) дисциплины, их содержание	Семестр	Виды учебных занятий, включая самостоятельную работу студентов, и их трудоемкость (в академических часах)						Формы текущего контроля успеваемости Форма промежуточной аттестации (по семестрам)
			Контактная работа						
			лекции	практические	лабораторные	консультации	аттестационные испытания	самостоятельная работа	
1.	Основы разработки приложений для ОС Android	8	2					5	
2.	Активности и интенды	8	2					5	
3.	Архитектура MVC в Android	8	2			1		5	Лабораторная работа
4.	Класс View и его возможности	8	2					5	Лабораторная работа
5.	Работа с ресурсами	8	2			1		6	
6.	Хранение данных	8	2					6	Лабораторная работа
7.	Пример приложения, использующего БД для хранения данных	8	2			1		5,9	
8.	Асинхронное выполнение	8	2					5,9	Лабораторная работа
9.	Провайдеры контента	8	2			1		5,9	
	Всего за 8 семестр		18			4		0,3	Зачет
	Всего		18			4		50	

Содержание разделов дисциплины:

1. Основы разработки приложений для ОС Android
 - 1.1. Android SDK
 - 1.2. Менеджер пакетов Android SDK
 - 1.3. Создание проекта
 - 1.4. Структура проекта

- 1.5. Файл манифеста
- 1.6. Сборка проекта
- 2. Активности и интен­ты
 - 2.1. Компоненты Android-приложения
 - 2.2. Интен­т
 - 2.3. Объявление активности в файле манифеста
 - 2.4. Жизненный цикл активности
 - 2.5. Вызов активности через интен­т
 - 2.6. Задачи и стек активностей
 - 2.7. Получение данных из интен­та
 - 2.8. Возврат результата из активности
- 3. Архитектура MVC в Android
 - 3.1. Архитектура «модель—вид—контроллер»
 - 3.2. Создание проекта
 - 3.3. Построение пользовательского интерфейса
 - 3.4. Загрузка пользовательского интерфейса из XML-файла и доступ к его компонентам
 - 3.5. Обработка событий элементов интерфейса пользователя
 - 3.6. Модель счётчика
 - 3.7. Встраивание модели в контроллер
 - 3.8. Активная модель
 - 3.9. Модификация класса активности для использования активной модели
 - 3.10. Преимущества и недостатки активной и пассивной модели
 - 3.11. Обработка смены ориентации экрана
- 4. Класс View и его возможности
 - 4.1. Назначение класса View
 - 4.2. События касания экрана
 - 4.3. События клавиатуры
 - 4.4. Правила обработки событий вдоль иерархии виджетов
 - 4.5. Рисование на виджетах
- 5. Работа с ресурсами
 - 5.1. Понятие ресурсов и их назначение
 - 5.2. Классификация ресурсов
 - 5.3. Использование ресурсов из приложения
 - 5.4. Ресурсы, зависящие от конфигурации
 - 5.5. Использование ресурсов для формирования меню и панели действий
 - 5.6. Обработка действий меню и панели задач
- 6. Хранение данных
 - 6.1. Способы хранения данных
 - 6.2. Механизм настроек
 - 6.3. Основные классы для работы СУБД SQLite
 - 6.4. Управление жизненным циклом БД
 - 6.5. Доступ к данным
 - 6.6. Работа с курсорами
- 7. Пример приложения, использующего БД для хранения данных
 - 7.1. Описание приложения
 - 7.2. Класс управления жизненным циклом БД
 - 7.3. Пользовательский интерфейс главной активности
 - 7.4. Инициализация главной активности
 - 7.5. Меню приложения и обработка добавления записи
 - 7.6. Пользовательский интерфейс активности редактора

- 7.7. Интерфейс взаимодействия активностей
- 7.8. Реализация активности редактора задач
- 7.9. Вызов активности редактора для изменения существующей задачи
- 7.10. Обработка результата вызова активности редактора в главной активности
- 8. Асинхронное выполнение
 - 8.1. Назначение механизмов асинхронного выполнения
 - 8.2. Класс Handler и очередь сообщений
 - 8.3. Пример использования класса Handler
 - 8.4. Класс AsyncTask
 - 8.5. Пример использования класса AsyncTask
- 9. Провайдеры контента
 - 9.1. Назначение провайдеров контента
 - 9.2. Пример стандартного провайдера контента
 - 9.3. Провайдер контента для списка задач
 - 9.4. Регистрация провайдера контента в файле манифеста
 - 9.5. Асинхронная загрузка данных, предоставляемых провайдером контента
 - 9.6. Вставка и обновление данных через провайдер контента

5. Образовательные технологии, используемые при осуществлении образовательного процесса по дисциплине

В процессе обучения используются следующие образовательные технологии:

Лекция-беседа или «диалог с аудиторией», является наиболее распространенной и сравнительно простой формой активного вовлечения студентов в учебный процесс. Эта лекция предполагает непосредственный контакт преподавателя с аудиторией. Преимущество лекции-беседы состоит в том, что она позволяет привлекать внимание студентов к наиболее важным вопросам темы, определять содержание и темп изложения учебного материала с учетом особенностей студентов.

Мастер-класс – это особая форма учебного занятия, когда преподаватель-мастер передает свой опыт путем прямого и комментированного показа последовательности действий, методов, приемов и форм педагогической деятельности. Целью проведения мастер-класса является профессиональное, интеллектуальное и эстетическое воспитание студентов, и прежде всего, развитие в ходе мастер-класса способности студента самостоятельно и нестандартно мыслить.

Лабораторная работа – организация учебной работы с реальными материальными и информационными объектами, экспериментальная работа с аналоговыми моделями реальных объектов.

6. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения и информационных справочных систем (при необходимости)

В процессе осуществления образовательного процесса используются: для разработки документов, презентаций, для работы с электронными таблицами

OfficeStd 2013 RUS OLP NL Acdmc 021-10232

LibreOffice (свободное)

издательская система LaTeX;

- Android Studio среда разработки мобильных приложений. Доступ свободный

– для поиска учебной литературы библиотеки ЯрГУ – Автоматизированная библиотечная информационная система "БУКИ-NEXT" (АБИС "Буки-Next").

7. Перечень основной и дополнительной учебной литературы, ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

а) основная:

1. Парамонов, И. В., Разработка мобильных приложений для платформы Android : учеб. пособие для вузов / И. В. Парамонов ; Яросл. гос. ун-т, Ярославль, ЯрГУ, 2013, 87с
2. Парамонов, И. В., Разработка мобильных приложений для платформы Android [Электронный ресурс] : учеб. пособие для вузов / И. В. Парамонов ; Яросл. гос. ун-т, Ярославль, ЯрГУ, 2013, 87с

б) дополнительная:

1. Соколова, В. В., Вычислительная техника и информационные технологии. Разработка мобильных приложений : учеб. пособие для прикладного бакалавриата / В. В. Соколова, М., Юрайт, 2016, 175с

в) ресурсы сети «Интернет»

- Сайт для разработчиков под Android <https://developer.android.com/index.html> Доступ свободный

- Android Studio <https://developer.android.com/studio/index.html> среда разработки мобильных приложений. Доступ свободный

8. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине включает в свой состав специальные помещения:

-учебные аудитории для проведения занятий лекционного типа и практических занятий (семинаров);

- учебные аудитории для проведения групповых и индивидуальных консультаций,

- учебные аудитории для проведения текущего контроля и промежуточной аттестации;

-помещения для самостоятельной работы;

-помещения для хранения и профилактического обслуживания технических средств обучения.

Специальные помещения укомплектованы средствами обучения, служащими для представления учебной информации большой аудитории.

Для проведения занятий лекционного типа предлагаются наборы демонстрационного оборудования и учебно-наглядных пособий, хранящиеся на электронных носителях и обеспечивающие тематические иллюстрации, соответствующие рабочим программам дисциплин.

Помещения для лабораторных занятий и самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду организации.

Число посадочных мест в лекционной аудитории больше либо равно списочному составу потока, а в аудитории для практических занятий (семинаров) – списочному составу группы обучающихся.

Автор(ы) :

Доцент кафедры КС _____

И. В. Парамонов

**Приложение №1 к рабочей программе дисциплины
«Основы мобильной разработки»
Фонд оценочных средств
для проведения текущей и промежуточной аттестации студентов
по дисциплине**

**1. Типовые контрольные задания или иные материалы, необходимые для оценки
знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы
формирования компетенций**

**1.1. Контрольные задания и иные материалы, используемые в процессе текущей
аттестации**

Задания для лабораторных работ

**Пример задания для лабораторной работы по теме «Архитектура MVC в
Android»:**

Разработайте приложение, которое позволяет пользователю отслеживать, сколько времени было потрачено на то или иное действие. На главном экране приложения пользователь может запустить один из существующих таймеров. После старта таймера на главном окне показывается: сколько времени прошло после старта таймера. После нажатия на кнопку «остановить текущий таймер» пользователю предлагается ввести краткую информацию о проделанных действиях. Таймер останавливается после сохранения введённой информации. Одновременно работать может только один из счётчиков времени.

Пользователь может создавать новые таймеры, изменять существующие, просматривать журнал потраченного времени и общее потраченное время по каждому из счётчиков.

Приложение должно быть спроектировано в соответствии с архитектурой «модель—вид—контроллер».

Пример задания для лабораторной работы по теме «Хранение данных»:

Разработайте приложение, которое позволяет вести личный дневник с заметками. На главном окне приложения показывается лента всех записей, созданных человеком. Записи отсортированы по времени последнего изменения. Для каждой записи, помимо текста, назначается набор тэгов. Приложение должно позволять создавать, удалять и редактировать записи. Записи можно фильтровать по тегам. Если некоторым тэгом не помечена ни одна запись, то он не показывается в окне фильтра.

Приложение должно быть спроектировано в соответствии с архитектурой «модель—вид—контроллер».

**Пример задания для лабораторной работы по теме «Класс View и его
возможности»:**

Разработайте приложение, которое позволяет пользователю играть с телефоном в игру Ним, в которой два игрока по очереди берут камни, разложенные на несколько кучек. За один ход может быть взято любое количество камней из одной кучки от 1 до некоторого заданного заранее числа n . Выигрывает игрок, взявший последний камень. Количество кучек и максимальное количество камней, которое можно взять из кучки, должно задаваться в настройках приложения.

Приложение должно быть спроектировано в соответствии с архитектурой «модель—вид—контроллер».

Пример задания для лабораторной работы по теме «Асинхронное выполнение»:

Разработайте приложение, которое позволяет двум пользователям играть в шашки по сети. Приложение должно быть спроектировано в соответствии с архитектурой «модель—вид—контроллер».

Критерии оценивания лабораторных работ.

Критерий	Баллы
Проект правильно организован. Приложение успешно запускается.	10
Функциональность приложения соответствует техническому заданию.	10
Исходный текст приложения должным образом прокомментирован.	10
Студент уверенно объясняет устройство приложения и принципы его организации.	10
Приложение имеет правильную архитектуру.	5
Приложение правильно обрабатывает сохранение и восстановление состояния при закрытии и последующем открытии, а также при смене конфигурации.	5
Пользовательский интерфейс приложения верно функционирует при разных разрешениях и ориентациях экрана.	5
Приложение сопровождается модульными и/или системными тестами, которые успешно выполняются.	5

В случае, когда выполненная работа соответствует критерию не полностью, баллы по этому критерию снижаются пропорционально степени соответствия.

Перевод баллов в оценки

Сумма баллов	Оценка	Степень сформированности ПК-2
0-29	Неудовлетворительно	не сформирована
30-39	Удовлетворительно	пороговый уровень
40-49	Хорошо	продвинутый уровень
50-60	Отлично	высокий уровень

1.2 Список вопросов и (или) заданий для проведения промежуточной аттестации

Список вопросов к зачету:

- Что такое Android SDK? Какие компоненты он содержит? Какие инструментальные средства можно использовать при разработке приложений на платформе Android?
- Что такое менеджер пакетов Android? Какие задачи он решает?
- Какова структура автоматически создаваемого проекта приложения для Android? Какие компоненты создаются и в каких каталогах они размещаются?
- Что такое файл манифеста? Какова его структура? Какие основные элементы могут встречаться в файле манифеста и для чего они нужны?
- Что такое ant? Как он используется для сборки приложений? Какие цели содержатся в автоматически сгенерированном файле сборки?
- Из каких компонентов могут состоять Android-приложения?
- В чём назначение и характерные особенности каждого из компонентов?
- Что такое интент? Какую роль играют интенты во взаимодействии компонентов на платформе Android?
- Как объявить активность в файле манифеста? С какой целью необходимо данное объявление?
- Что такое жизненный цикл активности? Какие особенности платформы заставляют вводить понятие жизненного цикла?
- Какие callback-методы жизненного цикла активности могут быть переопределены разработчиком? В какие моменты жизненного цикла они будут вызываться? Каково типичное назначение каждого из этих callback-методов?
- Назовите два способа вызова активности через интент. В чём заключается различие между ними? Когда используется каждый из этих способов?
- Что такое «задача» в терминах Android? Как задачи связаны с активностями? Как задачи выглядят с точки зрения пользователя?
- Как получить данные, переданные из одной активности в другую? Как получить результат вызова активности?
- Что такое архитектурный шаблон MVC? Из каких компонентов состоят системы, основанные на данном шаблоне?
- Опишите допустимые способы связи и взаимодействия компонентов в рамках MVC. Обоснуйте, почему именно эти способы взаимодействия разрешены, а другие — нет.
- Охарактеризуйте способ построения пользовательского интерфейса, применяемый в Android-приложениях.
- Опишите, каким образом можно загрузить описание пользовательского интерфейса из кода и как можно получить доступ к отдельным виджетам.
- Опишите два способа обработки событий в Android. Укажите достоинства и недостатки каждого из них.
- Определите активную и пассивную модели в терминах архитектурного шаблона MVC. Осветите достоинства и недостатки каждого из типов моделей.
- Опишите, как правильно обрабатывать событие поворота экрана пользователем. Что происходит при повороте с точки зрения жизненного цикла активности?
- Создайте проект приложения, описанный в этой главе. Скомпилируйте приложение и запустите его в эмуляторе или на реальном устройстве.
- Каково назначение класса View? В каких случаях требуется создавать subclasses этого класса?
- Опишите, что необходимо сделать, чтобы обработать события касания экрана устройства пользователем.

- Опишите, что необходимо сделать, чтобы обработать события от клавиатуры Android-устройства.
- Сформулируйте правило обработки событий вдоль иерархии виджетов. С какой целью это правило установлено?
- Назовите основные классы, позволяющие выполнять рисование на произвольных виджетах. Осветите возможности этих классов. Проиллюстрируйте эти возможности примерами.
- Что такое ресурсы? Для решения каких задач разработан данный механизм в Android? Какие преимущества даёт разработчику использование механизма ресурсов?
- Какие типы ресурсов существуют? Как размещены в проекте файлы ресурсов?
- Как можно использовать ресурсы в приложении непосредственно из программного кода, а также из других ресурсов?
- Что такое ресурсы, зависящие от конфигурации? Для чего предназначен данный механизм и как его можно использовать?
- Как, используя механизм ресурсов, создать главное меню или панель действий Android-приложения?
- В чём отличия в реализации меню для ранних и поздних версий платформы Android?
- Как обработать выбор действий из главного меню или панели действий?
- Перечислите способы постоянного хранения данных на платформе Android. Объясните, в каких случаях разумно применять каждый из них.
- Что такое механизм настроек? Для чего он предназначен? Как его применять?
- Перечислите основные классы Android, предназначенные для работы с базой данных SQLite. На примерах объясните, как их применять.
- Что такое жизненный цикл базы данных? Какие средства платформы Android позволяют управлять этим жизненным циклом?
- Назовите методы класса SQLiteDatabase, предназначенные для работы с данными. На примерах объясните, как их можно использовать.
- В чём отличие между методами query() и rawQuery() класса SQLiteDatabase? В каких случаях применяется каждый из них?
- Что такое курсор набора данных? Для чего он предназначен? На примерах объясните, как использовать курсоры.
- Что такое асинхронное выполнение? В каких случаях оно используется? Какие задачи решает?
- Перечислите средства асинхронного выполнения, предоставляемые Android API.
- Что такое очередь сообщений? Какую функцию выполняет класс Handler и как его правильно использовать?
- Для чего предназначен класс AsyncTask? Как его использовать?
- Объясните, что такое синхронизация потоков. В каких случаях она необходима? Как средства платформы Android помогают решать задачу синхронизации?
- Что такое провайдер контента? Какую роль играют провайдеры контента в инфраструктуре Android?
- Для чего нужен URI при использовании провайдера контента?
- Из каких частей он состоит? По каким правилам формируется этот URI?
- Какие методы класса ContentProvider необходимо переопределять при реализации провайдера контента?
- Каково предназначение класса CursorLoader? Какие преимущества несёт в себе использование класса CursorLoader по сравнению с непосредственным выполнением запросов к провайдеру контента на главном потоке приложения?
- Для чего предназначены callback-методы интерфейса LoaderManager.LoaderCallbacks? Приведите пример их использования.
- Создайте проект приложения, описанный в этой главе. Скомпилируйте приложение и запустите его в эмуляторе или на реальном устройстве.

Пример теста на зачете:

1. Android SDK Tools содержит следующие компоненты:

- компилятор языка Java
- **библиотеки, предоставляющие API платформы Android**
- система сборки приложений Gradle
- **менеджер виртуальных устройств**

2. Gradle wrapper

1. **предназначен для автоматической установки версии gradle, необходимой для сборки проекта**
2. позволяет использовать gradle для сборки проектов в интегрированных средах разработки
3. **представляет собой shell-скрипт или командный файл (.bat)**
4. должен выполняться только на устройствах с операционной системой Android

3. Адаптеры списков предназначены для

1. **предоставления данных виджетам-спискам**
2. **создания экземпляров класса View по требованию виджетов-списков**
3. хранения данных, которые отображаются виджетами-списками
4. отображения данных в форме списков

4. Архитектурно правильный способ взаимодействия активности и фрагмента предполагает, что

- **активность непосредственно вызывает методы фрагмента**
- фрагмент непосредственно вызывает методы активности
- **фрагмент вызывает методы активности через callback-интерфейс**
- активность и фрагмент не вызывают методы друг друга
- активность взаимодействует с фрагментом через сервисы или провайдеры

контента

5. В Android интент предназначен для

- **передачи данных из вызывающей активности в вызываемую**
- **передачи данных из вызываемой активности в вызывающую**
- обработки запросов пользователя
- регистрации активностей в файле манифеста
- обращения к ресурсам приложения

6. Выберите верные утверждения, определяющие правила обработки событий классами-видами (виджетами) в Android

3. **в первую очередь событие передаётся наиболее вложенному виджету, ответственному за область, в которой произошло событие**
4. **если обработчик события возвращает true, то считается, что виджет обработал событие, и дальнейшая обработка не требуется**
5. если обработчик события возвращает false, то считается, что виджет обработал событие, и дальнейшая обработка не требуется
6. если виджет не определяет собственного обработчика для произошедшего события, то такое событие не обрабатывается никаким виджетом

7. Выберите методы класса SQLiteDatabase

`insert()`

`query()`

`moveToNext()`

`getWritableDatabase()`

`onUpgrade()`

8. Для взаимодействия с провайдером контента клиенты должны специфицировать

- тип структуры данных, в которую надо записать результат обращения

- **URI ресурса, к которому производится обращение**
- **действие, выполняемое над ресурсом**
- имя базы данных, к которой выполняется обращение

9. Для восстановления сохранённого при повороте состояния активности используются методы:

- `onInstanceState()`
- `onConfigurationChanged()`
- **`onCreate()`**
- **`onRestoreInstanceState()`**

10. Для обеспечения передачи произвольного объекта из активности в активность соответствующий класс должен реализовывать интерфейс

- `Bundle`
- `Intent`
- `ContentValues`
- **`Parcelable`**

11. Для обращения к ресурсам приложения из программного кода используются

- строковые идентификаторы
- файлы в формате XML
- **числовые идентификаторы, определённые в классе `R`**
- стандартные константы, определённые в классе `Activity`

12. Для рисования на виджетах используются:

- методы класса `DC`
- **метод `onDraw()` класса-виджета**
- метод `draw()` класса-виджета
- **методы класса `Canvas`**
- методы класса `Activity`

13. Для установки макета пользовательского интерфейса активность вызывает метод

- `onCreate()`
- **`setContentView()`**
- `setOnClickListener()`
- `findViewById()`
- `onView()`

14. Если номер версии Android на устройстве меньше, чем номер версии, указанный в параметре `targetSdkVersion`, но больше, чем номер версии, указанный в параметре `minSdkVersion` в файле манифеста приложения, то

- при попытке установки приложения на устройство возникнет ошибка
- **приложение успешно установится и будет работать корректно, если в нём не используются вызовы API, с номером версии больше, чем на устройстве**
- приложение успешно установится и будет работать корректно
- приложение успешно установится, но будет работать некорректно

15. Класс `AsyncTask`

- использует очередь сообщений для организации фонового выполнения
- не предусматривает создания классов-наследников для организации фонового выполнения
- **предоставляет средства передачи сообщений о прогрессе на UI-поток выполнения**
- **осуществляет передачу результата фонового выполнения на UI-поток выполнения**

16. Типичный проект Android-приложения содержит:

- файл конфигурации компилятора языка Java
- **файл манифеста**
- файлы с исходными текстами на языке C\#
- **каталог с ресурсами**

Правильные ответы выделены жирным.

Критерии оценивания:

За каждый выбранный правильный ответ начисляется количество баллов равное единице, делённое на общее количество правильных ответов на вопрос. За каждый выбранный неправильный ответ у студента снимается количество баллов, равное 0.5 балла. Если суммарный балл студента за вопрос оказывается отрицательным, вместо него засчитывается 0.

Баллы	Оценка	Степень сформированности ПК-2
0-4	Неудовлетворительно	не сформирована
5-8	Удовлетворительно	пороговый уровень
9-12	Хорошо	продвинутый уровень
13-16	Отлично	высокий уровень

Методические указания по выставлению зачета

Зачет выставляется по результатам выполнения всех лабораторных работ и зачетного теста на оценку не ниже удовлетворительно.

2. Перечень компетенций, этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкалы оценивания

2.1. Шкала оценивания сформированности компетенций и ее описание

Оценивание уровня сформированности компетенций в процессе освоения дисциплины осуществляется по следующей трехуровневой шкале:

Пороговый уровень - предполагает отражение тех ожидаемых результатов, которые определяют минимальный набор знаний и (или) умений и (или) навыков, полученных студентом в результате освоения дисциплины. Пороговый уровень является обязательным уровнем для студента к моменту завершения им освоения данной дисциплины.

Продвинутый уровень - предполагает способность студента использовать знания, умения, навыки и (или) опыт деятельности, полученные при освоении дисциплины, для решения профессиональных задач. Продвинутый уровень превосходит пороговый уровень по нескольким существенным признакам.

Высокий уровень - предполагает способность студента использовать потенциал интегрированных знаний, умений, навыков и (или) опыта деятельности, полученных при освоении дисциплины, для творческого решения профессиональных задач и самостоятельного поиска новых подходов в их решении путем комбинирования и использования известных способов решения применительно к конкретным условиям. Высокий уровень превосходит пороговый уровень по всем существенным признакам.

2.2. Перечень компетенций, этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования

Код компетенции	Форма контроля	Этапы формирования (№ темы (раздела))	Показатели оценивания	Шкала и критерии оценивания компетенций на различных этапах их формирования		
				Пороговый уровень	Продвинутый уровень	Высокий уровень
Профессиональные компетенции						
ПК-2	Лабораторная работа Защита проекта Зачет	1 – 9	Знать: <ul style="list-style-type: none">• общие принципы построения приложений под мобильные платформы; Уметь: <ul style="list-style-type: none">• проектировать мобильные приложения в соответствии с требованиями пользователя и конкретными программных платформ. Владеть навыками: <ul style="list-style-type: none">• работы с инструментарием разработки приложений под различные мобильные платформы.	Студент должен знать современные платформы для разработки мобильных приложений; основные принципы построения и особенности современных инновационных мобильных сервисов; уметь использовать на практике принципы построения пользовательских интерфейсов приложений для мобильных устройств; иметь представление о современном состоянии и проблемах построения мобильных сервисов; воспринимать новые научные факты и гипотез. Знать общие принципы	Студент должен знать современные платформы для разработки мобильных приложений; основные принципы построения и особенности современных инновационных мобильных сервисов; уметь использовать на практике принципы построения пользовательских интерфейсов приложений для мобильных устройств; иметь представление о современном состоянии и проблемах построения мобильных сервисов; воспринимать новые научные факты и гипотез и уметь ориентироваться в методологических подходах Знать общие принципы построения приложений под мобильные	Студент должен знать современные платформы для разработки мобильных приложений; основные принципы построения и особенности современных инновационных мобильных сервисов; уметь использовать на практике принципы построения пользовательских интерфейсов приложений для мобильных устройств; иметь представление о современном состоянии и проблемах построения мобильных сервисов; воспринимать новые научные факты и гипотез и использовать полученные знания в процессе образования; уметь ориентироваться в методологических подходах и видеть их в контексте существующих парадигм разработки программного

				<p>построения приложений под мобильные платформы;</p> <p>Уметь проектировать мобильные приложения в соответствии с требованиями пользователя</p> <p>Владеть навыками работы с инструментарием разработки приложений под различные мобильные платформы.</p> <p>Уметь представлять и защищать выполненный проект.</p>	<p>платформы;</p> <p>Уметь проектировать мобильные приложения в соответствии с требованиями пользователя</p> <p>Владеть навыками работы с инструментарием разработки приложений под различные мобильные платформы.</p> <p>Уверенно выполнять все этапы работы над самостоятельным проектом.</p> <p>Знать принципы работы систем контроля версий и использовать их в работе</p> <p>Уметь представлять и защищать выполненный проект.</p>	<p>обеспечения и информационных технологий.</p> <p>Знать принципы построения приложений под мобильные платформы;</p> <p>Уметь проектировать мобильные приложения в соответствии с требованиями пользователя и guidelines конкретных программных платформ.</p> <p>Владеть навыками работы с инструментарием разработки приложений под различные мобильные платформы.</p> <p>Уверенно выполнять все этапы работы над самостоятельным проектом.</p> <p>Знать принципы работы систем контроля версий и уверенно использовать их в работе</p> <p>Уметь представлять и защищать выполненный проект.</p>
--	--	--	--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3. Методические рекомендации преподавателю по процедуре оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Целью процедуры оценивания является определение степени овладения студентом ожидаемыми результатами обучения (знаниями, умениями, навыками и (или) опытом деятельности).

Процедура оценивания степени овладения студентом ожидаемыми результатами обучения осуществляется с помощью методических материалов, представленных в разделе «Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций»

3.1 Критерии оценивания степени овладения знаниями, умениями, навыками и (или) опытом деятельности, определяющие уровни сформированности компетенций

Пороговый уровень (общие характеристики):

- владение основным объемом знаний по программе дисциплины;
- знание основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы без существенных ошибок;
- владение инструментарием дисциплины, умение его использовать в решении стандартных (типовых) задач;
- способность самостоятельно применять типовые решения в рамках рабочей программы дисциплины;
- усвоение основной литературы, рекомендованной рабочей программой дисциплины;
- знание базовых теорий, концепций и направлений по изучаемой дисциплине;
- самостоятельная работа на практических и лабораторных занятиях, периодическое участие в групповых обсуждениях, достаточный уровень культуры исполнения заданий.

Продвинутый уровень (общие характеристики):

- достаточно полные и систематизированные знания в объёме программы дисциплины;
- использование основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать выводы;
- владение инструментарием дисциплины, умение его использовать в решении учебных и профессиональных задач;
- способность самостоятельно решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;
- умение ориентироваться в базовых теориях, концепциях и направлениях по изучаемой дисциплине и давать им сравнительную оценку;
- самостоятельная работа на практических и лабораторных занятиях, участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

Высокий уровень (общие характеристики):

- систематизированные, глубокие и полные знания по всем разделам дисциплины;

- точное использование терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;
- безупречное владение инструментарием дисциплины, умение его использовать в постановке и решении научных и профессиональных задач;
- способность самостоятельно и творчески решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- полное и глубокое усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;
- умение ориентироваться в основных теориях, концепциях и направлениях по изучаемой дисциплине и давать им критическую оценку;
- активная самостоятельная работа на практических и лабораторных занятиях, творческое участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

3.2 Описание процедуры выставления оценки

В зависимости от уровня сформированности каждой компетенции по окончании освоения дисциплины студенту выставляется оценка «зачтено», «незачтено».

Показатели и критерии, используемые при выставлении оценки подробно описаны в разделе «Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций».

Высокий уровень формирования компетенций соответствует оценке «отлично» за лабораторные работы и тест.

Продвинутый уровень формирования компетенций соответствует оценке «хорошо» за лабораторные работы и тест.

Пороговый уровень формирования компетенций соответствует оценке «удовлетворительно» за лабораторные работы и тест.

Оценка «отлично» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована на высоком уровне.

Оценка «хорошо» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на продвинутом уровне.

Оценка «удовлетворительно» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на пороговом уровне.

Оценка «неудовлетворительно» выставляется студенту, у которого хотя бы одна компетенция (полностью или частично формируемая данной дисциплиной) сформирована ниже, чем на пороговом уровне.

Оценка «зачет» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на пороговом уровне.

Оценка «незачтено» выставляется студенту, у которого хотя бы одна компетенция (полностью или частично формируемая данной дисциплиной) сформирована ниже, чем на пороговом уровне.

Приложение №2 к рабочей программе дисциплины «Основы мобильной разработки»

Методические указания для студентов по освоению дисциплины

Занятия проводятся в интерактивной форме с использованием мультимедиа-технологий. Занятия предполагают наличие дискуссий по поводу тех или иных вопросов разработки программных приложений осуществляемых в результате соответствующего предложения преподавателя. Применение полученных знаний отрабатывается и во время лабораторных занятий, ориентированных помимо закрепления лекционного материала на разбор различных модельных ситуаций.

Для успешного освоения дисциплины очень важно решение достаточно большого количества задач, как в аудитории, так и самостоятельно в качестве домашних заданий. Примеры решения задач разбираются на лекциях и лабораторных занятиях, при необходимости по наиболее трудным темам проводятся дополнительные консультации. Основная цель решения задач – помочь усвоить основы разработки приложений под мобильные платформы.

Для решения всех задач необходимо знать и понимать лекционный материал. Поэтому в процессе изучения дисциплины рекомендуется регулярное повторение пройденного лекционного материала. Материал, законспектированный на лекциях, необходимо дома еще раз прорабатывать и при необходимости дополнять информацией, полученной на консультациях, практических занятиях или из учебной литературы.

Большое внимание должно быть уделено выполнению домашней работы. В качестве заданий для самостоятельной работы дома студентам предлагаются задачи, аналогичные разобранным на лекциях и лабораторных занятиях или немного более сложные, которые являются результатом объединения нескольких базовых задач.

Для проверки и контроля усвоения теоретического материала, приобретенных практических навыков в течение обучения проводятся мероприятия текущей аттестации в виде лабораторных работ. Также проводятся консультации (при необходимости) по разбору заданий для самостоятельной работы, которые вызвали затруднения.

В конце семестра студенты сдают зачет.

Учебно-методическое обеспечение самостоятельной работы студентов по дисциплине

Для самостоятельной работы особенно рекомендуется использовать учебную литературу, указанную в разделе № 7 данной рабочей программы.

Также для подбора учебной литературы рекомендуется использовать широкий спектр интернет-ресурсов:

1. Электронно-библиотечная система «Университетская библиотека online» (www.biblioclub.ru) - электронная библиотека, обеспечивающая доступ к наиболее востребованным материалам-первоисточникам, учебной, научной и художественной литературе ведущих издательств (*регистрация в электронной библиотеке – только в сети университета. После регистрации работа с системой возможна с любой точки доступа в Internet.).

2. Для самостоятельного подбора литературы в библиотеке ЯрГУ рекомендуется использовать:

1. Личный кабинет (http://lib.uniyl.ac.ru/opac/bk_login.php) дает возможность получения on-line доступа к списку выданной в автоматизированном режиме литературы, просмотра и копирования электронных версий изданий сотрудников университета (учеб. и метод. пособия, тексты лекций и т.д.) Для работы в «Личном кабинете» необходимо зайти на сайт Научной библиотеки ЯрГУ с любой точки, имеющей доступ в Internet, в пункт меню «Электронный каталог»; пройти процедуру авторизации, выбрав вкладку «Авторизация», и заполнить представленные поля информации.

2. Электронная библиотека учебных материалов ЯрГУ

(http://www.lib.uniya.ac.ru/opac/bk_cat_find.php) содержит более 2500 полных текстов учебных и учебно-методических материалов по основным изучаемым дисциплинам, изданных в университете. Доступ в сети университета, либо по логину/пароллю.

3. Электронная картотека [«Книгообеспеченность»](#)

(http://www.lib.uniya.ac.ru/opac/bk_bookreq_find.php) раскрывает учебный фонд научной библиотеки ЯрГУ, предоставляет оперативную информацию о состоянии книгообеспеченности дисциплин основной и дополнительной литературой, а также цикла дисциплин и специальностей. Электронная картотека [«Книгообеспеченность»](#) доступна в сети университета и через Личный кабинет.