

# Отчет по проекту для конференции «Инженеры будущего» Направление ИТ – программирование

Работу выполнил:

Ученик 11 класса ГБОУ школы 1502 «Энергия»

Жуков Никита Сергеевич

Научный руководитель:

Учитель информатики

Буркатовская Ольга Сергеевна

## Аннотация

- Целью проекта является создание приложения для демонстрации и визуализации законов кинематики и баллистики, а также предоставить пользователю интерфейс для решения практических задач на эти темы и ознакомление с теорией.
- Созданное приложение позволяет строить траекторию движения брошенного тела, в зависимости от указанных параметров. Также предоставляется возможность рассчитывать баллистические величины по заданным параметрам.

## Актуальность

- Данный проект можно использовать ученикам средних классов для изучения основ кинематики и баллистики, а также для практики в решении задач на эти темы.
- Помимо учеников, им могут пользоваться педагоги для демонстрации изменения движения брошенного тела в зависимости от изменяемых

величин. Например, показывать, как меняется дальность полета при изменении угла броска или как бросать тела с разных высот, чтобы они падали в одной точке.

- Также приложение будет полезно для методистов, составляющих сборники задач и при составлении самостоятельных и контрольных работ. Пользуясь технологией поиска неизвестной величины, можно выбирать какие значения давать в условии, чтобы ученик мог найти какую-то конкретную величину. Также перебирая различные значения для величин, можно подбирать так, чтобы ответ получался «красивым».

## Цель и задачи

1. Предоставить возможность демонстрировать полет брошенного тела при различных значениях таких параметров, как начальная скорость полета, угол броска и начальная высота, с которой бросают тело.
2. Реализовать сохранение результатов броска в базе данных и возможность просматривать, загружать, и удалять данные об экспериментах. Добавить выбор того, как сохранять график: картинкой или анимацией (.gif).
3. Сделать функционал для поиска всех баллистических величин: начальная скорость, угол броска, максимальная высота полета, дальность полета, время подъема, время спуска, полное время полета, максимальная скорость во время полета, минимальная скорость во время полета. Реализовать возможность ввода всех известных величин и поиск по ним остальных. А также интерфейс, который будет показывать варианты переменных, по которым можно рассчитывать какую-то целевую величину.
4. Реализовать возможность просматривать блок теории по кинематике и баллистике. А именно: просматривать уроки в браузере или локально просматривать параграфы из учебника.

## Порядок выполнения задач

1. Используя библиотеку `matplotlib` и `numpy`, осуществить построение графика траектории движение по указанным параметрам. Придумать удобный интерфейс для работы с графиком, а также реализовать анимация «движения» материальной точки при броске.

2. Далее придумать способ хранения величин и формул расчета в программном коде. После чего, используя библиотеку для символьных вычислений SymPy, реализовать функции и методы, рассчитывающие значения и формулы.
3. Пользуясь функциями для поиска всех баллистических величин, находить всю информацию о броске по трем заданным параметрам и демонстрировать их в отдельном окне. Добавить интерфейс для сохранения экспериментов в базу данных. После чего создать отдельный виджет для просмотров всех сохраненных экспериментов, с возможностью загрузки и удаления данных об отдельных из них.
4. Далее добавить окно настроек с указанием таких параметров как: значение константы ускорения свободного падения ( $g$ ), скорость отрисовки графика и необходимость делать анимацию графика.
5. Последним пунктом добавить возможность просматривать теорию в браузере или локально в учебнике.

## Анализ существующих решений

Просматривая в интернете различные сервисы, которые решают похожие задачи, можно наткнуться либо на сайты, которые строят графики по заданным функциям, либо различные сервисы с различными калькуляторами. Среди этих калькуляторов можно найти и те, которые рассчитывают баллистические величины, однако ими достаточно неудобно пользоваться, и из-за большого объема возможностей сервис становится очень объемный и тяжелый в использовании.

Полноценного сервиса, который реализует все поставленные задачи, найти так и не получилось.

## Новизна решения

Разработанное приложение получилось более тематическим и подходящим для решения всех задач по данным темам. Продукт объединяет в себе различные сервисы с удобным для пользователя интерфейсом. Большим плюсом является то, что приложение предоставляет огромной функционал (в том числе в области поиска различных величин), с возможностью настраивания различных параметров. Однако основную работу для рядового пользователя можно делать очень быстро и просто с удобным интерфейсом.

# Описание реализации

## Структура проекта

1. Структура проекта содержит в себе папку `release`, которая содержит в себе файлы и независимую сборку приложения (exe-файл с библиотеками, необходимыми для его работы).
2. Также в проекте содержится папка `resources`, содержащая в себе вложенные каталоги со всеми ресурсами для проекта.
  - a. Каталог `database`, содержащий в себе файл с базой данных, pdf файл с учебником (для просмотра теории) и двух текстовых файлов, демонстрирующих правильный и неправильный формат файла для загрузки информации с него.
  - b. Каталог `images`, содержащий в себе изображения, используемые в проекте
  - c. Каталог `results`, в котором находятся 2 папки `graph_files` и `info_files`. В этих папках хранятся текстовые файлы с информацией о сохраненных бросках и файлы с графиками (.jpg или .gif в зависимости от выбранного способа сохранения)
  - d. Каталог `ui_files`, в котором находятся файлы .py с разметкой всех окон и папка, содержащая исходные файлы разметок окон (.ui)
3. Проект содержит в себе основной модуль `main.py`, в котором прописана основная логика взаимодействия с интерфейсом, а также 3 вспомогательных модуля:
  - a. `database_requests.py`, в котором находятся функции для выполнения запросов в базу данных.
  - b. `formulas.py`, содержащий в себе описание всех баллистических величин, формул расчета и функций для вычисления всех параметров.
  - c. `constants.py`, в котором прописаны глобальные переменные, используемые в приложении
4. Модуль `get_variables_collections.py`, который выполняет функцию поиска всевозможных комбинаций переменных для всех баллистических величин и сохраняет результаты в специальную таблицу в базе данных для дальнейшего использования в программе при поиске конкретной величины.

## Отрисовка графика

Для отрисовки графиков используется библиотека `matplotlib`. Эта библиотека очень популярна и с ее помощью можно визуализировать почти любые данные. `matplotlib` предоставляет удобный функционал по отрисовке графиков, их кастомизации и созданию анимации, именно поэтому я выбрал именно эту библиотеку для реализации своего приложения.

В файле `main.py` содержится описание класса `TrajectoryGraph`, который наследуется от класса `FigureCanvasQTAgg` из `matplotlib`. В основном методе `plot()` выполняется отрисовка графика. Происходит считывание значений из виджетов, расчет основных параметров бросков и вычисление точек для графика (с помощью библиотеки `numpy`). После чего рисуется график, отображаются значения максимальной высоты подъема, дальности полета и времени полета, после чего происходит запуск анимации с указанной скоростью. Метод `scatter()` рассчитывает и рисует положение тела (материальную точку) в определенный момент времени.

## Вычисление баллистических величин

Для описания всех величин был написан класс `Symbol`, наследуемый от `sympy.Symbol`. Класс наследник нужен был корректировки работы с угловыми величинами. Так как для пользователя угол задается в графиках, а в математические функции принимают радианы.

Также модуль содержит в себе класс `Formula`, в котором описывается сама формула, величины, которые есть в формуле, а также метод, проверяющий возможность рассчитать какую-то величину по данной формуле и метод, выполняющий эти вычисления. Для вычисления значения, в формулу средствами библиотеки `SymPy` подставляются известные значения, после чего выполняется расчет относительно целевой переменной. Также отдельно рассчитывается формула для целевой переменной, чтобы потом вывести ее для пользователя.

Основная функция модуля `find()` реализует поиск всевозможных переменных, по тем, которые были переданы. Для этого функция проходит по всем известным формулам и проверяет по каким из них можно что-то рассчитать, и выполняет эти вычисления. После чего функция повторяет свою работу до тех пор, пока удастся найти новые переменные.

Также модуль меняет некоторые формулы расчета, в зависимости от известности начальной высоты  $y_0$ , так как некоторые из них являются частным случаем, когда  $y_0=0$ . Так как в задачах чаще всего это известно, а частные формулы более просты, то было принято решение пользоваться ими по возможности.

## Модуль для выполнения запросов в базу данных

Этот модуль был написан мной несколькими месяцами ранее для другого проекта, однако его использование показало его огромную эффективность, в связи с чем было принято решение использовать проверенные функции для различных запросов. Все функции генерируют строковые запросы по заданным параметрам, после чего выполняется запрос с помощью библиотеки `sqlite3`.

## Модуль `get_variables_collections`

Данный файл содержит в себе функции для определения всех комбинаций переменных для всех величин. Так как эти комбинации всегда одинаковы, то нецелесообразно каждый раз считать их заново, поэтому найденные комбинации сохраняются в базу данных, после чего в основной программе при необходимости просто считываются из определенной таблицы.

Для определения этих комбинаций используется ранее описанная функция `find()` и одна из функций библиотеки для работы с итераторами `itertools`. С помощью `itertools.combinations()` были получены всевозможные комбинации всех переменных всех длин, после чего эти итераторы были объединены с помощью `itertools.chain()`, а потом все эти комбинации фильтровались по двум критериям: можно ли найти целевую переменную зная эти величины (с помощью вызова `find()` от этой комбинации и проверки находится ли в словаре целевая переменная), уникальность этой комбинации. Например, если переменную `a` можно найти, зная `b` и `c`, то очевидно, что ее можно найти, зная `b`, `c` и `d`. В этом случае 2 комбинация не представляет интерес, так как требует дополнительную величину, поэтому все такие комбинации отбрасываются (посредством проверки не является ли хоть одна найденная комбинация подмножеством проверяемого множества в функции `is_uniqe()`).

## Перспективы проекта

У проекта много различных улучшений. Можно развивать работу с графиком, добавляя перемещение по нему и масштабирование. Дополнительная кастомизация графика и возможность настраивать его внешний вид и работу.

Развивать алгоритм поиска и вычисления величин, добавив вывод пошагового вычисления величины из формулы. Улучшить разбор программой вводимых значений и дополнительные проверки на корректность результата.

Также такая реализация калькулятора величин позволяет достаточно просто интегрировать проект под другие темы физики, благодаря чему можно двигать проект в сторону расширения тем, по которым можно обучаться и тренироваться.

## Литература

1. РЭШ (теория и формулы для кинематики и баллистики) <https://resh.edu.ru>
2. Официальная документация по библиотеке QT <https://doc.qt.io>
3. Официальная документация по библиотеке matplotlib <https://matplotlib.org/stable/index.html>
4. Официальная документация по библиотеке SymPy <https://docs.sympy.org/latest/index.html>