МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ

ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»



Направление подготовки/специальность   
09.03.01 Информатика и вычислительная техника

направленность (профиль)/специализация

«Технологии разработки программного обеспечения»

**Выпускная квалификационная работа**

«Разработка мобильного приложения для подготовки к ОГЭ по физике»

| Обучающегося 4 курса  Очной формы обучения  Cумароковой Екатерины  Максимовны |
| --- |
|  |
| Руководитель выпускной квалификационной работы:  Карпова Наталья Александровна, кандидат тех. наук, доцент |
|  |
|  |

2025

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 4](#_heading=h.gjdgxs)

[Глава 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 6](#_heading=h.30j0zll)

[1.1. Структура и содержание экзаменационных материалов ОГЭ по физике 6](#_heading=h.4bdg258ow3hd)

[1. 2. Анализ существующих мобильных решений и их ограничений 7](#_heading=h.v942dynx133e)

[1.3. Методические и пользовательские требования к функционалу 8](#_heading=h.p5nh3e5o9rbs)

[Глава 2 РАЗРАБОТКА И РЕАЛИЗАЦИЯ ПРОГРАММНОГО ПРОДУКТА 10](#_heading=h.2et92p0)

[2.1. Техническое задание на разработку программного продукта 10](#_heading=h.tyjcwt)

[2.1.1 Назначение разработки 10](#_heading=h.12eay12qhd75)

[2.1.2 Требования к платформе и реализации 10](#_heading=h.dymflhy6nzni)

[2.1.3 Структура пользовательского интерфейса 11](#_heading=h.275he5hifp5v)

[2.1.4 Хранение и обработка данных Данные, с которыми работает приложение, делятся на несколько категорий 12](#_heading=h.7dwe7ft0szxw)

[2.1.6 Офлайн-доступ и стабильность 12](#_heading=h.2m4ux4ivwx4d)

[2.2. АРХИТЕКТУРА МОБИЛЬНОГО ПРИЛОЖЕНИЯ 12](#_heading=h.l5w0d694bbcv)

[2.2.1 Общая структура архитектуры 13](#_heading=h.vettkxg78oe2)

[2.2.2 Разделение компонентов по модулям 15](#_heading=h.3zez5fao5jw3)

[Раздел «Теория» 15](#_heading=h.10qvqqwvcrst)

[Раздел «Тесты» 15](#_heading=h.fl1w1l4krj3x)

[Раздел «Задачи» 15](#_heading=h.er5cwmnetlqd)

[Раздел «Прогресс» 16](#_heading=h.jheqormsm5mk)

[2.2.3 Преимущества архитектуры MVVM 16](#_heading=h.fsnsvcexvst9)

[2.2.4 Используемые технологии и инструменты 17](#_heading=h.vc8gdzi4nae3)

[2.3. РЕАЛИЗАЦИЯ ПРОГРАММНОГО ПРОДУКТА 19](#_heading=h.q5p671yvd770)

[Этап 1. Инициализация проекта и настройка окружения 19](#_heading=h.mca8opflltsq)

[Этап 2. Разработка главного экрана 20](#_heading=h.t5byc5clu90r)

[Этап 3. Модуль «Теория» 20](#_heading=h.gag0wcqb33av)

[Этап 4. Модуль «Тесты» 20](#_heading=h.88jgkmo2pr9u)

[Этап 5. Модуль «Прогресс» 21](#_heading=h.1hchodgy896x)

[Этап 6. Модуль «Задачи» 21](#_heading=h.hiz9fwcrpzpu)

[Этап 7. Интеграция и тестирование 22](#_heading=h.69fjlx6y2o2i)

[**ЗАКЛЮЧЕНИЕ 23**](#_heading=h.7weqhmxz75hd)

[СПИСОК ЛИТЕРАТУРЫ 25](#_heading=h.2s8eyo1)

[ПРИЛОЖЕНИЕ А. Программа, файл – MainActivity.kt 27](#_heading=h.3rdcrjn)

[ПРИЛОЖЕНИЕ Б. Программа, файл – ProgressActivity.kt 28](#_heading=h.4jcctjvgbtes)

[ПРИЛОЖЕНИЕ C. Программа, файл – ProgressManager.kt 30](#_heading=h.lnxbz9)

[ПРИЛОЖЕНИЕ D. Программа, файл – TaskActivity.kt 31](#_heading=h.eb2efw6163lf)

[ПРИЛОЖЕНИЕ E. Программа, файл – TaskBlockSelectActivity.kt 37](#_heading=h.lxv9kxi89r5e)

[ПРИЛОЖЕНИЕ F. Программа, файл – TaskMenuActivity.kt 39](#_heading=h.tprhni2apkwf)

[ПРИЛОЖЕНИЕ G. Программа, файл – TestActivity.kt 40](#_heading=h.turnhcahdtog)

[ПРИЛОЖЕНИЕ H. Программа, файл – TestBlockActivity.kt 48](#_heading=h.fx7sio6jgnzj)

[ПРИЛОЖЕНИЕ I. Программа, файл – TestBlockSelectActivity.kt 49](#_heading=h.vgqetdtlf7v2)

[ПРИЛОЖЕНИЕ J. Программа, файл – TheoryBlockSelectActivity.kt 52](#_heading=h.1f34x0q5pg6f)

[ПРИЛОЖЕНИЕ K. Программа, файл–TheoryBlockSelectionActivity.kt 54](#_heading=h.xs1mniowyz6u)

[ПРИЛОЖЕНИЕ L. Программа, файл – TheoryContentActivity.kt 56](#_heading=h.y8qjgvw6vdbx)

[ПРИЛОЖЕНИЕ M. Программа, файл – TheoryData.kt 60](#_heading=h.t2sfv9p419nt)

# ВВЕДЕНИЕ

На фоне стремительного развития цифровых технологий, образование переживает значительную эволюцию, качественные изменения в формах подачи и усвоения учебного материала. Одной из наиболее востребованных задач современной образовательной среды становится повышение доступности и эффективности подготовки учащихся к итоговой государственной аттестации. Особенно остро этот вопрос стоит в отношении таких дисциплин, как физика, которая традиционно считается одной   
из наиболее трудных и требующих комплексного подхода к изучению.

Основной государственный экзамен (ОГЭ) по физике представляет собой важный рубеж в образовательной траектории учащихся, завершивших курс основной школы. Он требует от школьников не только прочного усвоения теоретических основ, но и развития навыков логического мышления, аналитической интерпретации данных, а также способности применять формулы и законы физики к решению практических задач. Традиционные формы подготовки - учебники, тетради, репетиторские занятия - не всегда отвечают требованиям индивидуализации и интерактивности, особенно в условиях ограниченного времени и высокой загруженности школьников.

Мобильные образовательные приложения выступают как инновационный инструмент, способный кардинально изменить подход к обучению. Они позволяют организовать гибкий график занятий, адаптировать нагрузку под уровень конкретного ученика, а также включать в процесс обучения элементы интерактивности и визуализации. Всё это способствует формированию устойчивого интереса к предмету, развитию самостоятельности и повышению учебной мотивации.

Разработка мобильного приложения, ориентированного на подготовку к ОГЭ по физике, представляет собой актуальную задачу, сочетающую в себе как инженерно-программные, так и методические компоненты. Для качественной реализации такого программного продукта необходимо учитывать особенности контрольно-измерительных материалов, анализировать опыт существующих цифровых решений, и сформировать набор функциональных, педагогических и пользовательских требований, обеспечивающих его эффективность.

Настоящая дипломная работа посвящена проектированию и реализации мобильного приложения, предназначенного для организации эффективной самостоятельной подготовки учащихся к сдаче Основного государственного экзамена (ОГЭ) по физике. В теоретической части формируется аналитическая и методическая база проекта: анализируются структура и содержание экзаменационных материалов, проводится сравнительный обзор существующих цифровых решений в данной области и выявляются их ограничения. На основе полученных выводов разрабатываются обоснованные требования к функциональности, интерфейсу и педагогической ценности разрабатываемого программного продукта. Практическая часть работы включает реализацию приложения на языке программирования Kotlin с использованием технологий, обеспечивающих адаптивность и интерактивность учебного процесса.

# 

# Глава 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

## Структура и содержание экзаменационных материалов ОГЭ по физике

Основной государственный экзамен по физике проводится на основе Федерального государственного образовательного стандарта (ФГОС) и охватывает ключевые элементы курса физики с 7 по 9 класс. Структура КИМов (контрольно-измерительных материалов) включает в себя задания, проверяющие как базовые знания, так и умения анализа, синтеза и применения физической информации.

Экзаменационная работа состоит из двух частей:

1. Часть первая — задания с кратким ответом (на выбор одного верного или с числовым ответом);
2. Часть вторая — задания с развернутым ответом, включающие задачи на вычисления, объяснение физических явлений и анализ графиков.

Содержание экзамена охватывает следующие разделы:

* Механика;
* Молекулярная физика и термодинамика;
* Электродинамика;
* Оптика;
* Основы квантовой физики (на школьном уровне).

Также проверяются умения интерпретировать данные экспериментов, анализировать графическую информацию и обоснованно излагать логические рассуждения при решении задач. Успешная сдача экзамена требует как крепкой теоретической базы, так и высокой степени практической подготовки.

## *1. 2.* *Анализ существующих мобильных решений и их ограничений*

На рынке мобильных приложений представлены десятки продуктов, связанных с подготовкой к ОГЭ. Среди них:

* «Решу ОГЭ» (официальное приложение) — предлагает базу заданий, но имеет устаревший интерфейс, не позволяет сформировать индивидуальный маршрут подготовки, не включает объяснения решений.
* «ОГЭ Физика: Подготовка и Тесты» — фокусируется на тестах, практически отсутствует теория и интерактивные элементы.
* «Учи.ру», «Фоксфорд», «Тетрика» — универсальные образовательные платформы, физике уделено мало внимания, а контент не адаптирован строго под структуру ОГЭ.

Основные выявленные недостатки существующих решений:

* отсутствие целостного курса по подготовке, соответствующего требованиям ФИПИ;
* ограниченная интерактивность (нет возможности поэтапного решения задач с проверкой);
* не реализована логика подготовки «от темы к типу задания».

Это указывает на необходимость разработки специализированного, методически выверенного продукта, ориентированного исключительно на экзаменационные цели.

## Методические и пользовательские требования к функционалу

Разработка мобильного приложения для подготовки к ОГЭ по физике предполагает чёткое структурирование функционала с учётом педагогических задач, потребностей целевой аудитории и особенностей экзаменационного формата. Приложение должно выступать в роли универсального помощника школьника, охватывающего все основные аспекты подготовки — от изучения теории до оценки текущих достижений.

Структура интерфейса приложения будет представлять собой главный экран (главную страницу), на котором пользователю предлагаются следующие основные блоки:

1. **Теория** — раздел, содержащий краткие, но содержательные конспекты по всем темам школьного курса физики (механика, термодинамика, электродинамика, оптика и основы квантовой физики).
2. **Тесты** — интерактивный раздел с задачами в формате экзамена ОГЭ (варианты с выбором ответа, числовые ответы). Тесты структурированы по темам, что позволяет школьнику прорабатывать конкретные разделы физики.
3. **Задачи** — блок, содержащий задачи открытого типа с развернутыми решениями. Задания разбиты по темам. Пользователь может использовать задачи для глубокого освоения материала.
4. **Прогресс** — аналитический модуль, отображающий индивидуальные результаты пользователя: количество пройденных тем, выполненных тестов, правильные и неправильные ответы, динамика успеваемости, а также рекомендации по повторению. Такой модуль формирует у ученика понимание своих сильных и слабых сторон и повышает учебную мотивацию.

Методические требования:

* Полное соответствие содержания школьной программе и ФГОС;
* Поддержка самостоятельного и повторного изучения материала;
* Включение механизмов самопроверки и тематического контроля знаний.

Пользовательские требования:

* Простой, интуитивный и минималистичный интерфейс;
* Поддержка мобильных платформ Android;
* Работа в офлайн-режиме для теории и справочников;

Таким образом, продуманная структура приложения и соответствие его функционала образовательным и пользовательским ожиданиям позволяет говорить о высоком потенциале продукта как эффективного инструмента подготовки к ОГЭ по физике.

# 

# Глава 2 РАЗРАБОТКА И РЕАЛИЗАЦИЯ ПРОГРАММНОГО ПРОДУКТА

## 2.1. Техническое задание на разработку программного продукта

### ***2.1.1 Назначение разработки***

Мобильное приложение, разрабатываемое в рамках данной выпускной квалификационной работы, предназначено для учащихся 9-х классов, готовящихся к сдаче Основного государственного экзамена (ОГЭ) по физике. Актуальность разработки обусловлена необходимостью адаптации учебного процесса к современным цифровым условиям и повышением эффективности самостоятельной подготовки школьников.

Основной задачей приложения является предоставление пользователю удобного, структурированного и многофункционального инструмента, сочетающего в себе как теоретический учебный материал, так и практические задания и тесты в формате, приближенном к экзаменационному.

Таким образом, назначение разработки — создание цифрового продукта, способного повысить уровень знаний школьников по физике, облегчить процесс подготовки и сократить потребность в дополнительных платных образовательных услугах.

### 2.1.2 Требования к платформе и реализации

Разработка мобильного приложения осуществляется в среде IntelliJ IDEA, которая поддерживает создание Android-приложений с использованием всех необходимых инструментов и библиотек. В качестве языка программирования выбран Kotlin — современный, краткий и безопасный язык, полностью совместимый с экосистемой Android Jetpack. Его использование позволяет создавать надёжный, удобочитаемый и легко сопровождаемый код.

Приложение ориентировано на работу на устройствах с операционной системой Android версии 8.0 (API level 26) и выше. Такой выбор обусловлен широкой распространенностью данной версии среди пользователей, а также наличием поддержки всех необходимых компонентов пользовательского интерфейса и архитектурных решений.

В качестве архитектурного шаблона в проекте используется MVVM (Model-View-ViewModel), обеспечивающий модульность, логическую разделенность ответственности и удобство тестирования.

### 2.1.3 Структура пользовательского интерфейса

Пользовательский интерфейс построен по принципу простоты и интуитивности, соответствуя рекомендациям Material Design от Google. Главный экран содержит пять основных интерактивных элементов (кнопок), ведущих к основным разделам:

* "Теория" — открывает список тем с теоретическими материалами;
* "Тесты" — предоставляет выбор по тематическим блокам и экзаменационные варианты;
* "Задачи" — раздел с расчетными и логическими задачами с поэтапными решениями;
* "Прогресс" — экран анализа успеваемости пользователя на основе статистики.

Дизайн приложения адаптирован под различные размеры экранов и поддерживает как портретный, так и ландшафтный режимы отображения. Цветовая палитра выбрана нейтральной, с акцентом на выделение активных элементов управления.

### 2.1.4 Хранение и обработка данных Данные, с которыми работает приложение, делятся на несколько категорий

1. Теоретические материалы (тексты, заголовки);
2. Тестовые задания (вопросы, варианты ответов, правильные ответы);
3. Задачи (условие, решение, пояснение);
4. Прогресс пользователя (выполненные задания, динамика).

### 2.1.6 Офлайн-доступ и стабильность

Одним из ключевых требований к мобильному приложению является его полная автономность. Учащийся должен иметь возможность использовать весь функционал без наличия доступа к интернету. Это реализуется путём предварительной загрузки всего учебного контента в локальную базу при первом запуске приложения.

Таким образом, техническое задание описывает приложение как надежный, гибкий и полнофункциональный инструмент подготовки к ОГЭ, полностью реализуемый в среде IntelliJ IDEA на языке Kotlin и соответствующий требованиям современной образовательной цифровой среды.

## 2.2. АРХИТЕКТУРА МОБИЛЬНОГО ПРИЛОЖЕНИЯ

Проектируемое мобильное приложение для подготовки к ОГЭ по физике построено на основе архитектурного шаблона MVVM (Model-View-ViewModel), рекомендованного для разработки Android-приложений в экосистеме Jetpack. Такой подход обеспечивает чёткое разделение логики, улучшает читаемость кода, упрощает тестирование и масштабирование.

### 2.2.1 Общая структура архитектуры

Архитектура приложения **OgePhysics**, разрабатываемого в среде **IntelliJ IDEA** на языке **Kotlin**, построена на основе разделения по уровням, обеспечивающего логическую структурированность, читаемость и упрощённую поддержку кода. В приложении реализованы следующие ключевые уровни:

#### 1. View (Представление):

Пользовательский интерфейс приложения реализован с использованием **Activity**. Каждый раздел приложения оформлен в виде отдельной Activity, без использования Fragment. View-слой отвечает за отображение информации, взаимодействие с пользователем и переходы между экранами.

В проекте реализованы следующие экраны:

* MainActivity — главный экран с навигацией по разделам.
* TheoryMenuActivity и TheoryContentActivity — выбор теоретического блока и отображение соответствующего материала.
* TestsMenuActivity и TestsActivity — выбор блока с тестами и выполнение тестовых заданий.
* TasksActivity — решение задач.
* ProgressActivity — отображение графика прогресса на основе сохранённых результатов.

Переходы между экранами реализованы через Intent, обработка нажатий — через слушатели кнопок (setOnClickListener).

#### 2. Логика взаимодействия:

Всё поведение приложения (логика переключения между вопросами, проверки ответов, отображения статистики и т.д.) реализовано напрямую в соответствующих Activity. В проекте **не используется** отдельный уровень ViewModel или архитектурные библиотеки Jetpack (LiveData, ViewModel, Navigation и др.), так как структура приложения остаётся сравнительно простой и не требует сложной архитектуры.

Таким образом, архитектура приложения основана на классическом подходе с разделением по активностям, использованием простых структур данных и локального хранилища (SharedPreferences). Такой подход обеспечивает быструю разработку, ясную структуру кода и простоту в сопровождении на текущем уровне сложности проекта.

### 2.2.2 Разделение компонентов по модулям

Каждый функциональный блок приложения реализуется как отдельная Activity, что соответствует выбранной архитектуре проекта и уровню его сложности. Такой подход обеспечивает простоту реализации, наглядность структуры и достаточную модульность при отсутствии избыточной архитектурной надстройки:

### **Раздел «Теория»**

* TheoryMenuActivity — отображает список теоретических блоков (тем), доступных для изучения.
* TheoryContentActivity — отображает краткое теоретическое содержание по выбранному блоку.
* Данные о блоках и темах загружаются из ресурсов приложения и представлены в виде обычных List<String>.

### **Раздел «Тесты»**

* TestsMenuActivity — интерфейс выбора тестов по пяти тематическим блокам.
* TestsActivity — отображает отдельные тестовые задания, реализует навигацию по вопросам, ввод и проверку ответов.
* Результаты тестов сохраняются через SharedPreferences, что позволяет отслеживать успехи пользователя без использования базы данных.

### **Раздел «Задачи»**

* TasksActivity — экран с набором задач по разделам ОГЭ. Каждая задача содержит условие, поле для ответа и кнопку проверки.
* Задачи и их пошаговые решения формируются программно и хранятся в виде структур данных внутри приложения.
* Результаты также сохраняются через SharedPreferences.

### **Раздел «Прогресс»**

* ProgressActivity — отображает статистику выполнения тестов и задач.
* Статистические данные (количество правильных и неправильных ответов) извлекаются из SharedPreferences.
* Для построения графика используется библиотека **MPAndroidChart**, позволяющая наглядно визуализировать учебный прогресс.

Таким образом, архитектура каждого раздела реализована в виде самостоятельной Activity, данные хранятся либо в ресурсах, либо в SharedPreferences. Это решение упрощает реализацию, делает проект удобным для сопровождения и позволяет сфокусироваться на функциональности, необходимой для подготовки к ОГЭ по физике.

### 2.2.3 Преимущества архитектуры MVVM

Выбор архитектурного шаблона MVVM на языке Kotlin обеспечивает следующие преимущества:

* **Модульность** — каждую часть логики можно тестировать и изменять независимо;
* **Реактивность** — LiveData позволяет обновлять интерфейс автоматически при изменении данных;
* **Гибкость** — легко добавлять новые разделы, темы, формулы без переработки архитектуры;
* **Безопасность типов и лаконичность кода** — благодаря использованию Kotlin синтаксис проекта остаётся чистым и понятным;
* **Тестируемость** — ViewModel и Repository можно проверять с помощью юнит-тестов.

### 2.2.4 Используемые технологии и инструменты

Среда разработки:

* IntelliJ IDEA — используется как основная среда разработки. Она обеспечивает поддержку Android-проектов, автодополнение кода, встроенный Gradle и эмулятор Android-устройств.

Язык программирования:

* Kotlin — выбран как основной язык благодаря своей лаконичности, безопасности типов, полной совместимости с Android SDK и активной поддержке со стороны Google.

Целевая платформа:

* Android 8.0 (API level 26) и выше — выбрана как минимальная версия, охватывающая большую часть современных устройств. Это обеспечивает корректную работу всех компонентов пользовательского интерфейса.

## 2.3. РЕАЛИЗАЦИЯ ПРОГРАММНОГО ПРОДУКТА

Разработка мобильного приложения «OgePhysics» в рамках проекта vkr осуществлялась поэтапно с учётом современных требований к пользовательскому опыту, производительности и методической ценности продукта. Приложение предназначено для подготовки учащихся к основному государственному экзамену по физике, а значит, должно обеспечивать не только корректную функциональность, но и соответствовать структуре экзаменационных материалов, быть интуитивно понятным и устойчивым к ошибкам.

Программная реализация велась в среде IntelliJ IDEA, где был создан Android-проект с использованием языка Kotlin. Данный язык был выбран благодаря его лаконичному синтаксису, активной поддержке со стороны сообщества и официальной поддержке со стороны Google как основного языка для Android-разработки. Ниже описывается последовательность этапов реализации программного продукта.

### Этап 1. Инициализация проекта и настройка окружения

Сначала был создан проект с именем vkr, в котором заданы параметры базового окружения: целевая версия Android SDK, уровень совместимости Kotlin, основные зависимости, а также минимально необходимый набор разрешений в AndroidManifest.xml. В проекте сразу был определён пакет com.example.myapplication, под которым структурировались все будущие компоненты приложения.

В процессе настройки были подключены необходимые библиотеки:

– для визуализации данных (например, MPAndroidChart);

– для управления навигацией между экранами;

– для хранения пользовательских данных через SharedPreferences.

## 

### Этап 2. Разработка главного экрана

Главное окно приложения реализовано как центральная точка входа, откуда пользователь может перейти в четыре основные секции: «Теория», «Тесты», «Задачи», «Прогресс». Разметка выполнялась в формате XML, логика — на языке Kotlin. Особое внимание было уделено адаптивности интерфейса: элементы управления масштабируются в зависимости от разрешения экрана, фон корректно подстраивается под ориентацию устройства.

## 

### Этап 3. Модуль «Теория»

Раздел «Теория» предназначен для ознакомления с ключевыми физическими понятиями, формулами и законами. Информация структурирована по пяти тематическим блокам, отражающим содержание ОГЭ. Пользователь выбирает интересующий блок, после чего открывается последовательность карточек с теоретическим материалом. Навигация осуществляется кнопками вперёд и назад. Для каждого блока была написана авторская краткая теория, адаптированная под школьный уровень и актуальные требования ФИПИ.

### Этап 4. Модуль «Тесты»

В разделе «Тесты» реализована система тренировки навыков решения задач в формате ОГЭ. Для каждого блока разработано по 10 тестовых заданий, каждое из которых представляет собой вопрос с несколькими вариантами ответа. Пользователь может выбрать один из вариантов, получить мгновенную обратную связь о правильности выбора, а также перейти к следующему заданию. Ответы проверяются на основе заранее заданной логики в коде, без подключения к интернету. По завершении теста результаты сохраняются.

## 

### Этап 5. Модуль «Прогресс»

Этот раздел позволяет учащемуся отслеживать свои успехи и видеть динамику. Все данные о прохождении тестов и решении задач сохраняются в локальное хранилище. При каждом открытии раздела строится график, где отображаются количество решённых заданий и процент правильных ответов по блокам. Это позволяет пользователю оценивать, в каких темах он силён, а какие требуют повторения. Для визуализации применена библиотека MPAndroidChart, позволяющая построить гибкие и наглядные диаграммы.

## 

### Этап 6. Модуль «Задачи»

Раздел «Задачи» разработан для углублённой практики. В отличие от тестов, здесь пользователю необходимо самостоятельно ввести численный ответ. Реализован механизм подсказок, отображения правильного решения после проверки и подробного пояснения. Каждая задача сопровождается проверкой введённого значения с заданной точностью. Пользователь также может переходить между задачами, повторно открывать их и видеть, какие из них уже были решены.

## 

### Этап 7. Интеграция и тестирование

После завершения разработки всех модулей приложение было собрано в единый продукт. Произведено комплексное тестирование на устройствах с разными версиями Android. Особое внимание уделялось корректности работы навигации, сохранению данных, адаптации интерфейса и общей стабильности приложения. Были устранены все выявленные ошибки, а интерфейс приведен к единому стилю.

# 

# ЗАКЛЮЧЕНИЕ

В ходе выполнения дипломного проекта была поставлена и успешно реализована задача разработки мобильного программного средства, предназначенного для подготовки учащихся к основному государственному экзамену по физике. Приложение «OgePhysics», созданное в рамках проекта vkr, ориентировано на обучающую и практико-ориентированную деятельность школьников, находящихся на этапе подготовки к ОГЭ, и представляет собой эффективный цифровой инструмент, объединяющий теоретическую базу, тренировочные задания, задачи повышенного уровня и систему отслеживания личного прогресса.

Разработка велась с применением современных технологий: языка программирования Kotlin, среды IntelliJ IDEA, и инструментов, предоставляемых Android SDK. Весь функционал был реализован с учётом требований к юзабилити, методической достоверности, адаптивности интерфейса и автономности использования без необходимости подключения к сети Интернет.

По результатам анализа и тестирования можно сделать вывод, что созданное приложение:

* соответствует структуре экзаменационных материалов ОГЭ по физике;
* предоставляет пользователю понятный интерфейс с логичной навигацией;
* способствует систематизации и закреплению знаний;
* позволяет в динамике отслеживать личные успехи и выявлять пробелы в подготовке;
* может использоваться как дополнение к традиционным формам обучения.

Таким образом, проект vkr обладает не только учебной ценностью, но и практической значимостью. Его реализация демонстрирует возможность применения мобильных технологий в образовательной среде как в контексте формального образования, так и для самостоятельной подготовки. Разработка подобного рода продуктов актуальна в условиях цифровизации образования и повышения роли индивидуальных траекторий обучения.

Представленная работа может стать основой для дальнейшего расширения функционала приложения, включая подключение удалённых баз данных, внедрение онлайн-тестирования, персонализированных рекомендаций и интеграции с государственными образовательными платформами. В будущем возможно масштабирование проекта для подготовки к другим предметам, входящим в перечень ОГЭ.

# 

# СПИСОК ЛИТЕРАТУРЫ

1. Bloch, J. Effective Java [Текст] / Joshua Bloch. — 3rd ed. — Boston: Addison-Wesley, 2018. — 416 c.
2. Meier, R. Professional Android [Текст] / Reto Meier, Ian Lake. — 4th ed. — Indianapolis: Wrox, 2018. — 928 c.
3. Nakamura, T. Physics in Your Pocket: Pocket-Sized Physics Handbook [Текст] / Takashi Nakamura. — Tokyo: Springer, 2020. — 180 c.
4. Флэнаган, Д. JavaScript. Подробное руководство [Текст] / Дэвид Флэнаган. — 6-е изд. — Санкт-Петербург : Питер, 2020. — 960 с.
5. Шилдт, Г. Java: Полное руководство [Текст] / Герберт Шилдт. — 12-е изд. — Москва: Вильямс, 2022. — 1440 с.
6. Андреев, С. Ю. Разработка мобильных приложений на платформе Android [Текст] / С. Ю. Андреев. — Москва: БХВ-Петербург, 2021. — 368 с.
7. Фаулер, М. Архитектура корпоративных приложений [Текст] / Мартин Фаулер. — Москва: Вильямс, 2020. — 560 с.
8. Воронов, В. И. Физика: краткий курс подготовки к ОГЭ [Текст] / В. И. Воронов. — Москва: Эксмо, 2023. — 240 с.
9. Громов, В. М. Информационные технологии в образовании [Текст] / В. М. Громов. — Санкт-Петербург: Питер, 2022. — 320 с.
10. Kotlin Documentation [Электронный ресурс] — URL: https://kotlinlang.org/docs/home.html (дата обращения: 25.05.2025).
11. Андреев, А. Н. Электронное обучение: современные технологии [Текст] / А. Н. Андреев. — Москва: Юрайт, 2023. — 430 с.
12. Burd, B. Java for Android Developers For Dummies [Текст] / Barry Burd. — Hoboken: Wiley, 2016. — 432 c.
13. Носов, К. И. Программирование на языке Kotlin: практическое руководство [Текст] / К. И. Носов. — Москва: ДМК Пресс, 2021. — 376 с.
14. Бабенко, А. Ю. Цифровая трансформация образования [Текст] / А. Ю. Бабенко. — Новосибирск: НГТУ, 2022. — 280 с.

## 

## ПРИЛОЖЕНИЕ А. Программа, файл – MainActivity.kt

package com.example.vkr

import android.content.Intent

import android.os.Bundle

import android.widget.Button

import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

val buttonTheory = findViewById<Button>(R.id.buttonTheory)

val buttonTests = findViewById<Button>(R.id.buttonTests)

val buttonTasks = findViewById<Button>(R.id.buttonTasks)

val buttonProgress = findViewById<Button>(R.id.buttonProgress)

buttonTheory.setOnClickListener {

val intent = Intent(this, TheoryBlockSelectActivity::class.java)

startActivity(intent)

}

buttonTests.setOnClickListener {

val intent = Intent(this, TestBlockSelectActivity::class.java)

startActivity(intent)

}

buttonTasks.setOnClickListener {

val intent = Intent(this, TaskBlockSelectActivity::class.java)

startActivity(intent)

}

buttonProgress.setOnClickListener {

val intent = Intent(this, ProgressActivity::class.java)

startActivity(intent)

}

}

}

## ПРИЛОЖЕНИЕ Б. Программа, файл – ProgressActivity.kt

package com.example.vkr

import android.content.Context

import android.os.Bundle

import android.widget.Button

import android.widget.ProgressBar

import android.widget.TextView

import androidx.appcompat.app.AppCompatActivity

class ProgressActivity : AppCompatActivity() {

private lateinit var progressBar: ProgressBar

private lateinit var textProgress: TextView

private lateinit var buttonBack: Button

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_progress)

progressBar = findViewById(R.id.progressBar)

textProgress = findViewById(R.id.textProgress)

buttonBack = findViewById(R.id.buttonBack)

// Получение сохранённой статистики (SharedPreferences)

val sharedPref = getSharedPreferences("progress", Context.MODE\_PRIVATE)

val totalQuestions = sharedPref.getInt("total", 0)

val correctAnswers = sharedPref.getInt("correct", 0)

val progressPercent = if (totalQuestions > 0) {

(correctAnswers \* 100) / totalQuestions

} else {

0

}

progressBar.progress = progressPercent

textProgress.text = "Правильных ответов: $correctAnswers из $totalQuestions\nПрогресс: $progressPercent%"

buttonBack.setOnClickListener {

finish()

}

}

}

## ПРИЛОЖЕНИЕ C. Программа, файл – ProgressManager.kt

package com.example.vkr

import android.content.Context

object ProgressManager {

private const val PREFS\_NAME = "progress\_prefs"

fun saveCorrectAnswer(context: Context, topic: String) {

val prefs = context.getSharedPreferences(PREFS\_NAME, Context.MODE\_PRIVATE)

val current = prefs.getInt(topic, 0)

prefs.edit().putInt(topic, current + 1).apply()

}

fun getCorrectAnswers(context: Context): Map<String, Int> {

val prefs = context.getSharedPreferences(PREFS\_NAME, Context.MODE\_PRIVATE)

return mapOf(

"Механика" to prefs.getInt("Механика", 0),

"Термодинамика" to prefs.getInt("Термодинамика", 0),

"Электричество" to prefs.getInt("Электричество", 0),

"Оптика" to prefs.getInt("Оптика", 0),

"Ядерная" to prefs.getInt("Ядерная", 0)

)

}

fun resetProgress(context: Context) {

val prefs = context.getSharedPreferences(PREFS\_NAME, Context.MODE\_PRIVATE)

prefs.edit().clear().apply()

}

}

## ПРИЛОЖЕНИЕ D. Программа, файл – TaskActivity.kt

package com.example.vkr

import android.os.Bundle

import android.widget.\*

import androidx.appcompat.app.AlertDialog

import androidx.appcompat.app.AppCompatActivity

class TaskActivity : AppCompatActivity() {

private lateinit var taskTextView: TextView

private lateinit var answerEditText: EditText

private lateinit var checkButton: Button

private lateinit var nextButton: Button

private lateinit var backButton: Button

private var currentIndex = 0

private var correctAnswers = 0

private var incorrectAnswers = 0

private lateinit var tasks: List<Pair<String, String>>

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_task)

val blockName = intent.getStringExtra("BLOCK\_NAME") ?: "Блок"

taskTextView = findViewById(R.id.taskTextView)

answerEditText = findViewById(R.id.answerEditText)

checkButton = findViewById(R.id.checkButton)

nextButton = findViewById(R.id.nextButton)

backButton = findViewById(R.id.backButton)

tasks = getTasksForBlock(blockName)

showCurrentTask()

checkButton.setOnClickListener {

val userAnswer = answerEditText.text.toString().trim()

if (userAnswer.isEmpty()) {

Toast.makeText(this, "Введите ответ", Toast.LENGTH\_SHORT).show()

return@setOnClickListener

}

val correctAnswer = tasks[currentIndex].second

if (userAnswer == correctAnswer) {

Toast.makeText(this, "Правильно!", Toast.LENGTH\_SHORT).show()

correctAnswers++

} else {

Toast.makeText(this, "Неправильно. Верный ответ: $correctAnswer", Toast.LENGTH\_SHORT).show()

incorrectAnswers++

}

checkButton.isEnabled = false

}

nextButton.setOnClickListener {

if (currentIndex < tasks.size - 1) {

currentIndex++

showCurrentTask()

} else {

showResult()

}

}

backButton.setOnClickListener {

if (currentIndex > 0) {

currentIndex--

showCurrentTask()

} else {

finish()

}

}

}

private fun showCurrentTask() {

taskTextView.text = tasks[currentIndex].first

answerEditText.text.clear()

checkButton.isEnabled = true

}

private fun showResult() {

AlertDialog.Builder(this)

.setTitle("Результат")

.setMessage("Правильных ответов: $correctAnswers\nНеправильных: $incorrectAnswers")

.setPositiveButton("ОК") { \_, \_ -> finish() }

.setCancelable(false)

.show()

}

private fun getTasksForBlock(blockName: String): List<Pair<String, String>> {

return when (blockName) {

"Механика" -> listOf(

"Тело движется с ускорением 2 м/с². Найдите путь за 5 секунд." to "25",

"С каким ускорением падает тело на Земле?" to "9.8",

"Найдите скорость тела через 4 с, если начальная 0, ускорение 3 м/с²." to "12",

"Вычислите импульс тела массой 2 кг и скоростью 3 м/с." to "6",

"Как называется первый закон Ньютона?" to "Инерция"

)

"Термодинамика" -> listOf(

"Как обозначается удельная теплоемкость?" to "c",

"Чему равна температура плавления льда?" to "0",

"Что такое теплопередача?" to "Передача тепла",

"Формула количества теплоты?" to "Q=cmΔt",

"Что происходит при испарении?" to "Охлаждение"

)

"Электричество" -> listOf(

"Формула закона Ома?" to "I=U/R",

"Единица измерения тока?" to "Ампер",

"Что такое сопротивление?" to "Препятствие току",

"Прибор для измерения напряжения?" to "Вольтметр",

"Как направлен ток?" to "От плюса к минусу"

)

"Оптика" -> listOf(

"Что такое отражение света?" to "Возврат",

"Закон отражения?" to "Угол падения = угол отражения",

"Скорость света в вакууме?" to "300000",

"Прибор для изучения спектра?" to "Спектроскоп",

"Что делает линза?" to "Преломляет"

)

"Ядерная" -> listOf(

"Что такое альфа-частица?" to "Ядро гелия",

"Опасен ли гамма-луч?" to "Да",

"Что делает радиация?" to "Ионизирует",

"Что изучает ядерная физика?" to "Ядра",

"Единица активности?" to "Беккерель"

)

else -> emptyList()

}

}

}

## ПРИЛОЖЕНИЕ E. Программа, файл – TaskBlockSelectActivity.kt

package com.example.vkr

import android.content.Intent

import android.os.Bundle

import android.widget.Button

import androidx.appcompat.app.AppCompatActivity

class TaskBlockSelectActivity : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_task\_block\_select)

findViewById<Button>(R.id.buttonTaskMechanics).setOnClickListener {

openTaskScreen("Механика")

}

findViewById<Button>(R.id.buttonTaskThermo).setOnClickListener {

openTaskScreen("Термодинамика")

}

findViewById<Button>(R.id.buttonTaskElectricity).setOnClickListener {

openTaskScreen("Электричество")

}

findViewById<Button>(R.id.buttonTaskOptics).setOnClickListener {

openTaskScreen("Оптика")

}

findViewById<Button>(R.id.buttonTaskAtomic).setOnClickListener {

openTaskScreen("Ядерная")

}

}

private fun openTaskScreen(block: String) {

val intent = Intent(this, TaskActivity::class.java)

intent.putExtra("BLOCK\_NAME", block)

startActivity(intent)

}

}

## ПРИЛОЖЕНИЕ F. Программа, файл – TaskMenuActivity.kt

package com.example.vkr

import android.content.Intent

import android.os.Bundle

import android.widget.Button

import androidx.appcompat.app.AppCompatActivity

class TaskMenuActivity : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_task\_menu)

findViewById<Button>(R.id.buttonMechanics).setOnClickListener {

openTaskActivity("Механика")

}

findViewById<Button>(R.id.buttonThermo).setOnClickListener {

openTaskActivity("Термодинамика")

}

findViewById<Button>(R.id.buttonElectricity).setOnClickListener {

openTaskActivity("Электричество и магнетизм")

}

findViewById<Button>(R.id.buttonOptics).setOnClickListener {

openTaskActivity("Оптика")

}

findViewById<Button>(R.id.buttonAtomic).setOnClickListener {

openTaskActivity("Атомная и ядерная физика")

}

}

private fun openTaskActivity(topic: String) {

val intent = Intent(this, TaskActivity::class.java)

intent.putExtra("TOPIC", topic)

startActivity(intent)

}

}

## ПРИЛОЖЕНИЕ G. Программа, файл – TestActivity.kt

package com.example.vkr

import android.os.Bundle

import android.widget.\*

import androidx.appcompat.app.AlertDialog

import androidx.appcompat.app.AppCompatActivity

class TestActivity : AppCompatActivity() {

private lateinit var questionText: TextView

private lateinit var answerInput: EditText

private lateinit var nextButton: Button

private lateinit var backButton: Button

private var topic: String = ""

private var currentQuestionIndex = 0

private var correctAnswers = 0

private lateinit var questions: List<Pair<String, String>> // Пары вопрос-ответ

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_test)

topic = intent.getStringExtra("TOPIC\_KEY") ?: "unknown"

// Загружаем вопросы по теме

questions = loadQuestionsByTopic(topic)

questionText = findViewById(R.id.textQuestion)

answerInput = findViewById(R.id.inputAnswer)

nextButton = findViewById(R.id.buttonNext)

backButton = findViewById(R.id.buttonBack)

showQuestion()

nextButton.setOnClickListener {

checkAnswer()

}

backButton.setOnClickListener {

finish()

}

}

private fun loadQuestionsByTopic(topic: String): List<Pair<String, String>> {

return when (topic) {

"mechanics" -> listOf(

"С какой скоростью движется тело, если за 5 секунд оно прошло 20 метров?" to "4",

"Как называется величина, равная произведению массы на скорость?" to "импульс",

"Какая сила действует на тело массой 2 кг, если ускорение 3 м/с²?" to "6",

"Какой путь пройдет тело за 2 с при скорости 10 м/с?" to "20",

"Что такое инерция?" to "сохранение состояния покоя или равномерного движения",

"Что такое масса?" to "мера инертности",

"Закон Ньютона, описывающий силу?" to "второй",

"Какая формула ускорения?" to "a=v/t",

"Единица измерения силы?" to "ньютон",

"Что показывает график зависимости x(t)?" to "путь от времени"

)

"thermo" -> listOf(

"Что такое температура тела?" to "мера средней кинетической энергии молекул",

"Как обозначается количество теплоты?" to "q",

"Какая формула расчёта теплоты при нагревании?" to "q=cmΔt",

"Что такое удельная теплоемкость?" to "количество теплоты для нагрева 1 кг вещества на 1 градус",

"Какие бывают способы теплопередачи?" to "теплопроводность, конвекция, излучение",

"Какой агрегатный переход сопровождается поглощением теплоты?" to "плавление",

"Что такое испарение?" to "переход жидкости в пар с поверхности",

"От чего зависит внутренняя энергия?" to "от температуры и агрегатного состояния",

"Формула количества теплоты при сгорании топлива?" to "q=qm",

"Что происходит при конденсации?" to "пар превращается в жидкость"

)

"electricity" -> listOf(

"Формула закона Ома для участка цепи?" to "i=u/r",

"Единица измерения силы тока?" to "ампер",

"Что такое напряжение?" to "работа по перемещению заряда",

"Что такое сопротивление?" to "характеристика проводника",

"Какие части есть в электрической цепи?" to "источник, проводник, нагрузка",

"Формула мощности тока?" to "p=ui",

"Какое напряжение в розетке?" to "220",

"Что такое ток?" to "упорядоченное движение заряженных частиц",

"Из чего складывается полное сопротивление?" to "из суммы всех сопротивлений",

"Что показывает амперметр?" to "силу тока"

)

"optics" -> listOf(

"Что изучает оптика?" to "свет и его свойства",

"Как называется отражение от зеркала?" to "зеркальное",

"Какой закон отражения света?" to "угол падения равен углу отражения",

"Что такое преломление?" to "изменение направления света на границе сред",

"Что такое линза?" to "прозрачное тело, ограниченное сферическими поверхностями",

"Какая линза собирает свет?" to "выпуклая",

"Как строится изображение в линзе?" to "с помощью главной оси и фокусов",

"Что такое фокусное расстояние?" to "расстояние от центра линзы до фокуса",

"Какой прибор увеличивает предметы?" to "лупа",

"Как происходит дисперсия света?" to "разложение на спектр"

)

"atom" -> listOf(

"Что такое атом?" to "наименьшая частица вещества",

"Из чего состоит атом?" to "ядро и электроны",

"Какие частицы в ядре?" to "протоны и нейтроны",

"Что такое изотопы?" to "атома с разным числом нейтронов",

"Как обозначается заряд электрона?" to "минус",

"Что такое радиоактивность?" to "самопроизвольный распад ядер",

"Какие бывают виды радиоактивного излучения?" to "альфа, бета, гамма",

"Что такое деление ядра?" to "распад с выделением энергии",

"Где используется ядерная энергия?" to "в атомных электростанциях",

"Что такое цепная реакция?" to "последовательное деление ядер"

)

else -> listOf("Нет вопросов" to "")

}

}

private fun showQuestion() {

if (currentQuestionIndex < questions.size) {

val (question, \_) = questions[currentQuestionIndex]

questionText.text = "Вопрос ${currentQuestionIndex + 1}: $question"

answerInput.text.clear()

} else {

showResult()

}

}

private fun checkAnswer() {

val userAnswer = answerInput.text.toString().trim().lowercase()

val correctAnswer = questions[currentQuestionIndex].second.lowercase()

if (userAnswer == correctAnswer) {

correctAnswers++

}

currentQuestionIndex++

showQuestion()

}

private fun showResult() {

AlertDialog.Builder(this)

.setTitle("Результат")

.setMessage("Правильных ответов: $correctAnswers из ${questions.size}")

.setPositiveButton("OK") { dialog, \_ ->

dialog.dismiss()

finish()

}

.setCancelable(false)

.show()

}

}

## ПРИЛОЖЕНИЕ H. Программа, файл – TestBlockActivity.kt

package com.example.vkr

import android.content.Intent

import android.os.Bundle

import android.widget.Button

import androidx.appcompat.app.AppCompatActivity

class TestBlockActivity : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_test\_block)

findViewById<Button>(R.id.buttonMechanics).setOnClickListener {

openTest("mechanics")

}

findViewById<Button>(R.id.buttonThermo).setOnClickListener {

openTest("thermodynamics")

}

findViewById<Button>(R.id.buttonElectricity).setOnClickListener {

openTest("electricity")

}

findViewById<Button>(R.id.buttonOptics).setOnClickListener {

openTest("optics")

}

findViewById<Button>(R.id.buttonAtomic).setOnClickListener {

openTest("atomic")

}

findViewById<Button>(R.id.buttonBack).setOnClickListener {

finish()

}

}

private fun openTest(topic: String) {

val intent = Intent(this, TestActivity::class.java)

intent.putExtra("TOPIC\_KEY", topic)

startActivity(intent)

}

}

## ПРИЛОЖЕНИЕ I. Программа, файл – TestBlockSelectActivity.kt

package com.example.vkr

import android.content.Intent

import android.os.Bundle

import android.widget.Button

import androidx.appcompat.app.AppCompatActivity

class TestBlockSelectActivity : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_test\_block\_select)

val buttonMechanics = findViewById<Button>(R.id.buttonMechanics)

val buttonThermo = findViewById<Button>(R.id.buttonThermo)

val buttonElectricity = findViewById<Button>(R.id.buttonElectricity)

val buttonOptics = findViewById<Button>(R.id.buttonOptics)

val buttonAtomic = findViewById<Button>(R.id.buttonAtomic)

val buttonBack = findViewById<Button>(R.id.buttonBack)

buttonMechanics.setOnClickListener {

// Переход к тестам по механике

}

buttonThermo.setOnClickListener {

// Переход к тестам по термодинамике

}

buttonElectricity.setOnClickListener {

// Переход к тестам по электричеству

}

buttonOptics.setOnClickListener {

// Переход к тестам по оптике

}

buttonAtomic.setOnClickListener {

// Переход к тестам по атомной физике

}

buttonBack.setOnClickListener {

finish()

}

}

}

## ПРИЛОЖЕНИЕ J. Программа, файл – TheoryBlockSelectActivity.kt

package com.example.vkr

import android.content.Intent

import android.os.Bundle

import android.widget.Button

import androidx.appcompat.app.AppCompatActivity

class TheoryBlockSelectActivity : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_theory\_block\_select)

val buttonMechanics = findViewById<Button>(R.id.buttonMechanics)

val buttonThermo = findViewById<Button>(R.id.buttonThermo)

val buttonElectricity = findViewById<Button>(R.id.buttonElectricity)

val buttonOptics = findViewById<Button>(R.id.buttonOptics)

val buttonAtomic = findViewById<Button>(R.id.buttonAtomic)

val buttonBack = findViewById<Button>(R.id.buttonBack)

buttonMechanics.setOnClickListener {

openTheoryContent("mechanics")

}

buttonThermo.setOnClickListener {

openTheoryContent("thermodynamics")

}

buttonElectricity.setOnClickListener {

openTheoryContent("electricity")

}

buttonOptics.setOnClickListener {

openTheoryContent("optics")

}

buttonAtomic.setOnClickListener {

openTheoryContent("atomic")

}

buttonBack.setOnClickListener {

finish()

}

}

private fun openTheoryContent(blockName: String) {

val intent = Intent(this, TheoryContentActivity::class.java)

intent.putExtra("block", blockName)

startActivity(intent)

}

}

## ПРИЛОЖЕНИЕ K. Программа, файл–TheoryBlockSelectionActivity.kt

package com.example.vkr

import android.content.Intent

import android.os.Bundle

import android.widget.Button

import androidx.appcompat.app.AppCompatActivity

class TheoryBlockSelectionActivity : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_theory\_block\_selection)

findViewById<Button>(R.id.buttonMechanics).setOnClickListener {

openTheory("mechanics")

}

findViewById<Button>(R.id.buttonThermodynamics).setOnClickListener {

openTheory("thermodynamics")

}

findViewById<Button>(R.id.buttonElectricity).setOnClickListener {

openTheory("electricity")

}

findViewById<Button>(R.id.buttonOptics).setOnClickListener {

openTheory("optics")

}

findViewById<Button>(R.id.buttonAtomic).setOnClickListener {

openTheory("atomic")

}

findViewById<Button>(R.id.buttonBack).setOnClickListener {

finish() // возвращение на главный экран

}

}

private fun openTheory(topic: String) {

val intent = Intent(this, TheoryContentActivity::class.java)

intent.putExtra("TOPIC\_KEY", topic)

startActivity(intent)

}

}

## ПРИЛОЖЕНИЕ L. Программа, файл – TheoryContentActivity.kt

package com.example.vkr

import android.os.Bundle

import android.widget.Button

import android.widget.TextView

import androidx.appcompat.app.AppCompatActivity

class TheoryContentActivity : AppCompatActivity() {

private lateinit var textViewContent: TextView

private lateinit var buttonPrev: Button

private lateinit var buttonNext: Button

private var currentPage = 0

private var pages: List<String> = listOf()

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_theory\_content)

textViewContent = findViewById(R.id.textViewTheory)

buttonPrev = findViewById(R.id.buttonPrev)

buttonNext = findViewById(R.id.buttonNext)

val topic = intent.getStringExtra("TOPIC\_KEY") ?: ""

pages = getTheoryPages(topic)

updateText()

buttonPrev.setOnClickListener {

if (currentPage > 0) {

currentPage--

updateText()

}

}

buttonNext.setOnClickListener {

if (currentPage < pages.size - 1) {

currentPage++

updateText()

}

}

findViewById<Button>(R.id.buttonBack).setOnClickListener {

finish()

}

}

private fun updateText() {

textViewContent.text = pages[currentPage]

}

private fun getTheoryPages(topic: String): List<String> {

return when (topic) {

"mechanics" -> listOf(

"Механика — раздел физики, изучающий механическое движение тел и причины, его вызывающие.\n\nФормулы:\nV = S / t\nF = ma\nP = mg",

"Виды движения:\n— Равномерное\n— Неравномерное\n— Прямолинейное\n— Криволинейное",

"Закон Ньютона:\n1. Инерция\n2. F = ma\n3. Действие = Противодействие"

)

"thermodynamics" -> listOf(

"Молекулярная физика и термодинамика изучают строение вещества и тепловые процессы.\n\nТемпература, внутренняя энергия, агрегатные состояния.",

"Формулы:\nQ = cmΔT\nQ = Lm\nПервый закон термодинамики: ΔU = Q - A"

)

"electricity" -> listOf(

"Электричество и магнетизм — раздел физики, изучающий электрические и магнитные явления.",

"Формулы:\nI = q / t\nU = IR\nP = UI\nA = UIt"

)

"optics" -> listOf(

"Оптика — раздел физики, изучающий световые явления, отражение, преломление, линзы.",

"Формулы:\n1/f = 1/d + 1/F\nn = sin(i)/sin(r)"

)

"atomic" -> listOf(

"Атомная и ядерная физика изучают строение атомов, радиоактивность, ядерные реакции.",

"Основные понятия: альфа-, бета-, гамма-излучения, деление ядер, масса дефекта, формула Эйнштейна: E = mc²"

)

else -> listOf("Ошибка загрузки теории.")

}

}

}

## ПРИЛОЖЕНИЕ M. Программа, файл – TheoryData.kt

package com.example.vkr

object TheoryData {

val mechanics = listOf(

"""

\*\*Механика — раздел физики\*\*, изучающий механическое движение тел и причины этого движения.

Механическое движение — изменение положения тела в пространстве относительно других тел с течением времени.

Основные понятия: материальная точка, траектория, путь, перемещение, скорость, ускорение.

""".trimIndent(),

"""

\*\*Основные законы механики:\*\*

• Первый закон Ньютона: существует инерциальная система отсчёта, в которой тело сохраняет состояние покоя или равномерного прямолинейного движения, если на него не действуют силы.

• Второй закон Ньютона: F = ma — ускорение тела прямо пропорционально силе и обратно пропорционально массе.

• Третий закон Ньютона: силы, с которыми два тела действуют друг на друга, равны по модулю и противоположны по направлению.

""".trimIndent(),

"""

\*\*Формулы механики:\*\*

• Ускорение: a = (v - v₀)/t

• Путь при равномерном движении: s = vt

• Путь при равноускоренном движении: s = v₀t + (at²)/2

• Скорость: v = s/t

• Импульс: p = mv

• Сила тяжести: F = mg

• Работа: A = F·s·cos(α)

• Кинетическая энергия: Ek = (mv²)/2

• Потенциальная энергия: Ep = mgh

• Закон сохранения энергии: Eполн = const

""".trimIndent()

)

val thermodynamics = listOf(

"""

\*\*Молекулярная физика и термодинамика\*\*

Молекулярная физика изучает строение вещества и его свойства, исходя из представлений о молекулах и атомах.

Все тела состоят из молекул, между которыми существуют силы взаимодействия. Частицы движутся хаотически, обладают кинетической энергией.

""".trimIndent(),

"""

\*\*Основные физические величины:\*\*

• Температура — мера средней кинетической энергии молекул

• Давление — сила, действующая перпендикулярно на единицу площади: P = F/S

• Внутренняя энергия — сумма кинетической и потенциальной энергии всех молекул тела

• Теплоёмкость: Q = cmΔT

""".trimIndent(),

"""

\*\*Первый закон термодинамики:\*\*

ΔU = Q - A

Изменение внутренней энергии тела равно количеству теплоты, переданному телу, минус работа, совершённая телом над внешними телами.

\*\*Формы передачи энергии:\*\*

• Теплопроводность

• Конвекция

• Излучение

""".trimIndent()

)

val electricity = listOf(

"""

\*\*Электричество и магнетизм\*\*

Электрический ток — упорядоченное движение заряженных частиц. Условия существования тока: наличие свободных носителей заряда и электрического поля.

Напряжение: U = A/q

Сила тока: I = q/t

Сопротивление: R = U/I

""".trimIndent(),

"""

\*\*Закон Ома для участка цепи:\*\*

I = U / R

\*\*Сопротивление проводника:\*\*

R = ρl / S

где ρ — удельное сопротивление, l — длина, S — площадь поперечного сечения.

\*\*Мощность тока:\*\*

P = IU

Количество теплоты: Q = I²Rt

""".trimIndent(),

"""

\*\*Магнитное поле:\*\*

Возникает вокруг движущихся зарядов.

Действует сила Лоренца: F = qvBsin(α)

Правило правой руки помогает определить направление силы.

Электромагнитная индукция: ЭДС индукции = -ΔΦ/Δt

Закон Ленца: индукционный ток всегда направлен так, чтобы препятствовать изменению магнитного потока.

""".trimIndent()

)

val optics = listOf(

"""

\*\*Оптика — раздел физики, изучающий свет, его распространение и взаимодействие с веществом.\*\*

Свет — электромагнитная волна. В вакууме скорость света: c = 3·10⁸ м/с.

Закон прямолинейного распространения света: в однородной среде свет распространяется прямолинейно.

""".trimIndent(),

"""

\*\*Отражение и преломление света:\*\*

Закон отражения: угол падения = угол отражения.

Закон преломления: n₁sin(α) = n₂sin(β), где n — показатель преломления.

Показатель преломления: n = c / v

(v — скорость света в среде)

""".trimIndent(),

"""

\*\*Линзы:\*\*

• Собирающие и рассеивающие

• Фокусное расстояние: f — расстояние от центра линзы до фокуса

• Формула тонкой линзы: 1/f = 1/d + 1/f′

• Увеличение: Γ = h′ / h = f′ / f

""".trimIndent()

)

val atomic = listOf(

"""

\*\*Атомная и ядерная физика\*\*

Атом — система, состоящая из ядра и электронов.

Ядро — протоны и нейтроны. Заряд ядра = Z·e.

Модель атома Бора: электроны движутся по стационарным орбитам. При переходе между уровнями испускается или поглощается квант энергии.

""".trimIndent(),

"""

\*\*Явления:\*\*

• Фотоэффект — выбивание электронов светом

• Радиоактивность — самопроизвольное превращение атомных ядер

• Альфа-, бета-, гамма-излучение

Закон радиоактивного распада: N = N₀·e^(-λt)

""".trimIndent(),

"""

\*\*Единицы измерения:\*\*

• 1 эВ (электронвольт) — энергия, которую приобретает электрон, проходя через разность потенциалов 1 В:

1 эВ = 1.6·10⁻¹⁹ Дж

• Активность вещества: Бк (Беккерель) — 1 распад в секунду

• Доза излучения: Зиверт, Грей

"""

.trimIndent()

)

}

xml файлы доделать