

Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Институт прикладной математики и компьютерных наук

УТВЕРЖДАЮ:

Директор



А. В. Замятин

20 22 г.

Рабочая программа дисциплины

Разработка приложений для мобильных платформ

по направлению подготовки

02.03.03 Математическое обеспечение и администрирование информационных систем

Направленность (профиль) подготовки :

DevOps-инженерия в администрировании инфраструктуры ИТ-разработки

Форма обучения

Очная

Квалификация

Бакалавр

Год приема

2022

Код дисциплины в учебном плане: Б1.04.14

СОГЛАСОВАНО:

Руководитель ОП

 С.П. Сущенко

Председатель УМК

 С.П. Сущенко

Томск – 2022

1. Цель и планируемые результаты освоения дисциплины

Целью освоения дисциплины является формирование следующих компетенций:

– ОПК-3 –Способен понимать и применять современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения;

– ПК-1 – Способен осуществлять программирование, тестирование и опытную эксплуатацию ИС с использованием технологических и функциональных стандартов, современных моделей и методов оценки качества и надежности программных средств.

Результатами освоения дисциплины являются следующие индикаторы достижения компетенций:

ИОПК-3.3 Использует современные информационные технологии, в том числе отечественного производства на всех этапах разработки программных систем.

ИОПК-3.1 Использует методы построения и анализа алгоритмов при проектировании и разработке программных систем.

ИПК-1.3 Кодирует на языках программирования и проводит модульное тестирование ИС.

ИПК-1.2 Проектирует программное обеспечение.

ИПК-1.1 Определяет, согласовывает и утверждает требования заказчика к ИС.

2. Задачи освоения дисциплины

– Обучить студентов навыкам проектирования, реализации и тестирования приложений для мобильных платформ

3. Место дисциплины в структуре образовательной программы

Дисциплина относится к обязательной части образовательной программы. Дисциплина входит в модуль Разработка программного обеспечения.

4. Семестр(ы) освоения и форма(ы) промежуточной аттестации по дисциплине

Восьмой семестр, зачет

5. Входные требования для освоения дисциплины

Для успешного освоения дисциплины требуются результаты обучения по следующим дисциплинам: Основы программирования.

6. Язык реализации

Русский

7. Объем дисциплины

Общая трудоемкость дисциплины составляет 2 з.е., 72 часов, из которых:

-лабораторные: 32 ч.

Объем самостоятельной работы студента определен учебным планом.

8. Содержание дисциплины, структурированное по темам

Тема 1. Введение в разработку для мобильных платформ. Синтаксис языка Обзор мобильной операционной системы. Изучение среды разработки. Синтаксис языка. Работа с потоками. Разработка и защита проекта (мобильного приложения).

Тема 2. Архитектурные особенности приложения для мобильных платформ Архитектурные шаблоны MVC, MVVM. Жизненные циклы компонентов и приложения. Паттерны и антипаттерны мобильной разработки.

Тема 3. Базовые элементы пользовательского интерфейса
Система отображения элементов GUI. Основные элементы пользовательского интерфейса. Обработка жестов пользователя. Разработка и защита проекта (мобильного приложения).

Тема 4. Списочные элементы пользовательского интерфейса. Анимация
Создание динамических таблиц. Работа с анимацией. Поддержка различных разрешений экрана. Разработка и защита проекта (мобильного приложения).

Тема 5. Хранение данных

Файловая система. Использование баз данных. Другие способы хранения данных.

Тема 6. Особенности разработки приложений для современных версий мобильных операционных систем

Особенности GUI для мобильных устройств. Работа с разрешениями. Модульные и UI-тесты. Разработка и защита проекта (мобильного приложения).

9. Текущий контроль по дисциплине

Текущий контроль по дисциплине проводится путем проведения устных опросов по лекционному материалу, а также в форме сдачи лабораторных работ. Результаты текущего контроля фиксируются в форме контрольной точки не менее одного раза в семестр.

10. Порядок проведения и критерии оценивания промежуточной аттестации

Теоретические и практические результаты формируются компетенциями ИОПК-3.3; ИПК-1.1; ИПК-1.2, ИПК-1.3 и результатами обучения:

№	Этапы формирования компетенций (разделы дисциплины)	Код и наименование результатов обучения	Вид оценочного средства (тесты, задания, кейсы, вопросы и др.)
1.	Введение в разработку для мобильных платформ. Синтаксис языка.	ОР-3.3.1, ОР-1.1.1, ОР-1.2.1, ОР-1.3.1	Вопросы Лабораторные работы
2.	Архитектурные особенности приложений для мобильных платформ.	ОР-3.3.1, ОР-1.1.1, ОР-1.2.1, ОР-1.3.1	Вопросы Лабораторные работы
3.	Базовые элементы пользовательского интерфейса.	ОР-3.3.1, ОР-1.1.1, ОР-1.2.1, ОР-1.3.1	Вопросы Лабораторные работы
4.	Списочные элементы пользовательского интерфейса. Анимация.	ОР-3.3.1, ОР-1.1.1, ОР-1.2.1, ОР-1.3.1	Вопросы Лабораторные работы
5.	Хранение данных.	ОР-3.3.1, ОР-1.1.1, ОР-1.2.1, ОР-1.3.1	Вопросы Лабораторные работы
6.	Особенности разработки приложений для современных версий мобильных операционных систем.	ОР-3.3.1, ОР-1.1.1, ОР-1.2.1, ОР-1.3.1	Вопросы Лабораторные работы

Описание лабораторных работ.

Язык программирования – Kotlin.

Написать приложение, отображающее список элементов на одной Activity (ListActivity) с возможностью просмотреть подробную информацию об элементе списка на другой Activity (DetailsActivity). В качестве таких элементов может выступать кто угодно – музыканты, актёры, политики или что угодно – организации, вещи и т. д.

Каждому элементу должны соответствовать, как минимум, иконка, название, описание рейтинг (float в интервале [0; 5]). В ListActivity достаточно отображать иконку, название и рейтинг (при наличии). В DetailsActivity должна быть возможность изменить рейтинг. Начальный список элементов должен храниться локально в каком-то хранилище (Preferences, файл или база данных), а затем обновляться с сервера. При этом на сервере не будет рейтинга элемента, рейтинг исключительно локальный. Эти изменения должны быть отображены в ListActivity динамически. Данные можно разместить, например, на Google Disk, получив постоянную ссылку на него. Обращение к нему должно происходить посредством Retrofit или OkHttp. Список элементов в ListActivity должен отображаться с помощью RecyclerView. Асинхронные вызовы выполнять с помощью kotlin coroutines. Для каждой Activity нужно реализовывать свой ViewModel.

Последовательность задач при выполнении лабораторных работ

1. Создать ListActivity. Набор элементов генерируется. Сразу должна присутствовать ViewModel с LiveData внутри.
 - a. Добавить сверху экрана надпись с количеством элементов (использовать plurals).
2. Набор элементов для ListActivity получается из локального хранилища.
3. Создать DetailsActivity.
4. Добавить Dependency Injection, реализованный с помощью Dagger.
5. Реализовать получение элементов с сервера, иконки при этом заданы в виде ссылок, которые пока не используются. Если с сервера подгружен элемент, который не хранится локально, для него используется иконка по умолчанию.
6. Загружать иконки по ссылкам динамически с помощью Glide.
 - a. Добавить предзагрузку иконок путём интеграции Glide и RecyclerView.
7. Авторизация. Добавить 2 активности – для регистрации SignUpActivity и для ввода пароля при входе LoginActivity. Проверять пароль автоматически при вводе пользователем значения с задержкой в 150 мс. Пароль на этом этапе можно просто сохранить в Preferences. Если пользователь забыл пароль, должна быть возможность стереть все данные и перейти к SignUpActivity.
 - a. Добавить авторизацию с помощью TouchId.
 - b. Если пользователь вышел из приложения более, чем на 5 минут, показывать при возвращении LoginActivity.
8. Добавить новый тип сборки (build type) «staging». Использовать различные ссылки на файл с элементами для различных типов сборки.

Ввиду практической направленности курса, зачет как таковой не проводится. Вместо этого аттестация осуществляется по результатам выполнения лабораторных работ. Оценка «отлично» («зачтено») ставится при выполнении 7-8 заданий из приведённого выше списка, оценка «хорошо» («зачтено») ставится при выполнении 5-6 заданий и, наконец, «удовлетворительно» («зачтено») – при выполнении 4 заданий.

11. Учебно-методическое обеспечение

- a) Электронный учебный курс по дисциплине в электронном университете «Moodle» - <https://moodle.tsu.ru/course/view.php?id=11604>
- б) Оценочные материалы текущего контроля и промежуточной аттестации по дисциплине.
- в) Методические указания по проведению лабораторных работ.

12. Перечень учебной литературы и ресурсов сети Интернет

а) основная литература:

№ п/п	Авторы / составители	Заглавие	Издательство	Год издания
1.	Jackson W.	Android Apps for Absolute Beginners: Covering Android 7	Apress	2017
2.	Smith D., Hellman E.	Android Recipes: A Problem-Solution Approach	Apress	2016
3.	Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж.	Приемы объектно-ориентированного проектирования. Паттерны проектирования	Питер	2016
4.	Дейтел П., Дейтел Х., Уолд А.	Android для разработчиков	Питер	2016
5.	Филлипс Б.	Программирование под Android	Питер	2014

б) дополнительная литература:

1.	David Mark	Beginning iPhone Development with Swift: Exploring the iOS SDK	Apress	2014
2.	Усов В.	Swift. Основы разработки приложений под iOS и macOS	Питер	2018

в) ресурсы сети Интернет:

Официальный сайт платформы Андроид для разработчиков
<https://developer.android.com>

13. Перечень информационных технологий

а) лицензионное и свободно распространяемое программное обеспечение:
– Среда разработки Android Studio 3.0 и новее

14. Материально-техническое обеспечение

Аудитории для проведения занятий лекционного типа.

Аудитории для проведения занятий семинарского типа, индивидуальных и групповых консультаций, текущего контроля и промежуточной аттестации.

Помещения для самостоятельной работы, оснащенные компьютерной техникой и доступом к сети Интернет, в электронную информационно-образовательную среду и к информационным справочным системам.

15. Информация о разработчиках

Дружинин Денис Вячеславович, канд.техн.наук, доцент кафедры теоретических основ информатики.