

**ОЛИМПИАДА ШКОЛЬНИКОВ «ШАГ В БУДУЩЕЕ»**

**НАУЧНО-ОБРАЗОВАТЕЛЬНОЕ СОРЕВНОВАНИЕ «ШАГ В БУДУЩЕЕ, МОСКВА»**

ШМ0508

*регистрационный номер*

Информатика и системы управления (ИУ)

*название факультета*

Программное обеспечение ЭВМ и информационные технологии (ИУ-7)

*название кафедры*

Разработка образовательной программы

*название работы*

**Автор:**

Иванов Всеволод Алексеевич  
*фамилия, имя, отчество*

ГБОУ Инженерная школа № 1581 города Москвы, 11 Г  
*наименование учебного заведения, класс*

**Научный руководитель:**

\_\_\_\_\_  
*фамилия, имя, отчество*

\_\_\_\_\_  
*место работы*

\_\_\_\_\_  
*звание, должность*

\_\_\_\_\_  
*подпись научного руководителя*

**Москва - 2018**

## СОДЕРЖАНИЕ:

Введение	2
1. Аналитическая часть	3
2. Конструкторская часть	5
2.1. Типы данных	5
2.2. Теоретический материал	5
2.2.1. Задачи	5
2.2.2. Тесты	6
2.3. Типы графических интерфейсов	6
2.3.1. Главное меню	6
2.3.2. Выбор темы	6
2.3.3. Теоретический материал	7
2.3.4. Задачи	7
2.3.5. Тесты	7
3. Технологическая часть	8
3.1. Используемое программное обеспечение	8
3.2. Описание функциональных окон	8
3.2.1. Меню	8
3.2.2. Выбор темы	9
3.2.3. Выбор темы (режим редактирования)	9
3.2.4. Редактирование теоретического материала	10
3.2.5. Редактирование задач	11
3.2.6. Редактирование тестов	12
3.2.7. Просмотр теории	13
3.2.8. Просмотр задач	14
3.2.9. Решение тестов	15
3.2.10. Просмотр результатов теста	16
3.3. Особенности реализации	17
Заключение	18
Ссылки	18

## **Введение**

В наше время большую роль в образовательной системе играют информационные технологии. Существуют электронные задачки, сетевые учебники с внушительными наборами библиотек. Что нового можно предложить на этом, казалось бы, освоенном пространстве?

Образование всегда являлось одной из самых актуальных тем в жизни человека. На данный момент в образовательную систему активно внедряются информационные технологии (например, электронный дневник МРКО, ставший заменой бумажному). Однако, до сих пор является актуальным вопрос о технологии, позволяющей пользователю свободно формировать учебные пособия для других пользователей. Так, информационные технологии могут помочь репетиторам и преподавателям, использующим собственные наборы задач теоретического материала. Данный проект предлагает новый взгляд на программу для обучения. Новизна состоит прежде всего в том, что формирование и редактирование контента полностью находится в руках пользователя.

В образовательном процессе немаловажно закрепление пройденного материала. Большинство образовательного ПО содержит только теоретический материал, либо практические тесты. Данная программа создаёт возможность тесно связывать теоретический материал и задания для проверки знаний пользователя. Данная функция позволит пользователю, изучившему теоретический материал, сразу проверить свои знания с помощью задач или тестов. И, наоборот, в случае непонимания пользователем тестирующего материала существует возможность восполнить «пробелы» в знаниях, изучив теоретический материал.

Целью данного проекта является создание программы, позволяющей просматривать и изменять учебный материал в доступной для любого пользователя форме.

В процессе разработки программы были поставлены следующие задачи:

- Изучение общей структуры теоретического материала, задач, тестов для

создания обобщённой формы их представления;

- Создание и наполнение структуры для хранения учебного материала;
- Создание редактора структуры данных для добавления, изменения и удаления учебного материала;
- Создание алгоритма отображения учебного материала.

## **1. Аналитическая часть**

Основной целью проекта является создание образовательной программы с возможностью редактирования материала. Для решения данной задачи в основном используются печатные пособия. Поэтому, для определения требований к программе следует проанализировать основные особенности материала и его структуры, а также преимущества и недостатки печатных пособий.

Неотъемлемой частью любого учебника является наличие изображений. Они помогают читателю лучше понимать материал, или вовсе несут главную его суть. Программа должна обладать графическим интерфейсом, позволяющим использовать изображения. Также структура оформления материала в программе должна позволять использовать текст и изображения вместе.

Для представления теоретического материала, в основном, используется текст, комбинированный с изображениями (изображения находятся непосредственно в самом тексте для удобства его восприятия). Следовательно, для создания статьи необходимо использовать текст и изображения с указанием их места в тексте.

Для представления задачи необходимы условие и варианты ответов. Условие может быть представлено и текстовым форматом, и графическим, и их комбинацией. В отличие от представления теоретического материала, нет необходимости строгой привязки рисунка и текста. Следовательно, должны быть продуманы метки принадлежности данных и по их сути, и по их форме.

Для представления вопросов из тестов используются текст и изображения.

Так же, как и в задачах, изображения не требуют привязки к тексту. Сами вопросы делятся на вопросы типа «А», в которых предоставляется несколько вариантов ответа, из которых нужно выбрать один правильный, и вопросы типа «В», в которых нужно дать краткий ответ. Существуют также и вопросы типа «С», для которых требуются развёрнутый ответ, но ответы на такие вопросы может проверять только педагог. В некоторых случаях, в вопросах типа «А» вариантами ответа являются изображения (в случаях, когда вариантами ответа являются графики и т.п.). Следовательно, для создания вопроса из теста необходимо определить его условие, тип, изображения, связанные с условием, варианты ответа или вариант, при этом должна существовать возможность использовать в качестве варианта ответа изображение.

Для всех вышеперечисленных типов образовательного материала требуется определение принадлежности к изучаемой теме и взаимосвязи практического материала с теоретическим, в том числе графическим. Для хранения всего учебного материала требуется создание базы данных на основе предварительно разработанной структуры.

Одним из недостатков традиционных учебных пособий является отсутствие возможности редактирования материала, что делает невозможным изменение уже написанного материала. Следовательно, в программе должна быть реализована возможность свободного редактирования всего материала, размещённого в базе данных.

Также в традиционных учебных пособиях отсутствует возможность связывать теоретический материал с задачами или тестами. Как правило, они содержат в себе либо только теорию, либо только задачи. Программа должна иметь возможность тесно связывать теоретический материал и задания для проверки знаний пользователя. Данная функция позволит пользователю, изучившему теоретический материал, сразу проверить свои знания с помощью задач или тестов. И наоборот, в случае непонимания пользователем тестирующего материала существует возможность восполнить «пробелы» в знаниях, изучив теоретический материал.

Существенным недостатком традиционных учебных пособий является сложность их создания. Так, помимо написания самого материала, требуется произвести его оформление и печать. Следовательно, программа должна использовать общие принципы в оформлении учебного материала, и компоновать данные по ним. Таким образом, оформление данных становится функцией программы.

Опираясь на данные рассуждения, можно сделать вывод о том, что для представляемой программе требуется:

- Создать базу данных для хранения учебного материала;
- Разработать алгоритм отображения данных;
- Разработать структуры базы данных;
- Создать возможность использования изображений в оформлении материала;
- Создать возможность связывать материал при помощи ссылок.

## **2. Конструкторская часть**

### **2.1. Типы данных**

На основе созданного представления о структуре учебного материала, должны быть созданы базы данных, разделяющие материал по следующим признакам:

### **2.2. Теоретический материал**

Для его хранения в базе данных используются поля: названия предмета, раздела и темы, к которым относятся данная статья, название статьи, номер в базе данных, текст, и ссылки на другие статьи, связанные по смыслу с данной статьёй.

#### **2.2.1. Задачи**

Для их хранения в базе данных используются поля: названия предмета, раздела и темы, к которым относятся данная задача, номер в базе данных, её

название (если название не указано пользователем, то используется её номер в базе данных), условие задачи, названия изображений, используемых в условии, ответ, решение задачи, названия изображений, используемых в решении, ссылки на статьи, связанные с задачей. Решение, ссылки на статьи, изображения, связанные с условием и решением не обязательны для заполнения пользователем

### **2.2.2. Тесты**

Для хранения тестов в базе данных, тест разбивается на отдельные вопросы, для хранения которых используются поля: номер в базе данных, названия предмета, раздела и темы, к которым относятся данный вопрос, номер варианта, номер вопроса, тип вопроса (вопрос типа «А» (с выбором из нескольких вариантов ответа) или типа «В» (с кратким ответом)), 4 варианта ответа (для вопроса типа «В» используется только одно из 4 полей), изображения, используемые в вопросе. Для хранения правильного варианта ответа в вопросах типа «А» используется первое поле.

## **2.3. Типы графических интерфейсов**

Для представления материала требуется использовать язык программирования, имеющий возможность создания графического интерфейса, поддерживающего изображения.

Для навигации по материалу в графическом интерфейсе должны быть разработаны следующие разделы:

### **2.3.1. Главное меню**

В данном окне пользователь может выбрать в какое из окон (теория, задачи, тесты) ему нужно перейти. Также возможен переход в режим редактирования.

### **2.3.2. Выбор темы**

Пользователь выбирает интересующую его тему, используя список предметов, разделов и тем, материалов, занесённых в БД. Если пользователь

выбрал режим редактирования, то он может и добавить новый предмет раздел или тему, если материал по ним отсутствует

### **2.3.3. Теоретический материал**

Пользователь просматривает статьи по выбранной теме. Существует возможность перемещения между статьями данной темы и на статьи, указанные в ссылках. Материал предоставляется последовательностью текста и изображений. В режиме редактирования пользователь может изменять название статьи, её текст, привязанные изображения, ссылки на другие статьи. Также возможно удаление статьи.

### **2.3.4. Задачи**

Пользователь решает задачи по выбранной теме, вводя ответ в отведённое поле. В случае, если для задачи написано решение, пользователь может просмотреть его. Существует возможность перехода на теоретические статьи, на которые были указаны ссылки. В режиме редактирования пользователь может изменять условие задачи, ответ, решение, привязанные изображения и ссылки на теоретические статьи. Также возможно удаление статьи.

### **2.3.5. Тесты**

Пользователь проходит один из тестов по выбранной теме. Для вопросов типа «А» требуется выбрать один из предложенных вариантов ответа, для вопросов типа «В» требуется записать ответ в отведённое поле. После завершения теста пользователю выводится таблица с результатами теста. В таблице указываются баллы за тест, правильный ответ и ответ пользователя. В случае, если вариантов по данной несколько, существует возможность пройти другой вариант.

### 3. Технологическая часть

#### 3.1. Используемое программное обеспечение

Для написания программы был выбран язык «Python 3» по причине наличия встроенных библиотек, нужных для реализации поставленных задач.

Для хранения материала используются базы данных «MySQL». Взаимодействие с базами данных осуществляется за счёт встроенной в «Python» библиотекой «sqlite3».

Для создания графического интерфейса была использована встроенная графическая библиотека «tkinter».

#### 3.2. Описание функциональных окон

В программе были созданы 10 классов подпрограмм, отвечающих за создание окон, используемых для представления материала. Данные классы при помощи виджетов «tkinter» создают окна:

##### 3.2.1. Меню

Класс «Mainpage» служит для создания главного меню, содержащее кнопки при нажатии на которые пользователь переходит в класс «ChooseTheme» (для просмотра материала) или «AddThem» (для редактирования).



Рисунок 1. Меню

### 3.2.2. Выбор темы

Класс «ChooseTheme» используется для выбора просматриваемой темы, интересующей пользователя. Программа составляет список всех предметов, разделов, тем и отображает их при помощи меню. После выбора темы пользователь переходит в класс «TheoryWindow», «TaskWindow» или «TestWindow».

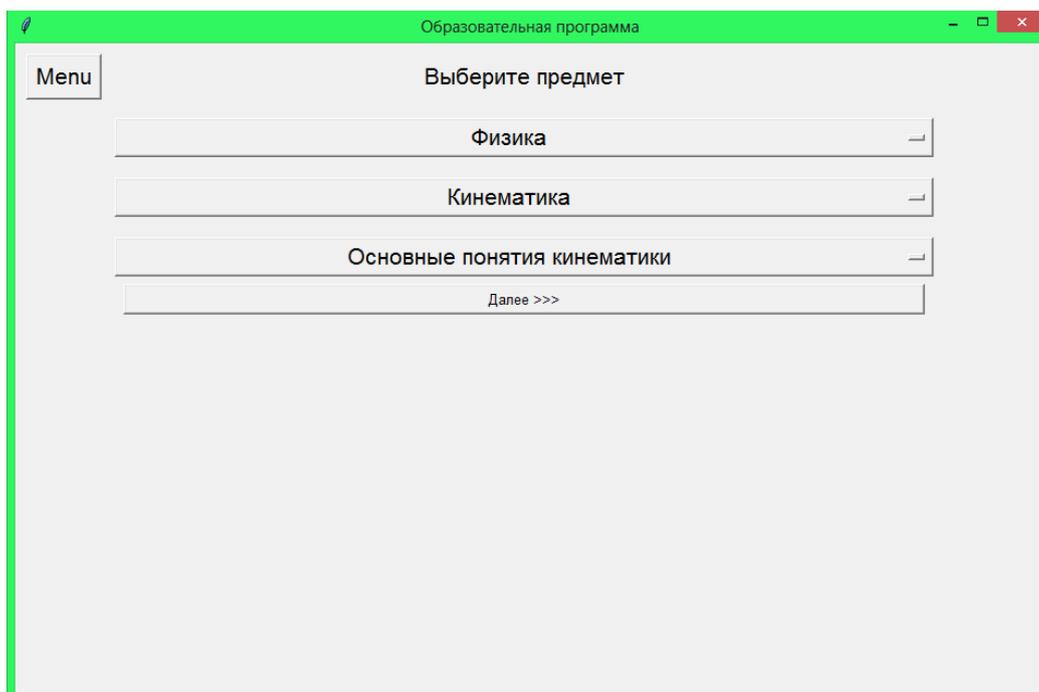


Рисунок 2. Выбор темы

### 3.2.3. Выбор темы (режим редактирования)

Класс «AddThem» используется для выбора редактируемой темы, интересующей пользователя. Программа составляет список всех предметов, разделов и тем и отображает их при помощи меню. В меню также присутствует пункт «Добавить». С помощью данной функции пользователь может добавлять новый предмет, раздел или тему, если материала по ним в базе данных ещё нет. После выбора или добавления новой темы пользователь переходит в класс «AddTheory», «AddTask» или «AddTask».

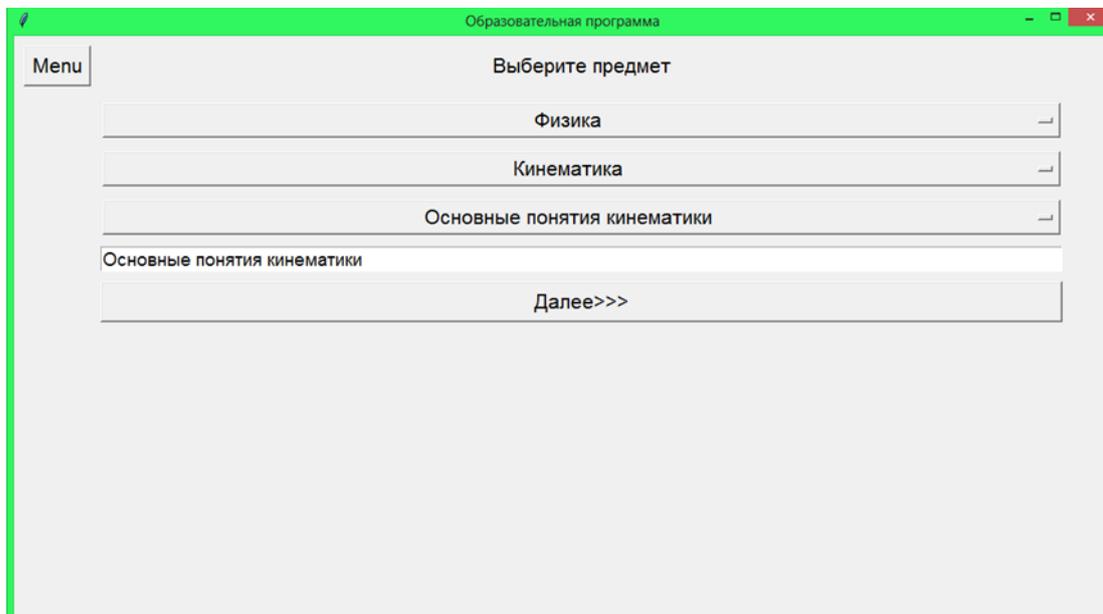


Рисунок 3. Выбор темы (режим редактирования)

### 3.2.4. Редактирование теоретического материала

Для добавления и редактирования теоретического материала требуется создание редактора текста. Для реализации редактора теоретического материала используется класс «AddTheory». Для удобства редактирования и последующего отображения существует возможность деления темы на отдельные статьи. В данном классе для редактирования текста статей используется виджет «Text» (поле ввода), в который из базы данных загружаются текстовые данные. Так как требуется использование изображений, привязанных к определённым отрывкам текста, простым в реализации и несложным в освоении является указание имён изображений внутри текста. Для добавления изображения в статью пользователь должен указать его имя в фигурных скобках (пример: {picture.gif}) в нужном ему месте в тексте.

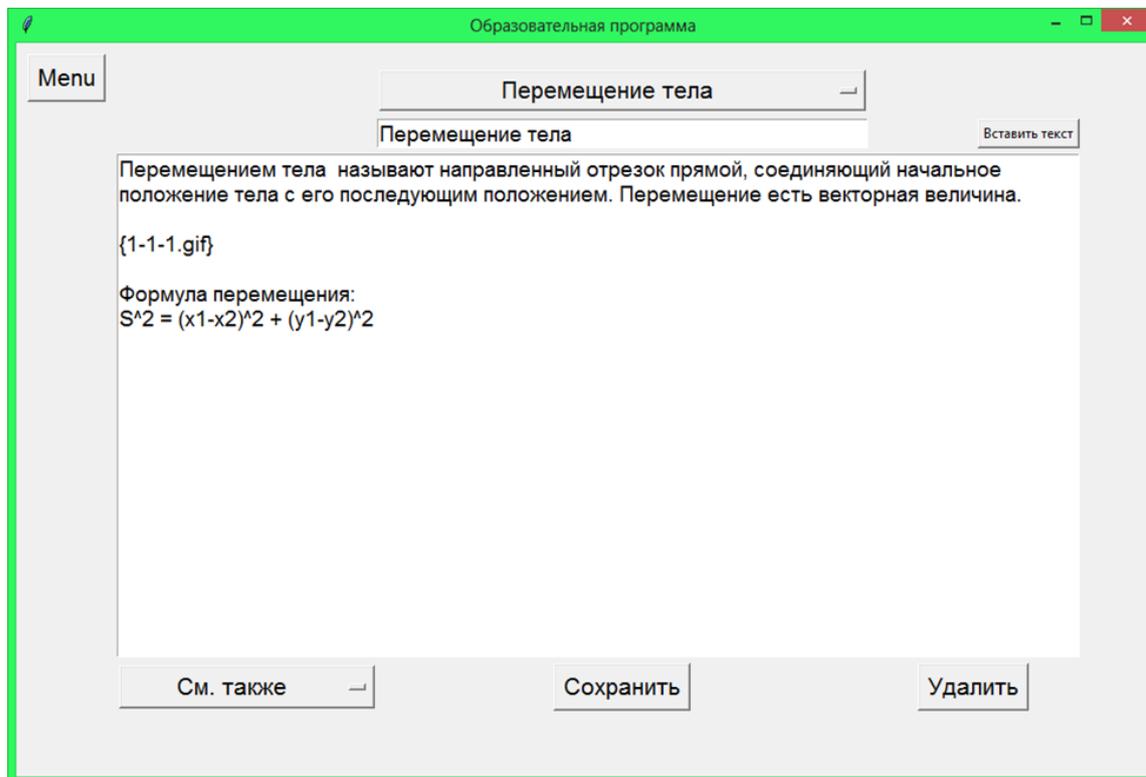


Рисунок 4. Редактирование теоретического материала

### 3.2.5. Редактирование задач

Для реализации редактора задач используется класс «AddTask». В данном классе для редактирования условия и решения задачи используются поля ввода. Так как изображения в условии и решении не требуют привязки к тексту их можно указать отдельно от условия и решения. Для добавления изображений, связанных с условием и решением задачи используются поля ввода («Entry»). Для добавления изображений в данных полях нужно указать имена изображений, разделяя их пробелами (пример: picture1.gif picture2.gif). В случае если изображение требуется вставить в определённое место текста, можно использовать метод, используемый для добавления изображений в теоретический материал.

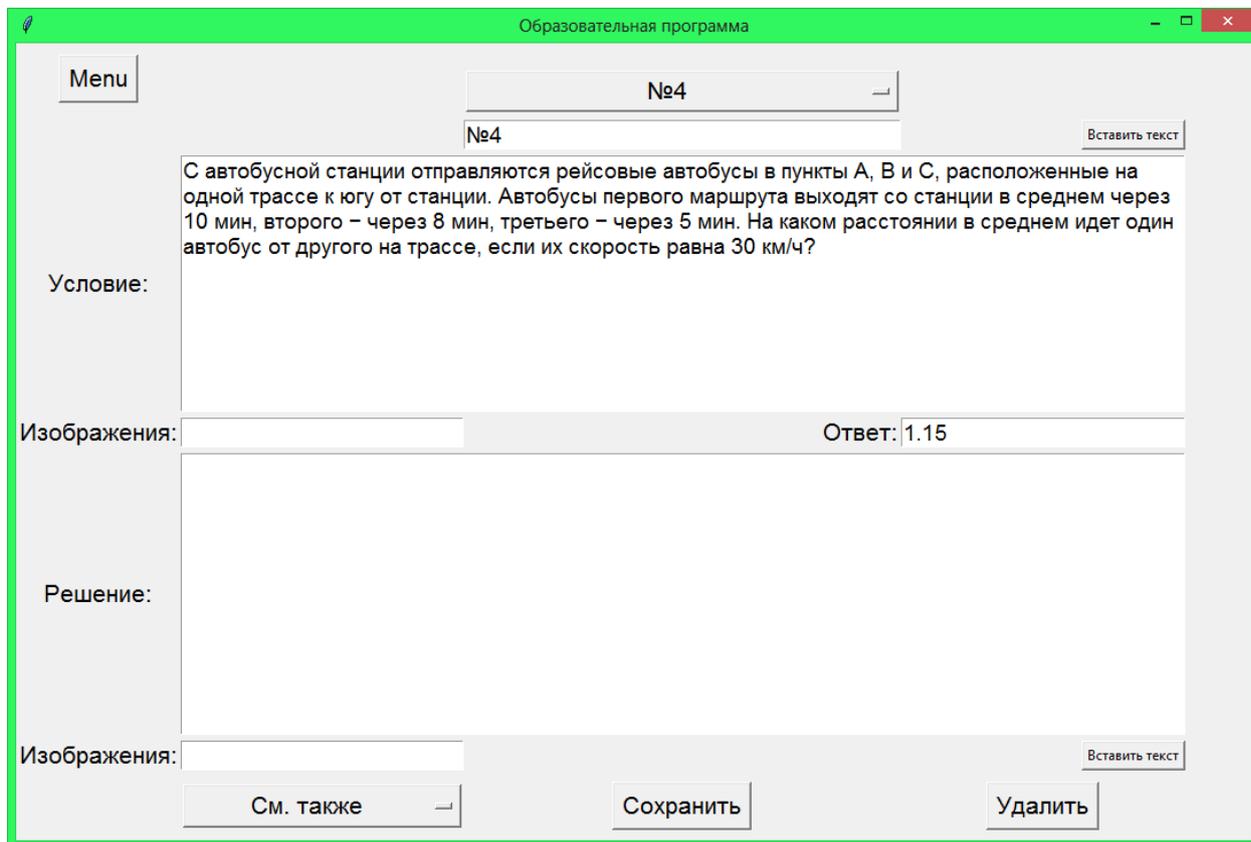


Рисунок 5. Редактирование задач

### 3.2.6. Редактирование тестов

Для реализации редактора тестов используется класс «AddTest». В данном классе для редактирования вопросов и добавления изображений используется поле ввода. Аналогично изображениям в задачах, пользователь может добавлять изображения, привязанные, и не привязанные к определённому месту в тексте. В редакторе пользователь может выбрать тип теста. В случае выбора вопроса типа А пользователь может ввести от 2-х до 4-х вариантов ответа на вопрос. Ответом может являться и изображение (в таком случае должно быть указано его имя, например, {picture.gif}). В случае выбора вопроса типа В пользователь может указать ответ на вопрос.

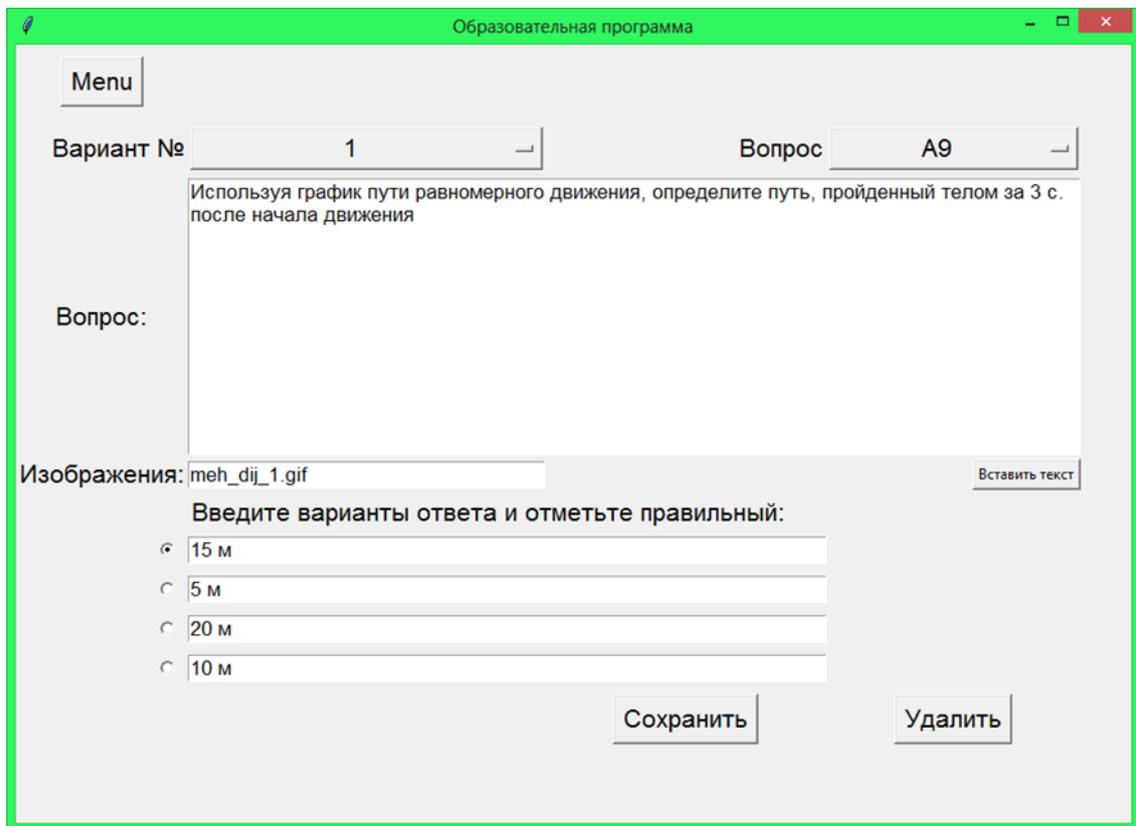


Рисунок 6. Редактирование тестов

### 3.2.7. Просмотр теории

Для отображения теоретического материала используется класс «TheoryWindow». В данном классе пользователю предоставляется материал по теме, выбранной пользователем. При переходе в класс создаётся список всех статей данной темы, отображаемый при помощи меню, и, в случае перехода в этот класс из «ChooseThem», отображает первую статью из списка, или отображает определённую статью, в случае если переход был осуществлён по ссылке или из этого же класса. Текст статьи разбивается программой на обычный текст и изображения (названия изображений указываются в фигурных скобках). После текст и изображения последовательно отображаются при помощи виджетов «Message» (для отображения текста), и «PhotoImage», «Label» (для отображения изображений). В случае если формат изображения не поддерживается или отсутствует в папке «pic», выводится надпись «Некорректное изображение!». После статьи следует меню, содержащее список

всех ссылок на другие статьи.

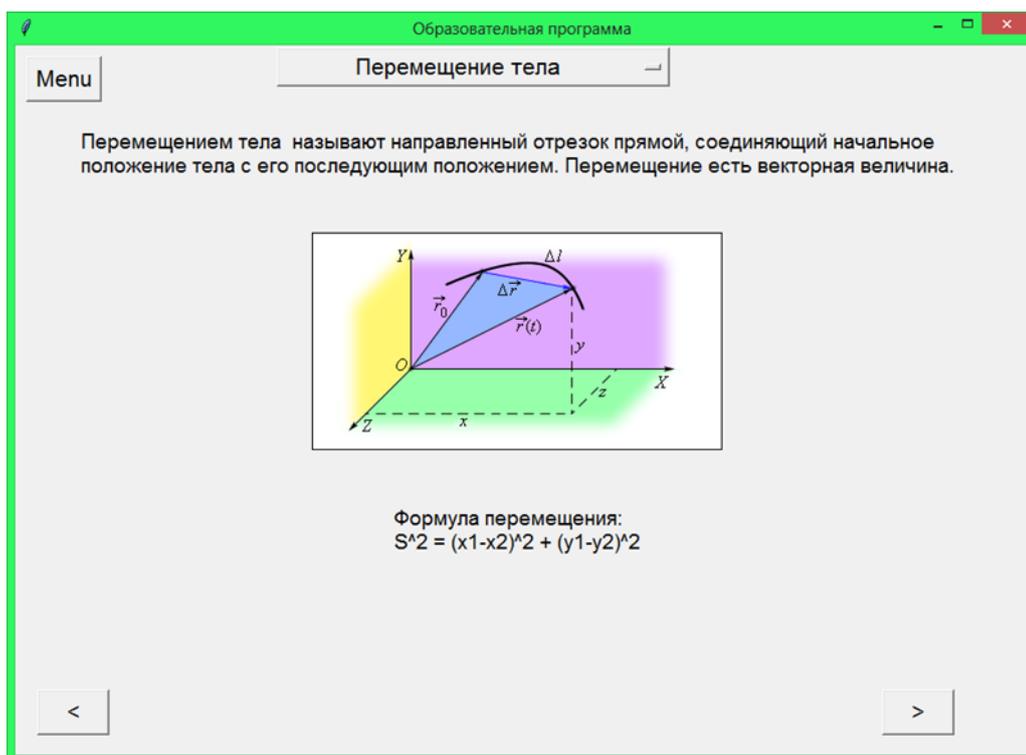


Рисунок 7. Просмотр теории

### 3.2.8. Просмотр задач

Для отображения задач используется класс «TaskWindow». В данном классе пользователю предоставляется задача по выбранной теме. При переходе в данный класс создаётся список всех задач по данной теме, отображаемый при помощи меню. В случае перехода в класс из класса из «ChooseThem» отображается первая задача из этого списка. Программа отображает изображения и условие задачи. Условие задачи отображается аналогично тексту в классе «TheoryWindow». Рядом с условием отображаются изображения, введённые пользователем в отдельной строке. Под условием размещено поле ввода для ответа и кнопка «Ответить». После введения ответа он сравнивается с правильным ответом и, если они совпадают, то выводится надпись «Правильно», если относительная погрешность введённого ответа менее 5%, то ответ считается правильным и выводится надпись «Правильно» и указание относительной погрешности. Если погрешность ответа более 5%, то ответ считается неправильным и надпись «Неправильно» и кнопка «Решение», при

нажатию которой под условием отображается решение задачи и изображения, связанные с решением.

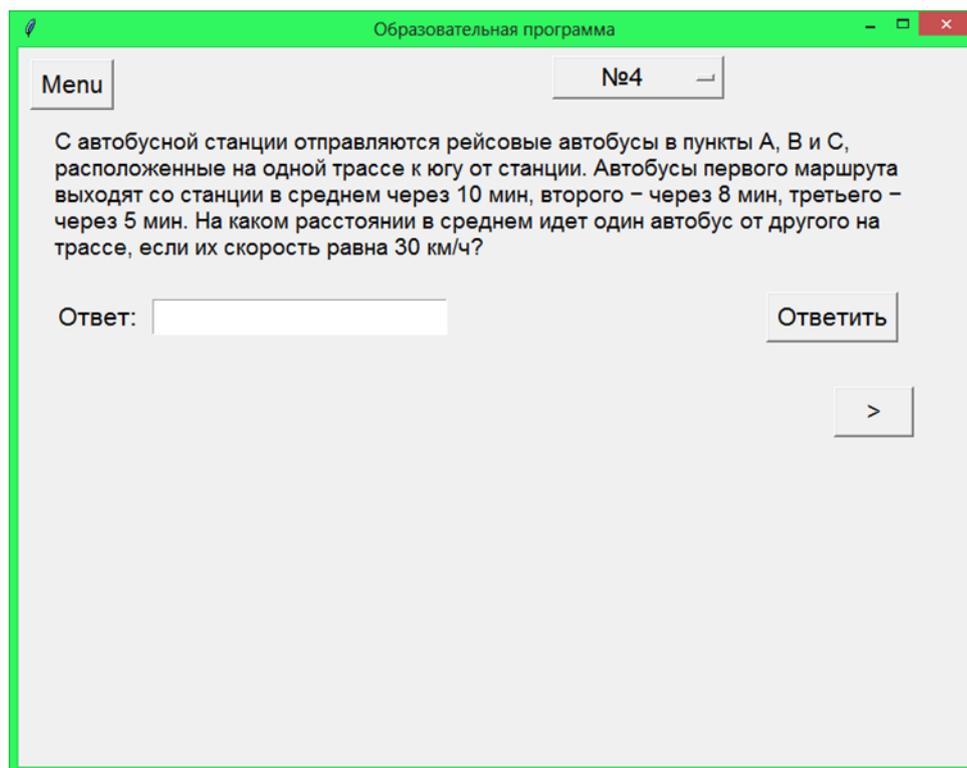


Рисунок 8. Просмотр задач

### 3.2.9. Решение тестов

В классе «TestWindow» пользователю предоставляется набор вопросов одного из тестов по выбранной теме. При переходе в данный класс программой случайным образом выбирается один из тестов по заданной теме. Из вопросов, включённых в выбранный тест, составляется список, отображаемый при помощи меню. Случайным образом определяется и запоминаются программой расположения вариантов ответов в вопросах типа «А».

Отображается первый вопрос из списка. Для отображения вопроса используется текст вопроса и изображения, связанные с ним. Пользователю представляются вопросы для которых он должен выбрать один из предложенных ответов (вопрос типа «А») или написать краткий ответ (вопрос типа «В»). В последнем вопросе выбранного теста представляется кнопка «Завершить тест». При нажатии данной кнопки пользователь переходит в класс

«TestTotal».

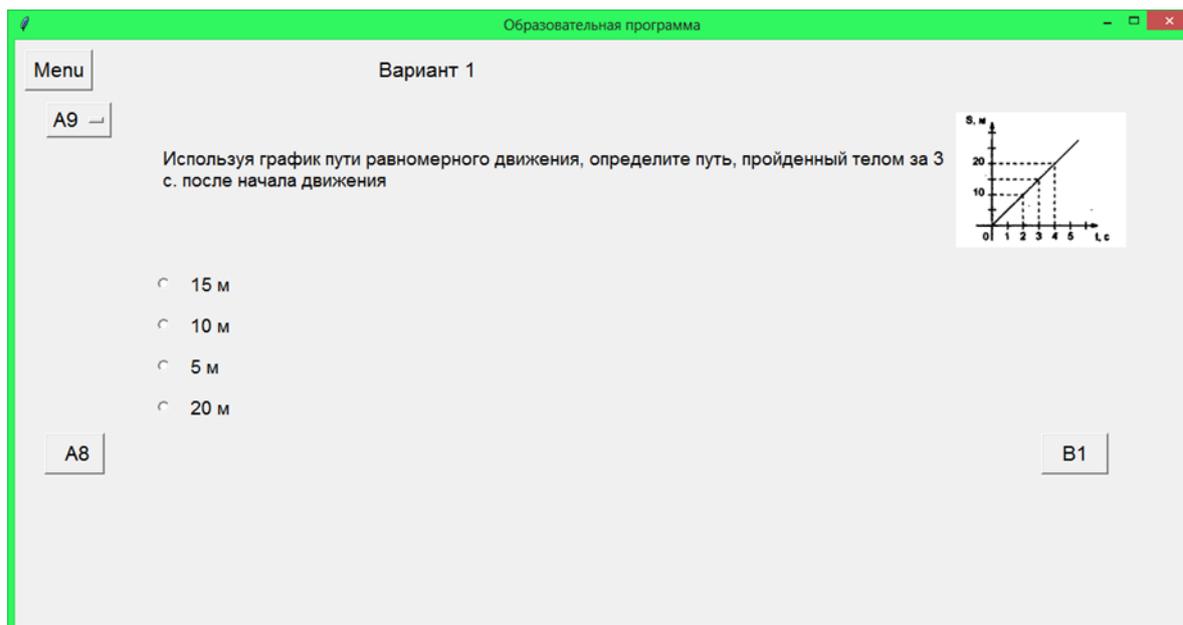


Рисунок 9. Решение тестов

### 3.2.10. Просмотр результатов теста

В классе «TestTotal» пользователю предоставляются результаты теста, в виде таблицы. Таблица содержит названия вопросов, ответы, данные пользователем, правильные ответы и количество баллов, полученное за данный вопрос. За правильный ответ на вопрос типа А присваивается 1 балл, 2 балла за вопрос типа 2. После таблицы подсчитываются все баллы, полученные пользователем. При помощи отображаемых кнопок с надписями «Другой тест» и «Теория» пользователь может пройти другой тест по данной теме и изучить теоретический материал по данной теме (если такой существует).

Образовательная программа

Menu

Результаты теста:

Вопрос	Ваш ответ	Правильный ответ	Баллы
A1	Механическим движением	Механическим движением	1
A2	Движение, при котором тело в любые равные промежутки времени проходит равные пути	Движение, при котором тело в равные промежутки времени проходит равные пути	0
A3	Земли	Вагона	0
A4	Времени прохождения телом единицы пути	Пути, пройденному телом за единицу времени	0
A5	30 000	30 000	1
A6	20 км/ч	5 м/с	0
A7	9 м/с	1,5 м/с	0
A8	1250 с.	50 с.	0
A9	15 м	15 м	1
B1	111111	3600	0
Всего:			3 / 11

Рисунок 10. Просмотр результатов теста

### 3.3. Особенности реализации

Для сохранения введённых данных используется кнопка «Сохранить». При переходе между статьями, задачами и тестами изменения не сохраняются. Для удаления используется кнопка «Удалить».

Изображения, используемые в материале, должны находиться в папке «pic» в одной директорией с программой.

Для удобства навигации и понимания материала существует возможность добавления ссылок на теорию, связанную с редактируемым материалом.

Перед сохранением материала программа проверяет корректность введённых данных и выдаёт сообщение, в случае введения некорректных данных.

Для использования программы на компьютерах без установленного «Python 3» программу требовалось скомпилировать в исполняемый файл. Для данной цели была использована программа «cx\_Freeze».

## Заключение

Данная программа может быть использована школьниками и студентами для изучения учебного материала, проверки знаний путём решения задач и тестов. За счёт функции добавления образовательного материала данная программа может быть использована педагогами для создания пособий, сборников тестов и задач.

За счёт использования обобщённой схемы представления теоретического материала, задач и тестов и за счёт использования изображений на основе программы возможно реализовать большинство школьных предметов в рамках данной программы.

Результатом работы над данным проектом стала образовательная программа. Полученный продукт обладает преимуществом перед аналогами за счёт возможности привлечь к себе внимание преподавателей, заинтересованных в простом создании собственных пособий. Также программа создаёт возможность школьникам и студентам изучать предмет по самостоятельно собранному материалу.

## Ссылки

При изучении языка «Python 3.4.3» и библиотек «tkinter», «sqlite3» были использованы сайты:

<http://younglinux.info/tkinter>

<http://john16blog.blogspot.ru/2011/03/python-sqlite3.html>

<http://ru.stackoverflow.com/questions/231251/Изображения-в-python>

<https://pythonworld.ru/osnovy/program-compilation-with-cx-freeze.html>

[http://www.lfd.uci.edu/~gohlke/pythonlibs/#cx\\_freeze](http://www.lfd.uci.edu/~gohlke/pythonlibs/#cx_freeze)

Для добавления учебного материала были использованы сайты:

<http://www.physics.ru>

<http://fizmatsite.narod.ru>

<http://www.afportal.ru/physics/test/easy>