

Сборник заданий для практических занятий  
по курсу  
«Разработка мобильных приложений» (МГПУ)

# Содержание

<b>1</b>	<b>Общие сведения</b>	<b>3</b>
<b>2</b>	<b>Практические задания (осень)</b>	<b>3</b>
2.1	Практическое задание «Простейшие программы для Android» . . . . .	4
2.2	Лабораторная работа «Проектирование приложения с использованием MVVM» (2 часа) . . . . .	6
2.3	Практическая работа «Работа со списками» . . . . .	7
2.4	Семинар «Работа с базами данных» . . . . .	9
2.5	Лабораторная работа «Unit-тесты и внедрение зависимостей» (1/2 недели) .	10
2.6	Лабораторная работа «UI-тестирование и Rest API» . . . . .	11
2.7	Лабораторная работа «Paging» . . . . .	12
2.8	Лабораторная работа «First offline» . . . . .	13
2.9	Экзамен . . . . .	14
<b>3</b>	<b>Практические задания (весна)</b>	<b>14</b>
3.1	Лабораторная работа «Сенсоры» . . . . .	15
3.2	Лабораторная работа «Анимация» . . . . .	17
3.3	Лабораторная работа «Камера» (2 недели) . . . . .	18
3.4	Лабораторная работа «Уведомления» (для зачета) . . . . .	19
3.5	Зачёт . . . . .	20
<b>4</b>	<b>Список литературы</b>	<b>20</b>

## 1 Общие сведения

Сборник содержит задания для практических занятий по курсу «Разработка мобильных приложений».

Лекции проводятся в форме мастер-классов, поэтому практические занятия заключаются в выполнении заданий, аналогичных по структуре работам, рассмотренным на мастер-классах.

В практических и лабораторных работах используется язык Kotlin в среде Android Studio. При этом необходимо соблюдать Coding Conventions (требования к стилю кода).

Перед сдачей работы необходимо убедиться, что среда не выдаёт предупреждений при запуске подпункта **Inspect Code** меню **Analyze**.

Во всех заданиях требуется проверять корректность входных данных: программа не должна «падать» ни при каких условиях.

Для групп, где промежуточная аттестация проводится в форме зачёта: зачёт имеет накопительный характер; оценивается каждое практическое задание (которое подлежит защите), строго учитываются сроки сдачи и, в некоторых случаях, качество выполнения. Для групп, где промежуточная аттестация проводится в форме экзамена, экзамен представляет собой защиту проекта (это может быть либо совокупность практических заданий с занятий, либо индивидуальный проект, предварительно согласованный с преподавателем).

В каждом задании предусмотрено 25 вариантов. Выбирайте вариант согласно вашему номеру в журнале.

## 2 Практические задания (осень)

## 2.1 Практическое задание «Простейшие программы для Android»

Разработайте программу, работающую под управлением Android с использованием Jetpack Compose. Проверьте, что программа корректно работает с различными размерами экрана, а также при повороте экрана.

Необходимо учитывать интернационализацию (приложение должно корректно работать как при настройке локали с русским языком по умолчанию, так и английским).

Необходимо использовать принцип DRY (Don't repeat yourself).

Проверяйте и информируйте пользователя обо всех ошибках, встречающихся при вводе (подсвечивая, в том числе, элементы управления) и при подсчетах (в частности переполнение).

1. Программа решения квадратного уравнения
2. Программа решения неравенства вида  $ax + b > 0$
3. Программа решения неравенства вида  $ax + b < 0$
4. Программа решения неравенства вида  $ax + b \geq 0$
5. Программа решения неравенства вида  $ax + b \leq 0$
6. Программа поиска дня недели по числу и месяцу в текущем году
7. Программа перевода числа из 10-ой в 16-ую, 8-ую и 2-ую систем.
8. Программа поиска времени, когда окончится интервал. Дано: часы и минуты начала интервала и количество минут, сколько он идет. Результат: часы и минуты окончания интервала.
9. Программа поиска обратной матрицы для матрицы  $3 \times 3$ .
10. Программа поиска длины интервала. Дано: часы и минуты начала интервала и часы и минуты конца интервала. Результат: количество минут в интервале.
11. Программа умножения и деления двух комплексных чисел.
12. Программа нахождения площади треугольника по координатам вершин.
13. Программа нахождения углов треугольника по координатам вершин (проще всего это сделать по теореме косинусов).
14. Программа перевода числа из 16-ой, 8-ой и 2-ой системы в 10-ую систему счисления.
15. Программа нахождения количества денег на вкладе после окончания его срока по начальному взносу, проценту и срока в годах.
16. Программа нахождения степени комплексного числа. Исходные данные: действительная, мнимая часть числа и степень. Результат: действительная и мнимая часть результата.
17. Программа умножения и деления чисел, представленных в виде обыкновенных дробей (состоящих из целой части, числителя и знаменателя). Не забудьте выполнить сокращение дроби и приведение ее к правильному виду.

18. Программа сложения и вычитания чисел, представленных в виде обыкновенных дробей (состоящих из целой части, числителя и знаменателя). Не забудьте выполнить сокращение дроби и приведение ее к правильному виду.
19. Программа определения по дате (число и месяц) знака зодиака.
20. Программа определения по обыкновенной дроби (числителю и знаменателю) периода десятичной дроби.
21. Программа перевода комплексного числа из обычной формы в тригонометрическую и наоборот.
22. Программа-игра Баше. При реализации этого задания не требуется ничего рисовать, вся информация вводится и выводится в виде чисел в обычные элементы управления.
23. Программа разложения числа на простые множители.
24. Программа нахождения наибольшего общего делителя и наименьшего общего кратного двух натуральных чисел.
25. Программа-тест по предмету «Разработка мобильных приложений». Создайте программу-тест из 10 вопросов с выбором вариантов ответов и показом результатов прохождения теста.

## **2.2 Лабораторная работа «Проектирование приложения с использованием MVVM» (2 часа)**

1. Осуществите рефакторинг проекта, полученного в предыдущей работе, путём внедрения паттерна MVVM. Обратите внимание на то, что для реализации прослушивания изменений значений во ViewModel (и в Model при необходимости) необходимо использовать StateFlow. Используйте библиотеки Jetpack везде, где они применимы.
2. Реализуйте автоматическое UNIT-тестирование класса (объекта) модели и класса ViewModel. Осуществите необходимый рефакторинг. UNIT-тесты должны соответствовать всем стандартными принципам и иметь полное покрытие.
3. Реализуйте автоматическое интеграционное тестирование модели и ViewModel.
4. Реализуйте автоматическое UI-тестирование приложения.

## 2.3 Практическая работа «Работа со списками»

Реализуйте приложение с использованием паттерна MVVM, предназначенное для ввода, изменения, удаления и просмотра информации согласно вашему варианту.

Требования к реализации:

- необходимо использовать Navigation Compose;
  - необходимо создать столько ViewModel, сколько и страниц;
  - необходимо строго соблюдать требования к ViewModel, Model и View (даже строже, чем это предлагают авторы Android).
1. База данных студентов группы. Поля: фамилия, имя, отчество, пол, возраст.
  2. База данных расходов семьи. Поля: товар, стоимость, количество, дата.
  3. База данных загрузки аудиторий. Поля: дата и время, начала, дата и время конца, аудитория, преподаватель.
  4. База данных учета доходов и расходов предпринимателя. Поля: дата, тип операции (доход/расход), объем операции, описание, корреспондент.
  5. База данных велоклуба. Поля: ФИО, тип велосипеда (МТВ и др.), стаж участия в велоклубе.
  6. База данных рейсов авиакомпании. Поля: дата и время вылета, аэропорт вылета, аэропорт прилета, дата и время прилета, марка самолета.
  7. База данных автобусных маршрутов. Поля: номер маршрута, номер парка, времена начала и окончания движения, длина маршрута в км.
  8. База данных электричек. Поля: вокзал, номер поезда, количество вагонов, тип (экспресс/обычный/спутник).
  9. База данных товаров Интернет-магазина. Поля: название товара, категория, цена товара, описание товара.
  10. База заказов Интернет-магазина. Поля: ФИО заказчика, стоимость заказа, скидка (в процентах), адрес доставки.
  11. База данных выборов. Поля: участок, кандидат, количество голосов.
  12. База данных практических работ. Поля: практическая работа, студент, номер варианта, номер уровня, дата сдачи, оценка.
  13. База данных операторов и телеканалов. Поля: Название, тип (спутник, кабель, Интернет), охват (кол-во миллионов домохозяйств), минимальная стоимость подписки.
  14. База данных тарифных планов оператора. Поля: название, тип вещания (обычный/HD), флаг общедоступности.
  15. База данных незаконно огороженных берегов. Поля: водный объект, регион, GPS-координаты, длина недоступного участка берега, дата фиксации нарушения.

16. База данных временного прекращения движения в метро. Поля: дата и время начала прекращения движения, дата и время окончания прекращения движения, станция, станция (от какой до какой станции прекращено движение).
17. База данных проката фильмов. Поля: дата, время, кинотеатр, фильм, номер зала, тип сеанса (3D/Imax/обычный).
18. База данных эвакуированных автомобилей. Поля: улица, автостоянка, GPS-координаты, тип нарушения (стоянка на проезжей части в месте запрета, стоянка на тротуаре, стоянка на газоне), номер автомобиля, тип автомобиля (легковой/грузовой малой тоннажности/грузовой большой тоннажности).
19. База данных средних специальных учебных учреждений. Поля: название, адрес, тип подчинения (федеральный/региональный), год основания, номер лицензии, номер аккредитации, дата окончания действия аккредитации.
20. База данных поселков. Поля: название, девелопер, площадь, количество жителей.
21. База данных сухопутной военной техники. Поля: название, модель, разработчик, предприятие, стоимость, тип.
22. База данных деревьев в городе. Поля: GPS-координаты, вид дерева, округ, год посадки.
23. База данных футбольных матчей. Поля: дата, команда, команда, счет, место проведения.
24. База данных обращений жителей. Поля: дата, время, объект, заявитель, содержание обращения (до 255 символов), дата ответа, ответ на обращение (до 255 символов).
25. База данных студентов колледжа. Поля: ФИО, группа, признак бюджетности, стипендия (нет/обычная/повышенная), флаг наличия социальной стипендии, дата рождения.

## 2.4 Семинар «Работа с базами данных»

Внедрите в приложение, полученное на прошлом занятии, постоянное хранение с использованием библиотеки Room, а также реализуйте частично архитектуру Clean Architecture (в части UseCase и Repository).

Также реализуйте автоматическое внедрение зависимостей посредством Hilt, а также реализуйте автоматическое тестирование usecases.

## **2.5 Лабораторная работа «Unit-тесты и внедрение зависимостей» (1/2 недели)**

Внедрите в приложение, полученное на прошлом занятии, автоматическое внедрение зависимостей посредством Hilt (если требуется) а также реализуйте автоматическое тестирование всех классов приложения, созданных Вами, кроме Activity с полным покрытием, а также UI-тестирование (кроме Room для группы 425).

## **2.6 Лабораторная работа «UI-тестирование и Rest API»**

Осуществите полное UI-тестирование приложения, полученного в прошлой работе.

Осуществите подключение к REST API сервису, полученному в первой работе по предмету «Архитектура веб- и мобильных приложений», и обеспечьте выполнение всех операций приложения с его использованием (создайте новый репозиторий и используйте НЛТ).

## 2.7 Лабораторная работа «Paging»

Реализуйте загрузку с сервиса не целиком всех данных, а постепенно с использованием Paging.

## 2.8 Лабораторная работа «First offline»

Реализуйте работу с данными в офлайн режиме с синхронизацией изменений в фоне (WorkManager, RemoteMediator).

Реализуйте регистрацию, аутентификацию и передачу токенов для проведения операций на сервер.

## **2.9 Экзамен**

Экзамен проводится в форме защиты проекта, реализованного в течение семестра или собственного проекта, который разработан для Android с использованием Jetpack Compose с обязательной реализацией чистой архитектуры, Offline First и работой с внешним сервисом.

## **3 Практические задания (весна)**

### 3.1 Лабораторная работа «Сенсоры»

Разработайте мобильное приложение, предназначенное для отслеживания активности пользователя согласно варианту.

Информация о сохраненных активностях должна сохраняться между сеансами, пользователь должен иметь возможность корректировать результаты неточных измерений.

Ваша задача повысить точность измерения до возможного максимума.

Обратите внимание на использование всех изученных рекомендованных подходов к проектированию приложения.

Замечание: задание апробируется, потому некоторые варианты могут быть недостаточно корректными.

1. учёта количества и глубины приседаний (в предположении, что телефон в руке, а рука поднимается во время приседа вверх);
2. учёта количества отжиманий (в предположении, что телефон в кармане брюк);
3. учёта количества и высоты прыжков на месте (телефон – в кармане брюк);
4. учёт длительности выполнения планки (телефон – в кармане брюк, изначально человек стоит, а потом переходит в позу планки, в конце – встаёт);
5. учёта количества шагов и скорости при беге на месте (в предположении, что телефон в кармане брюк);
6. учёта количества подтягиваний (в предположении, что телефон в кармане брюк);
7. учёта количества подъёма туловища из положения лёжа (в предположении, что телефон - в кармане толстовки);
8. учёта количества отжиманий от стены (в предположении, что телефон в кармане толстовки);
9. учёта количества выпадов вперёд (в положении стоя, телефон – в кармане брюк);
10. учёта количества выпадов в сторону (в положении стоя, телефон – в кармане брюк);
11. учёта количества выпадов назад (в положении стоя, телефон – в кармане брюк);
12. учёта количества и качества выполнений виньясы (телефон – в кармане брюк);
13. учёта количества прыжков через скакалку (телефон – в кармане брюк);
14. измерение глубины и прогресса наклона вперёд (пользователь держит телефон в руке);
15. измерение качества выполнения мостика (телефон лежит на животе);
16. измерение высоты вытяжения (телефон поднимается максимально высоко над головой);
17. измерение количества и качества выполнения взмахов рук в противоположные стороны (одна рука вверх, другая вниз, телефон в руке);
18. подсчёт времени проведенном в сидячем положении (телефон в кармане брюк);

19. подсчёт времени пробегания 30 метров;
20. подсчёт количества отжиманий, выполненных за 60 секунд;
21. подсчёт количества приседаний, выполненных за 60 секунд;
22. подсчёт количества подтягиваний, выполненных за 60 секунд.

### **3.2 Лабораторная работа «Анимация»**

Доработайте приложение, выполненное в предыдущей практической, чтобы оно в виде анимации показывало то, как выполнялось упражнение или измерение пользователем (качество графики оцениваться не будет).

### **3.3 Лабораторная работа «Камера» (2 недели)**

Доработайте приложение, выполненное в предыдущей практической, чтобы оно могло оценивать выполнение упражнения не только по сенсорам, но и по видео (добиться идеала не удастся, тут это не требуется). Используйте библиотеку `mlkit`.

### **3.4 Лабораторная работа «Уведомления» (для зачета)**

Реализуйте вкладку (экран) со статистикой выполнения упражнений пользователем (как табличный, так и графический вариант).

Реализуйте возможность включить напоминание о том, что надо выполнить упражнение в указанное время дня.

### 3.5 Зачёт

Зачёт выставляется посредством вычисления средней арифметической оценки по всем лабораторным работам весеннего семестра с учетом:

- сроков сдачи,
- полного понимания представленного исходного кода (как были написаны самописные части кода, откуда брались части кода, что есть в документации, как в целом производится работа с проектом в среде разработки, полное понимание всех использованных конструкций языка и классов/функций фреймворков и библиотек),
- доли выполнения задания по отношению к идеально выполненному заданию.

Вместо заданий из задачника вы можете разработать приложение в большей степени отвечающее вашим интересам, при условии что оно будет включать сложную работу с сенсорами, нестандартный анализ видео-потока, нестандартную анимацию (зависящую от исходных данных) и оповещения пользователю по аналогии с будильником.

## 4 Список литературы

1. Официальная документация по языку Kotlin: <https://kotlinlang.org/docs/reference/>
2. Официальная документация по платформе Android: <https://developer.android.com/docs>
3. Быстрое введение в Kotlin от авторов языка: <https://stepik.org/course/4222/promo>