

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ «ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНО-ЭКОНОМИЧЕСКИЙ КОЛЛЕДЖ»

Специальность 2-40 01 01 «Программное обеспечение

информационных технологий»

**КУРСОВОЙ ПРОЕКТ**

по дисциплине: «Основы алгоритмизации и программирования»

**на тему: «»**

Пояснительная записка

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Разработал |  |  |  |  | Батюк Е. А. |
|  | (дата) |  | (подпись) |  | (ФИО) |
| Руководитель |  |  |  |  | Глушенок А. В. |
|  | (дата) |  | (подпись) |  | (ФИО) |

Гомель, 2022

СОДЕРЖАНИЕ

[Введение 3](#_Toc101636964)

[Простые типы данных 4](#_Toc101636965)

# Введение

Цель курсового проекта – разработка мобильного приложения «».

Мобильное приложение

Приложение «Физика » включает в себя повторение пройденного материала, также изучение нового.

В основном, учебными пособиями по предмету пользуются как справочным материалом для подготовки домашнего задания и к проверочным работам. Разрабатываемая система даёт возможность не затрачивать дополнительного времени на поиск нужной информации и проводить самоконтроль и выявление слабых мест в изученном материале.

Задачи:

1. Создание краткого справочного материала по предмету «Физика»;
2. создание тестов самоконтроля.

Этапы создания системы:

1. Изучение учебного пособия, на основе которых создается справочный материал;
2. составление тестов самоконтроля по определённым темам;
3. изучение методов и алгоритмов языка программирования «Dart» в области создания мобильных приложений;
4. создание страницы поиска нужного материала;
5. разработка методов показа информации;
6. разработка методов прохождения тестов самоконтроля;
7. разработка интерфейса.

Возможности приложения:

* Просматривать справочный материал по темам;
* просматривать формулы по классам;
* просматривать таблицы значений;
* проходить тесты самоконтроля по темам;
* просматривать статистику прохождения тестов;
* просматривать историю пройденных тем;
* просматривать историю;
* просматривать статистику исходя из результатов пройденных тестов.

По итогу разработки системы будет получено приложение, которое несёт в себе краткую выдержку из тем, находящиеся в учебном пособии, тесты самоконтроля, формулы и таблицы.

# Простые типы данных

Тип данных — множество значений и операций над этими значениями.

Типизация — операция назначения типа информационным сущностям.

В языке Object Pascal существуют следующие простые типы данных:

* целочисленный;
* вещественный;
* символьный;
* строковый;
* логический.

Целочисленные типы содержат в себе значения, интерпретируемые как числа (знаковые и беззнаковые).

На рисунке 1.1. представлен таблица целочисленных типов

|  |  |  |
| --- | --- | --- |
| **Тип** | **Диапазон** | **Формат** |
| shortint | -128 ..+127 | 8 битов со знаком |
| integer | -32768 .. 32767 | 16 битов со знаком |
| longint | -2147483648 +2147483647 | 32 бита со знаком |
| byte | 0 .. 255 | 8 битов без знака |
| word | 0 .. 65535 | 16 битов без знака |

Рисунок 1.1 – Таблица всех целочисленных

На рисунке 1.2. представлен пример описания целочисленной переменной типа «integer».

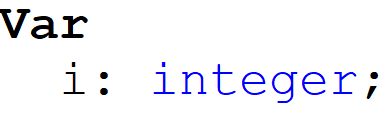


Рисунок 1.2 – Объявление переменной типа «Integer»

Над целочисленными переменными проводятся все математические операции, а также операции целочисленного деления.

К вещественному типу относится подмножество вещественных чисел, которые могут быть представлены в формате с плавающей запятой с фиксированным числом цифр.

На рисунке 1.2. представлена таблица переменных вещественных типов.

|  |  |  |
| --- | --- | --- |
| **Тип** | **Диапазон** | **Цифры** |
| Real  Single  Duble  Extende  comp | 2.9x10Е-39 до 1.7x10Е 38  1.5x10Е-45 до 3.4x10Е 38  5.0x10Е-324 до 1.7x10Е 308  3.4x10Е-493 до 1.1x10Е 403  -2Е 63 до 2Е 63 | от 11 до 12  от 7 до 8  от 15 до 16  от 19 до 20  от 19 до 20 |

Рисунок 1.2 – Таблица переменных вещественных типов

На рисунке 1.3. представлен пример описания вещественной переменной типа «real».

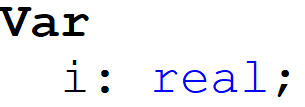


Рисунок 1.3 – Объявление переменной типа «Real»

Над вещественными переменными проводятся все математические операции, а также операции отбрасывания дробной части (TRUNC(x)) и округления (ROUND(x)).

Символьный тип данных - тип данных, предназначенный для хранения одного символа (управляющего или печатного) в определённой кодировке.

При описании символьной переменной используется зарезервированное слово «char», пример представлен на рисунке 1.4.

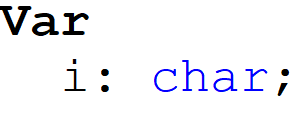


Рисунок 1.4 – Объявление переменной типа «Char»

Порядковые номера букв также упорядочены по возрастанию, но не обязательно следуют друг за другом, представлены на рисунке 1.5

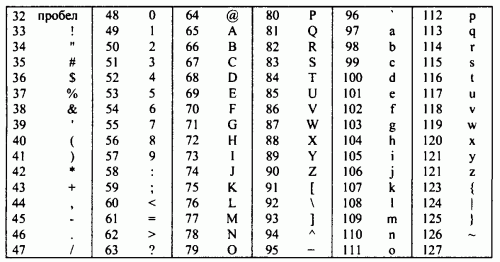


Рисунок 1.5 – Таблица символов ASCII

Функции, которые применимы к символьным переменным:

* «ORD(X)» - определяет порядковый номер символа «Х»;
* CHR(X) - определяет символ по номеру;
* PRED(X) - выдает символ, стоящий перед символом Х;
* SUCC(X) - выдает символ, следующий после символа Х.

Строка– последовательность из определенного количества символов.

Объявление типа «string» показаны на рисунке 1.6

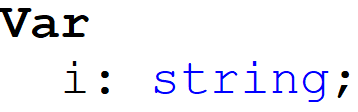


Рисунок 1.6 – Объявление переменной типа «string»

Процедуры обработки строк:

* Delete(St, Poz, N) – удаление N символов строки St, начиная с позиции Poz;
* Insert(Str1, Str2, Poz) – вставка строки Str1 в строку Str2, начиная с позиции Poz;
* Str(I, Srt) – преобразование числового значения величины I (целого или вещественного типа) и помещение результата в строку Str.
* Val(Str, I, Cod) – преобразует значение Str в величину целочисленного или вещественного типа и помещает результат в I. Функции обработки строк:
* Copy(Str, Poz, N) – выделяет из строки St подстроку длинной N символов, начиная с позиции Poz.
* Concat(Str1, Srt2, …, StrN) – выполняет сцепление строк Str1, Srt2, …, StrN в том порядке, в каком они указаны в списке параметров.
* Length(Str) – вычисляет длину в символах строки Str;
* Pos(Str1, Str2) – обнаруживает первое появление в строке Str2 подстроки Str1.

Логический тип данных – первоначальный тип данных в информатике, который принимают два значения переменной, определяемой истиной (true) и ложью (false)

# ОПИСАНИЕ АЛГОРИТМОВ И ТЕХНОЛОГИИ РЕШЕНИЯ

Приложение должно иметь удобный и понятный для пользователя интерфейс, приятный стиль оформления и рабочий функционал.

В приложении должны присутствовать следующие функции:

* просмотр материала;
* фильтрация материала по классам и разделам;
* прохождение тестов самоконтроля;
* просмотр формул, отфильтрованных по классам;
* просмотр таблиц значений.

Данные приложения должны храниться в файлах. В курсовом проекте используются файлы формата «.dart».

На главной форме пользователю предоставляется информация об его изучении материала. Ниже от статистики пользователя располагается список выбора, предоставляемого материала разбитый по классам.

После выбора класса, пользователю будет предоставлен выбор теории или формул.

После выбора теории, пользователь может выбрать нужный ему раздел. Далее в текстовой и графическом формате, ему будет представлена информация материала.

После выбора формул, пользователю будет предоставлены формулы в графической и текстовой форме.

После прохождения каждого раздела пользователю будет представлена возможность пройти тест по пройденному материалу.

Форма – это место, где пользователь общается с программой. Приложение может иметь несколько форм, каждая из которых выполняет свое особое предназначение.

Главная форма – это основная форма, с которой начинается построение программы.

Макет графического интерфейса начальной формы представлен на рисунке 2.1.

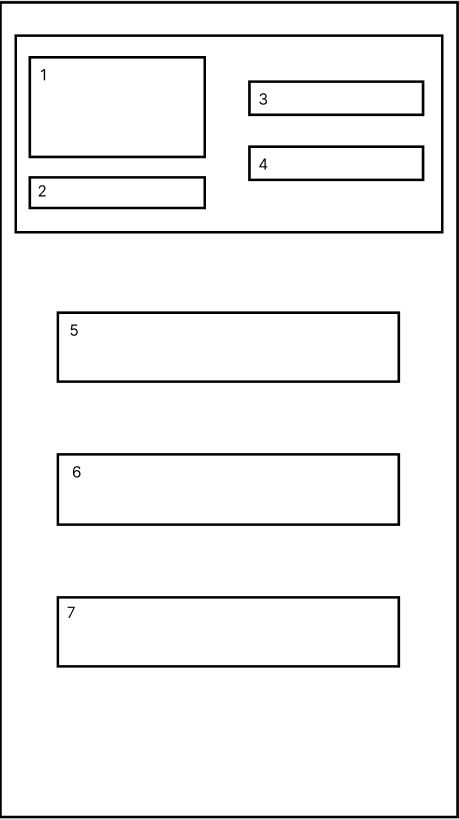


Рисунок 2.1 – Главная форма приложения. 1 – имя пользователя; 2 – средняя оценка по пройденным тестам; 3 – история пройденных тем; 4 – история пройденных тестов; 5 – материал по 10 классу; 6 – материал по 11 классу; 7 – список таблиц значений.

Макет графического интерфейса формы выбора внутри класса представлен на рисунке 2.2.

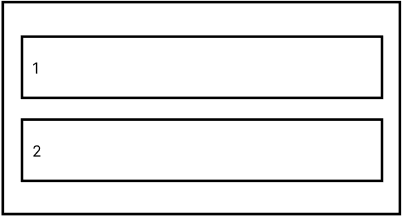


Рисунок 2.2 – Форма выбора внутри выбранного класса. 1 – Список формул; 2 – Список материала по выбранному классу.

Макет графического интерфейса формы показа материала представлен на рисунке 2.3.

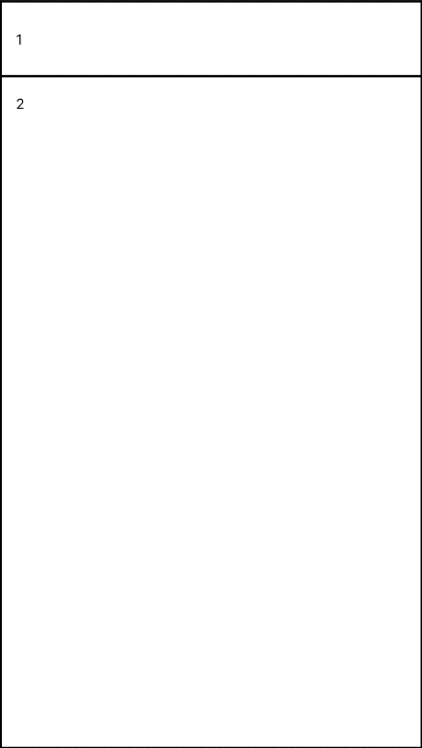


Рисунок 2.3 – Форма показа материала. 1 – название раздела; 2 – текст материала.

Макет графического интерфейса формы прохождения тестов представлен на рисунке 2.4.

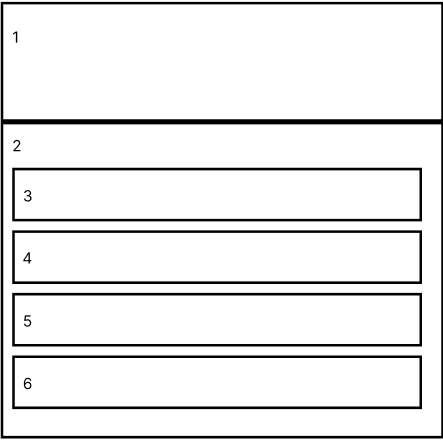


Рисунок 2.4 – Форма прохождения тестов. 1 – Вопрос теста; 2 – Список выбираемого ответа; 3,4,5,6 – варианты выбора.

# 3 ИСПОЛНЕНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ

## 3.1 Описание интерфейса

Интерфейс — это совокупность средств и методов предназначенных для взаимодействия элементов системы (или целых систем) между собой.

Графический пользовательский интерфейс — это интерфейс, который обеспечивает передачу информации между пользователем и программно-аппаратными компонентами компьютерной системы.

При открытии приложения, показывается начальное окно. Начальным окном является форма, где располагается список доступного материала, разбитый по классам и разделам.

Начальная форма представлена на рисунке 3.1.



Рисунок 3.1 – Главная форма приложения

При нажатии на атрибут списка выбора, появляется список доступного материала, разбитый на разделы.

Список материалов представлен на рисунке 3.2.

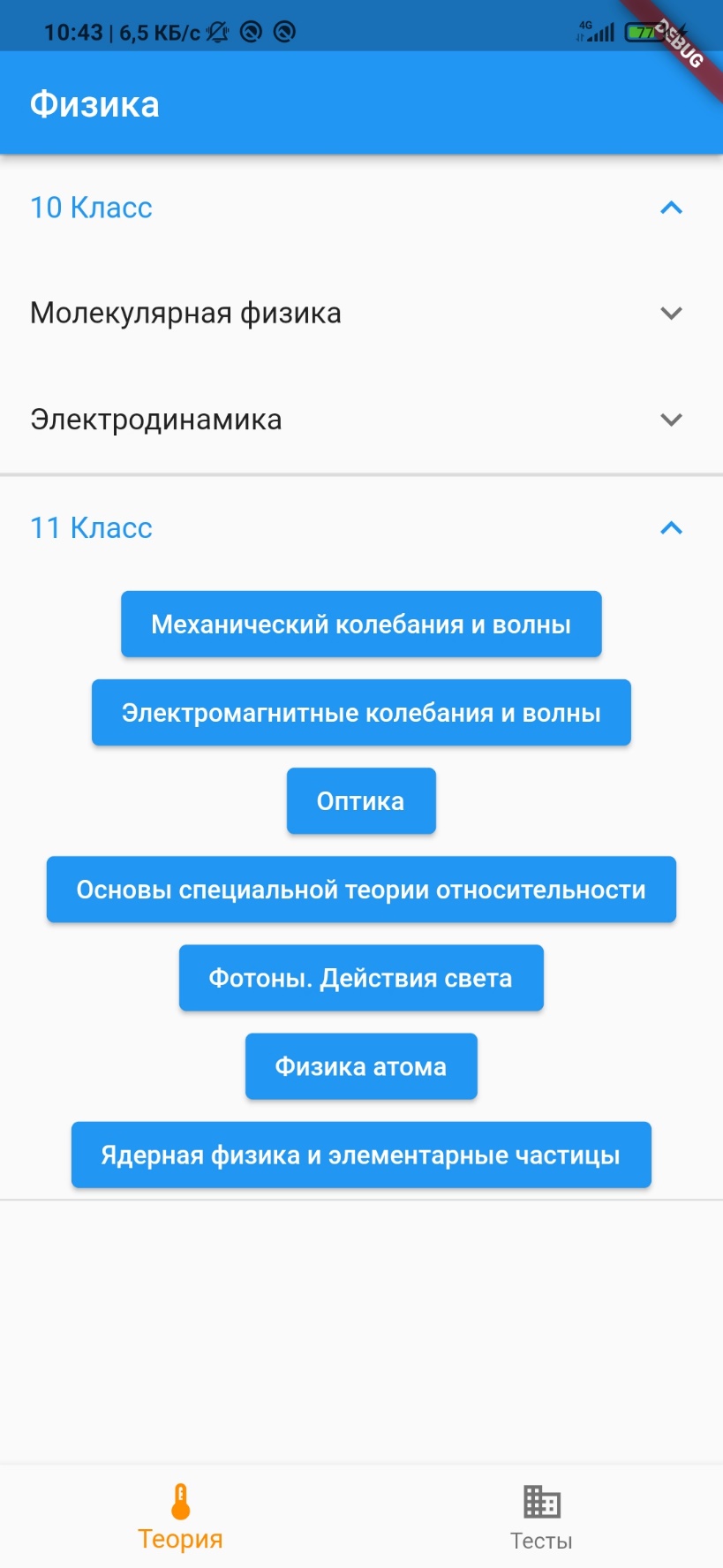


Рисунок 3.2 – Форма выбора материала

По нажатию на кнопку с названием раздела, открывается страница с выбранным материалом.

На рисунке 3.3 представлена форма показа материала.

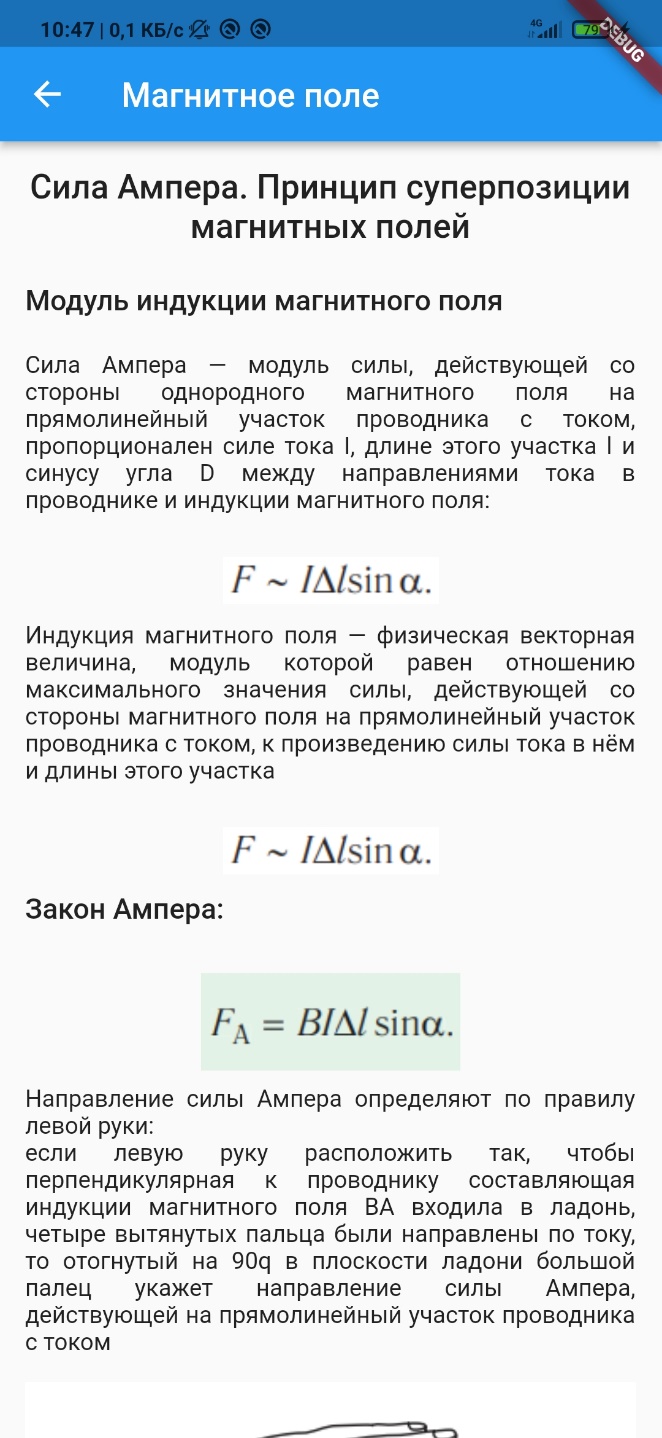


Рисунок 3.3 – Форма показа материала

Просмотр определенного параграфа происходит путем пролистывания формы вниз. Переход по параграфам происходит путем свайпа влево или вправо.

По нажатию на кнопку тесты, снизу экрана приложения, открывается страница со списком тестов.

Форма просмотра списка тестов представлен на рисунке 3.4.

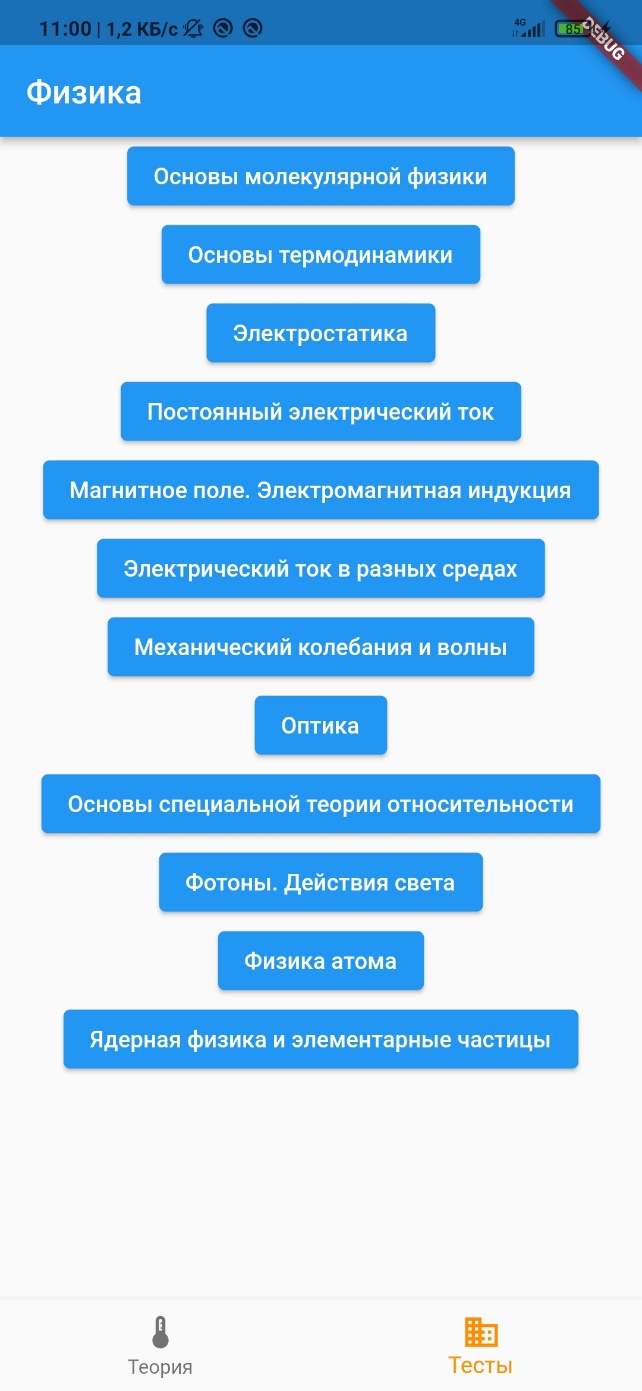


Рисунок 3.4 – Форма показа списка тестов

При нажатии кнопки с желаемой темой прохождения теста, открывается окно с прохождением теста. Для начала прохождения теста, необходимо сверху нажать на кнопку «Начать тест». Правильный ответ помечается круглой галочкой напротив ответа. Для проверки правильности ответа следует нажать на кнопку «Проверить»

Форма прохождения теста представлена на рисунке 3.5.

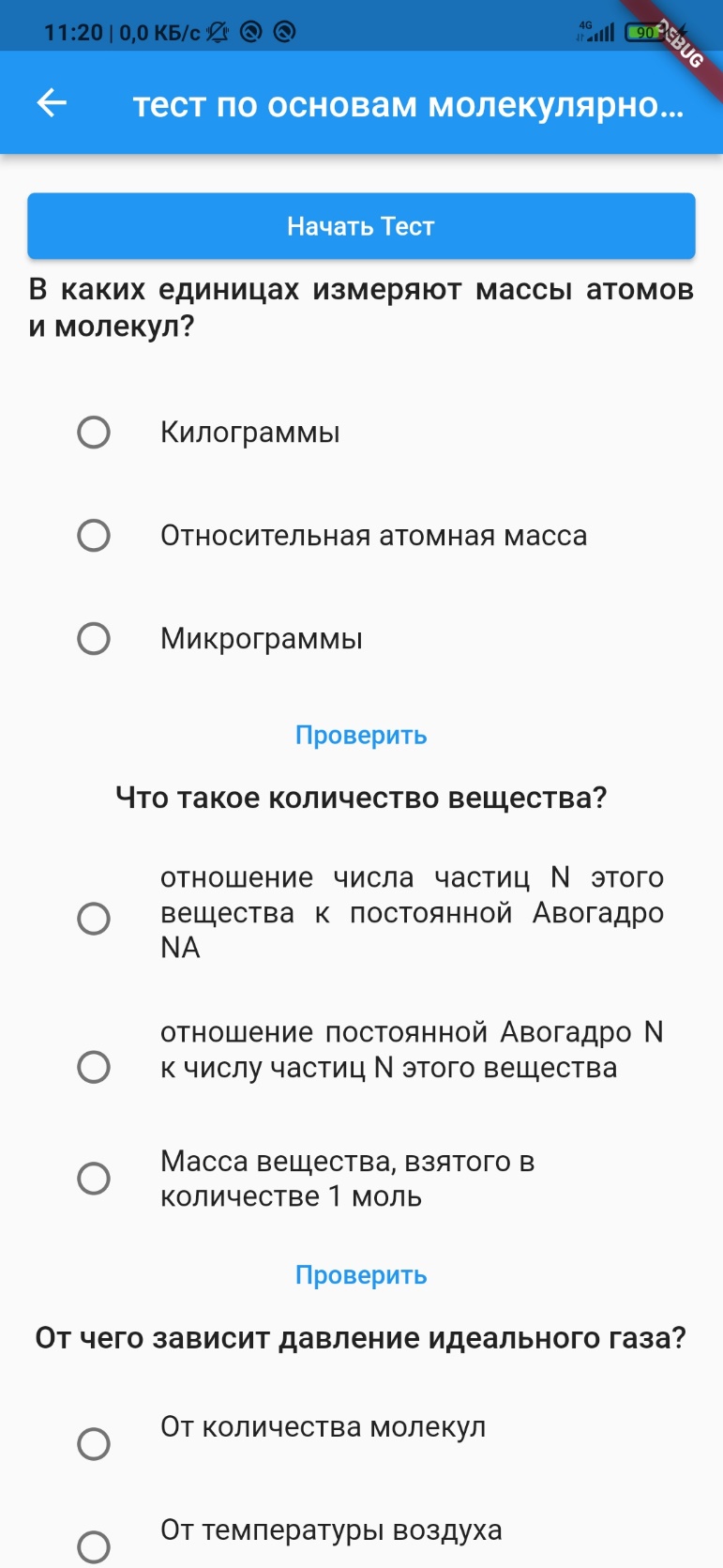


Рисунок 3.5 – Форма прохождения теста

Для просмотра числа правильных ответов, необходимо нажать на кнопку «Результат». После нажатия кнопки, откроется форма с числом правильных ответов. При нажатии на кнопку «На главную», пользователь перенаправляется на главную страницу.

Форма показа числа правильных ответов представлена на рисунке 3.6.



Рисунок 3.6 – Форма показа числа правильных ответов