Департамент образования и науки города Москвы

Юго-восточный административный округ

Государственное бюджетное общеобразовательное учреждение

города Москвы «Школа № 1547»

**«ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ИЗУЧЕНИЯ РАБОТОСПОСОБНОСТИ МОЗГА»**

**Автор:** Соловьев Николай Константинович,  
ученик 11-И класса

ГБОУ «Школа №1547»

**Руководитель:** Кекконен Э. А.,  
учитель ГБОУ «Школа №1547»

**Москва, 2022г.**

**Содержание**

[Аннотация 3](#_Toc95854558)

[Введение 4](#_Toc95854559)

[Описание проделанной работы 5](#_Toc95854560)

[Заключение 26](#_Toc95854573)

[Приложения 27](#_Toc95854574)

[Библиографический список 32](#_Toc95854575)

# Аннотация

Данный проект является частью исследовательской работы, проводимой в институте ВНД (Высшей Нервной Деятельности) магистрами Биологического Факультета МГУ. Исследование состоит в изучении работы человеческого мозга: доказать (или опровергнуть), что моторные зоны коры больших полушарий головного мозга (мелкая моторика) просыпаются из глубокого сна быстрее, чем когнитивные зоны (способность решать аналитические задачи).

В проекте представлено программное обеспечение для проведения нейрофизиологических экспериментов, разработанное по требованиям, представленным биологами. В качестве языка программирования для реализации поставленных задач, был выбран Python3[1]. В том числе были изучены и использованы библиотеки: Tkinter[2], Pygame[3], OS[4], Time[5], Serial[6]. В рамках проекта были разработаны:

* Графические интерактивные программы:
* для оценки качества решения простейших арифметических задач
* для определения точности моторных функций (движение руки)
* Графический интерфейс для настройки конфигурации приложения и алгоритм сохранения и восстановления этой конфигурации
* Алгоритмы обработки и сохранения результатов экспериментов
* Интеграция с программно-аппаратным комплексом “Энцефалан”[7] для создания временных меток на энцефалограммах.

# Введение

**Обоснование актуальности:** Актуальность данного решения обусловлена отсутствием общеизвестных приложений, решающих описанные задачи. Так как основную часть проекта составляет создание и разработка программного приложения, он относится к направлению «Программирование».

**Цель и задачи**

**Основной целью моего проекта** является создание графического приложения, предназначенное для выполнения моторно-зрительных и когнитивных задач испытуемым после пробуждения из глубокой фазы сна.

**Задачи**, с которыми мне предстояло столкнуться:

1. Выбор языка программирования и библиотек, которые понадобятся для реализации цели.
2. Реализация программы “Мыши” для проверки моторной функции.
3. Реализация программы “Задачки” для проверки когнитивной функции.
4. Реализация интеграции программ с электроэнцефалограф-регистратором.
5. Реализация графического интерфейса для настройки описанных программ и сборка в одно приложение.

# Описание проделанной работы

## Выбор языка программирования и библиотек, которые понадобятся для реализации цели

Первым делом я приступил к выбору языка программирования. Мною был выбран язык Python3, т.к. он очень прост в освоении, для него есть большое количество библиотек и документаций к ним. Также нужно было решить задачи разработки динамической графики, работы с файлами, работы с внешними устройствами. Для этого я использовал библиотеки:

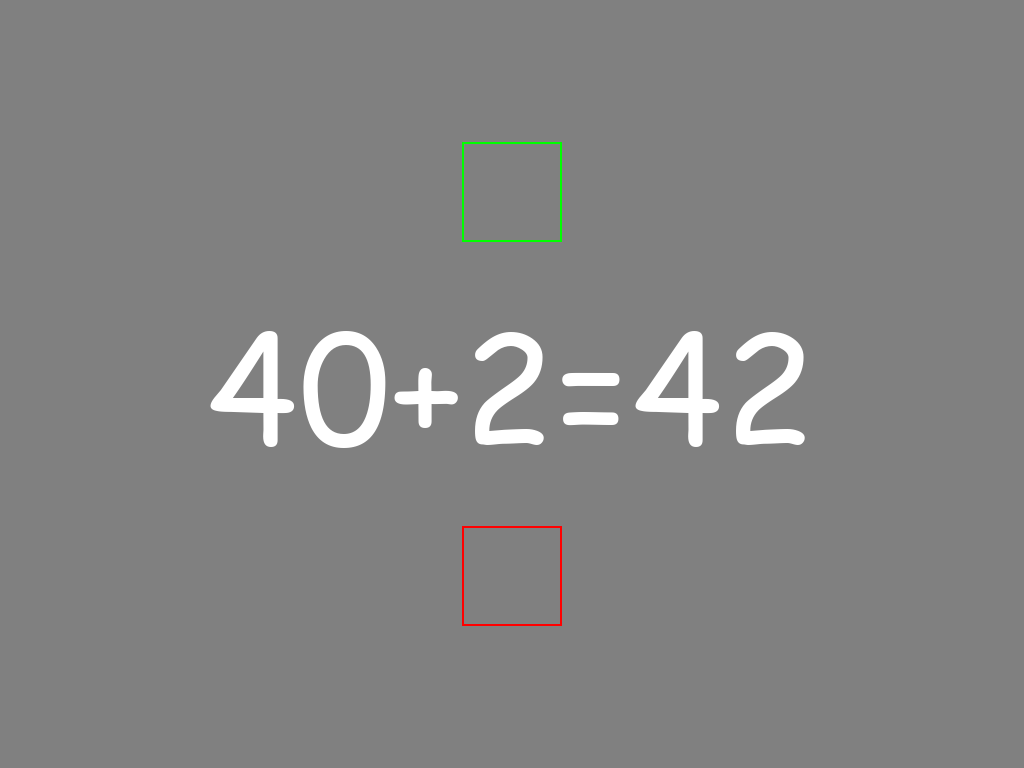
* + - 1. Pygame – это библиотека для разработки динамической графики и обработки изображений.
      2. OS – встроенная библиотека для работы с файлами и директориями.
      3. Serial – библиотека для работы с внешними устройствами за счет коммуникации с помощью USB порта.
      4. Tkinter – библиотека, с помощью которой можно создавать простые графические интерфейсы

## 

## Реализация программы «Задачки»

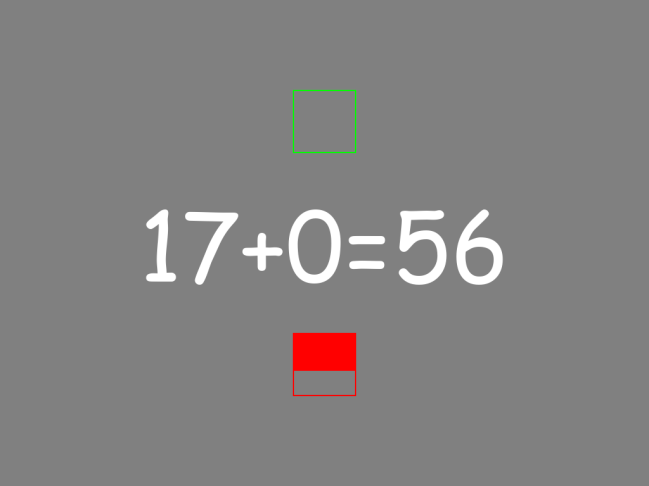
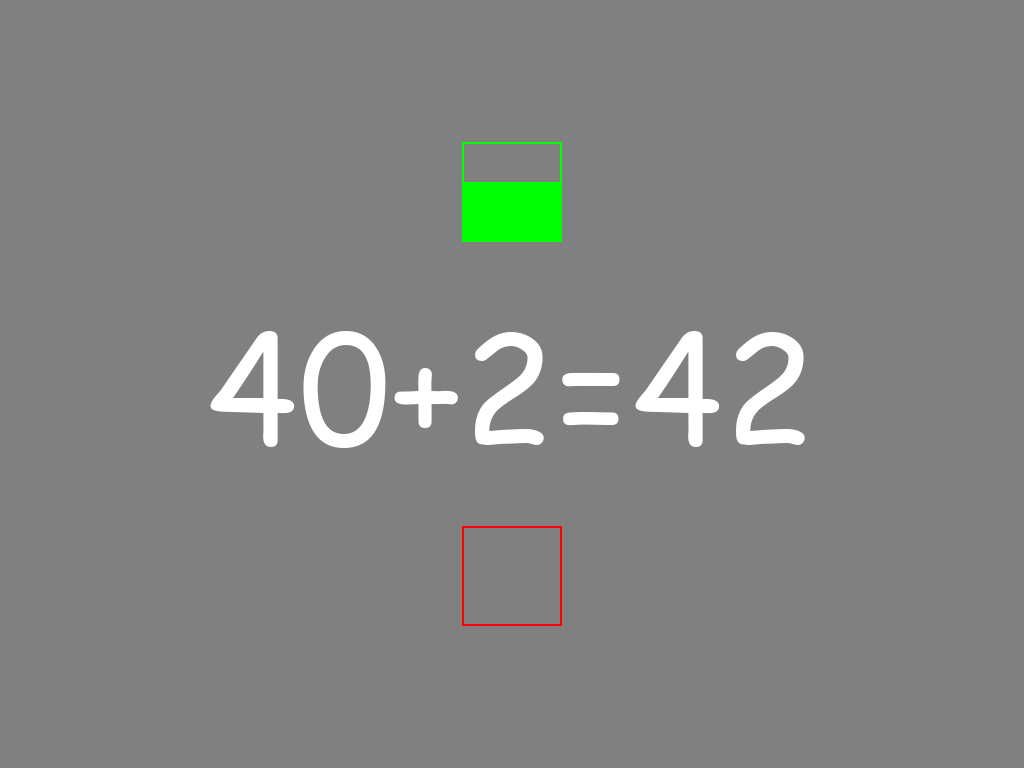
Я начал разработку приложения с реализации программы “Задачки”, с помощью которой проводятся опыты по проверке когнитивной функции человека (см. Рисунок 1).

### Основной алгоритм



**Рисунок 1. Пример выполнения программы «Задачки»**

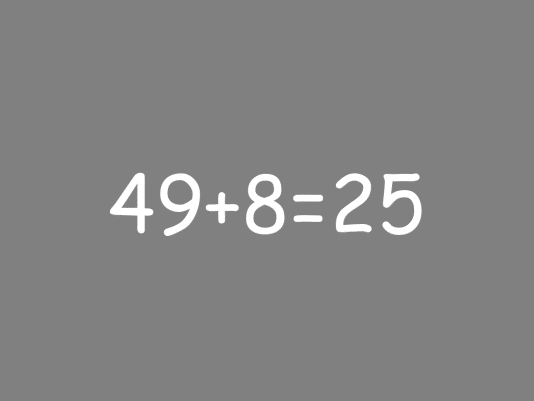
Программа воспроизводит звуковой тон для пробуждения испытуемого. На экране появляется окно с отображением сгенерированного равенства и два квадрата: зеленый и красный, сверху и снизу. Испытуемый должен крутить колесико мыши вверх, чтобы ответить: “Равенство верно” или вниз, чтобы ответить: “Равенство неверно” (см. Рисунок 2). Также на экране отображается заполнение одного из квадратов, показывающее ответ испытуемого. После каждого опыта программа запоминает ответ для каждого равенства.



**Рисунок 2. Примеры ответов: «Равенство верно» (слева), «Равенство неверно» (справа)**

Процесс разработки программы был итерационным: биологи выставляли требования, я делал реализацию, они пробовали проводить эксперименты. В результате этих пробных экспериментов требования корректировались.

Например, в первой версии был предложен немного другой ход работы программы: на экране отображалось только сгенерированное равенство (см. Рисунок 3). Если испытуемый считал, что равенство правильное, он нажимал клавишу мыши, чтобы программа засчитала ответ и показала следующее равенство. При неверном равенстве он просто ожидал переключения на следующее равенство.



**Рисунок 3. Пример выполнения первой версии программы «Задачки»**

Но в ходе экспериментов выяснилось, что с этой версией возникла следующая проблема: после пробуждения испытуемый мог опять заснуть и просто перестать работать с программой, а алгоритм интерпретировал это, как пропуск пользователем неверного равенства. Поэтому было решено ввести требования явных действий (колесико мыши вверх/вниз) и дополнительно учитывать пропущенные равенства (испытуемый заснул).

### Алгоритм генерация равенств

С каждым новым опытом программа генерирует новое равенство:

1. Генерируется случайное значение: «да» или «нет», означающее будет ли верным данное равенство
2. Генерируется первое случайное число от 10 до 50
3. Генерируется второе случайное число от 0 до 9
4. На экран выводятся эти два числа в виде сложения
5. Если равенство должно быть верно, то еще выводится их сумма, иначе выводится еще одно случайное число от 0 до 60

### Интерпретация результатов эксперимента

После проведения определенного количества опытов, программа сохраняет результаты в файл с названием эксперимента, датой и временем. В нем сохраняются данные от каждого опыта: номер опыта, правильность сгенерированного равенства, ответ испытуемого, а также другая статистика: кол-во задач правильных и неправильных, сколько и как ответил испытуемый и сколько пропустил (проспал). Пример файла «Tasks 31.01.22 15.57.01.txt» представлен в Таблице 1.

Таблица 1. Пример файла главного отчета по всему эксперименту «Задачки».

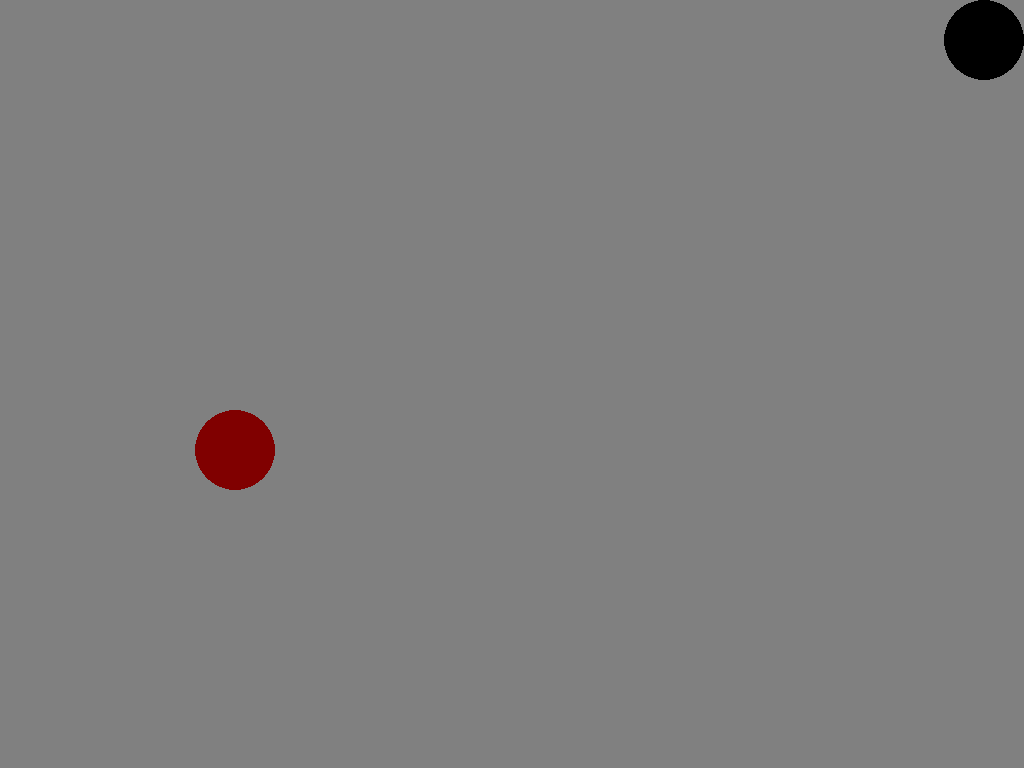
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Опыт | Равенство | | Оценка равенства | | | Ответ | | | |
| 1 | 14+8=22 | | True | | | True | | | |
| 2 | 39+8=47 | | True | | | True | | | |
| 3 | 49+0=57 | | False | | | False | | | |
| 4 | 3+2=27 | | False | | | True | | | |
| 5 | 1+5=6 | | True | | | Missed | | | |
| 6 | 44+7=31 | | False | | | False | | | |
| Задачи | | | | Ответил | | | | | |
| True | | False | | T -> T | F -> F | | T -> F | F -> T | Missed |
| 3 | | 4 | | 2 | 2 | | 0 | 1 | 1 |

Такой формат результатов удобен для последующих обработок в других программах, поддерживающих работу с электронными таблицами, таких как MS Excel. Полный алгоритм работы программы “Задачки” в блок-схемах представлен в Приложении 2 и 3.

## Реализация программы «Мыши»

Далее я приступил к разработке программы «Мыши», с помощью которой проводятся эксперименты по проверке моторной функции испытуемого (см. Рисунок. 4).

### Основной алгоритм



### Рисунок 4. Пример работы программы «Мыши»

На экране отображаются две окружности: красная, подвижная – «Мышь»; черная, неподвижная, расположенная в правом верхнем углу окна – «Нора». Красная окружность из левого нижнего угла движется в сторону «Норы» по заданной программой траектории в виде параболы. Программа просчитывает траекторию так, чтобы «Мышь» самостоятельно не могла дойти до «Норы». Если мышь коснулась одной из сторон окна, то программа считает, что «мышь не дошла до норы», и начинает новый опыт, выставляя красную окружность на изначальную позицию.

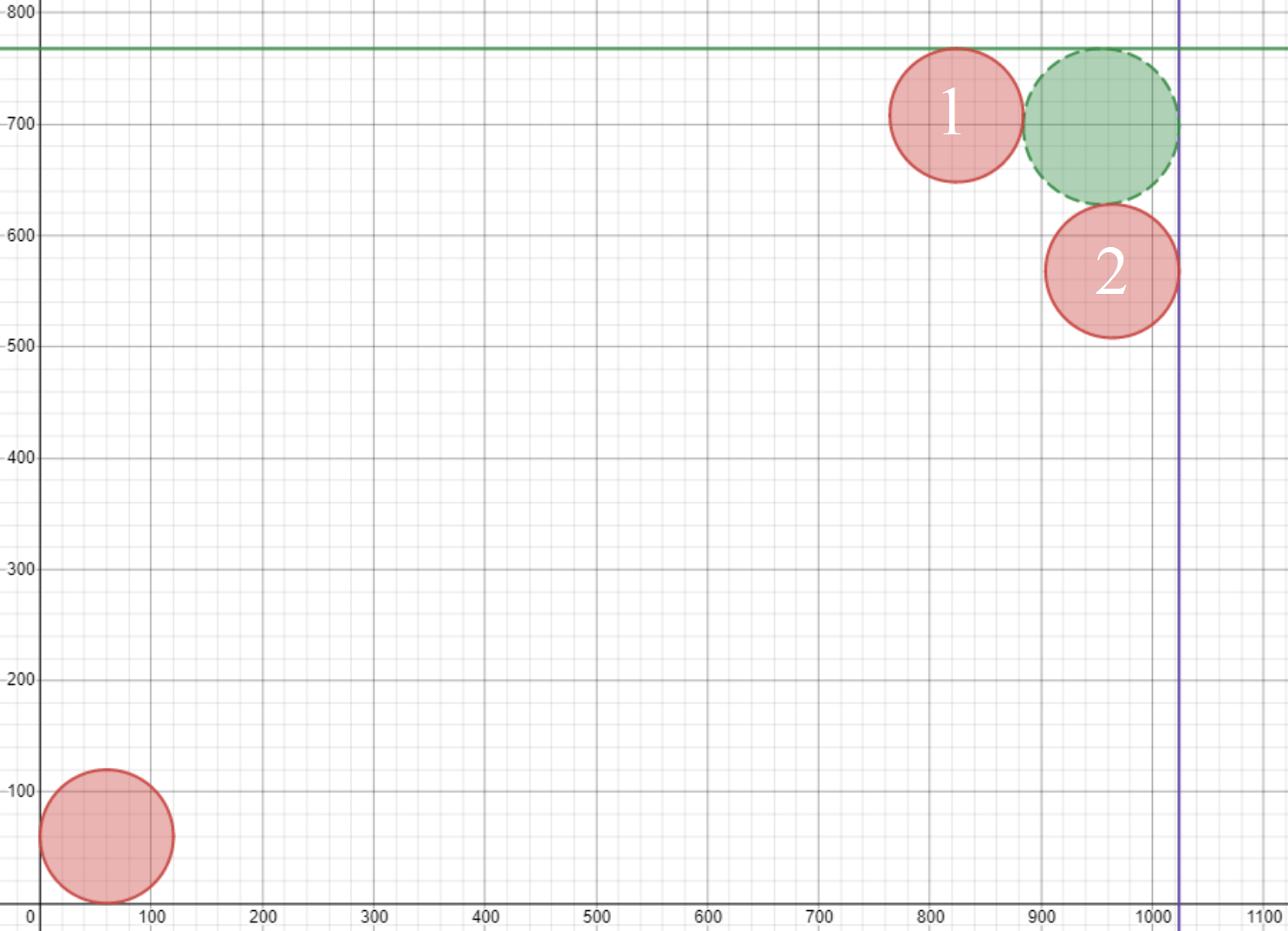
Задача испытуемого заключается в том, чтобы поднимать и опускать заданную траекторию красной окружности по вертикали с помощью колесика мыши, тем самым направляя «мышь в нору».

Траектория движения в виде параболы была выбрана биологами как оптимальная с точки зрения сложности и скорости прохождения: когда «Мышь» двигается по прямой, она очень быстро достигает цели и испытуемый не успевает подправить траекторию. Траектории более сложные, чем парабола, плохо подаются корректировки испытуемыми после пробуждения из глубокого сна.

Также задается «Зона ожидания»: квадратная область в начале траектории, в рамках которой программа запрещает управление движением «Мыши» (см. красный квадрат на Рисунке 10)

### Алгоритм генерации траектории мыши.

Для решения задачи построения параболических траекторий движения «Мышей», чтобы они не попадали в «Нору», я построил следующую математическую модель (см. Рисунок 5).



**Рисунок 5. Модель программы «Мыши» с критическими случаями**

Модель имеет параметры:

* Общие:
* Объект «Мышь»:
* Объект «Нора»:

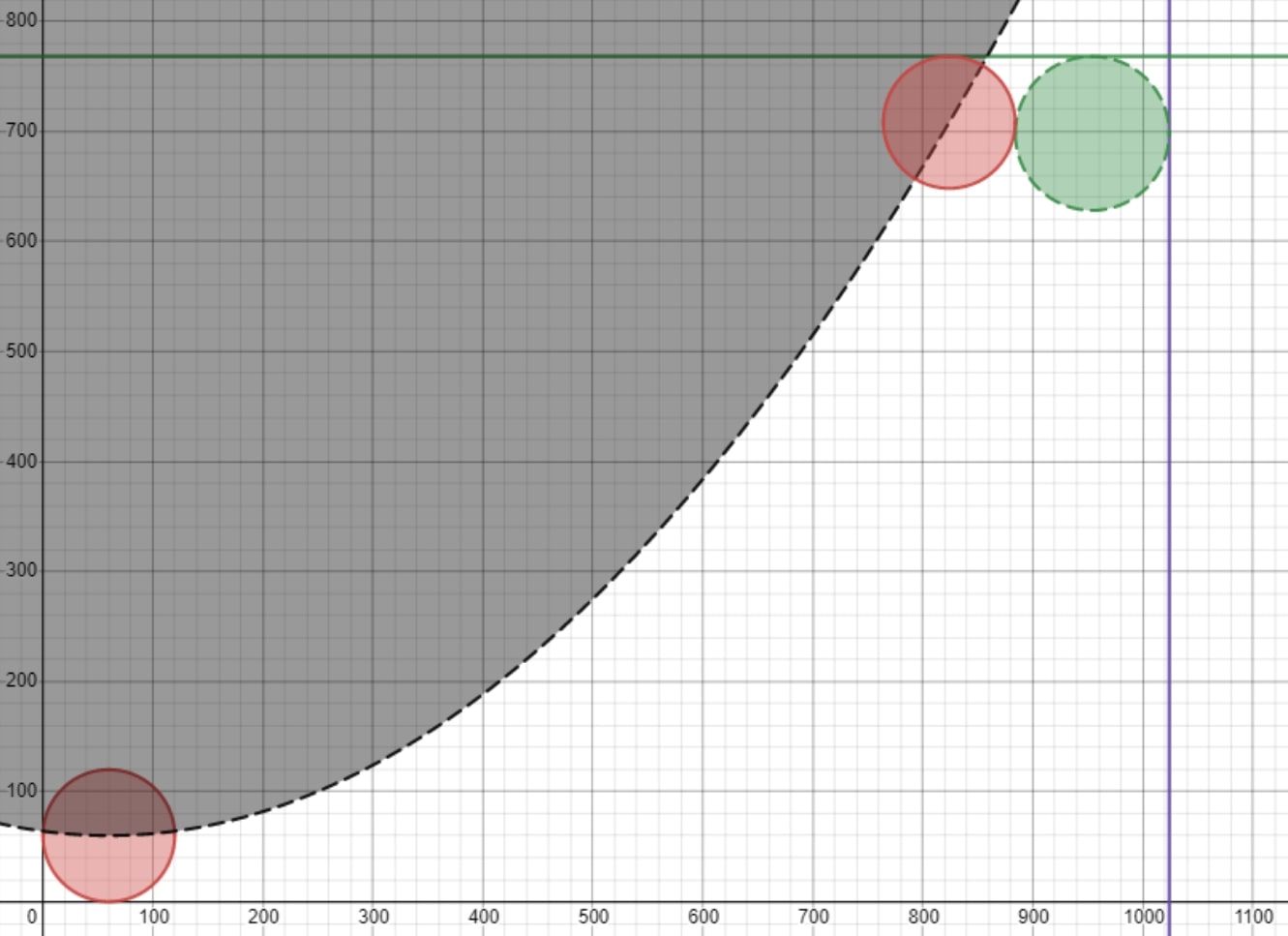
А также координаты критических случаев положения мышей:

(«Мышь» не попала в «Нору», но коснулась верхней стороны окна)

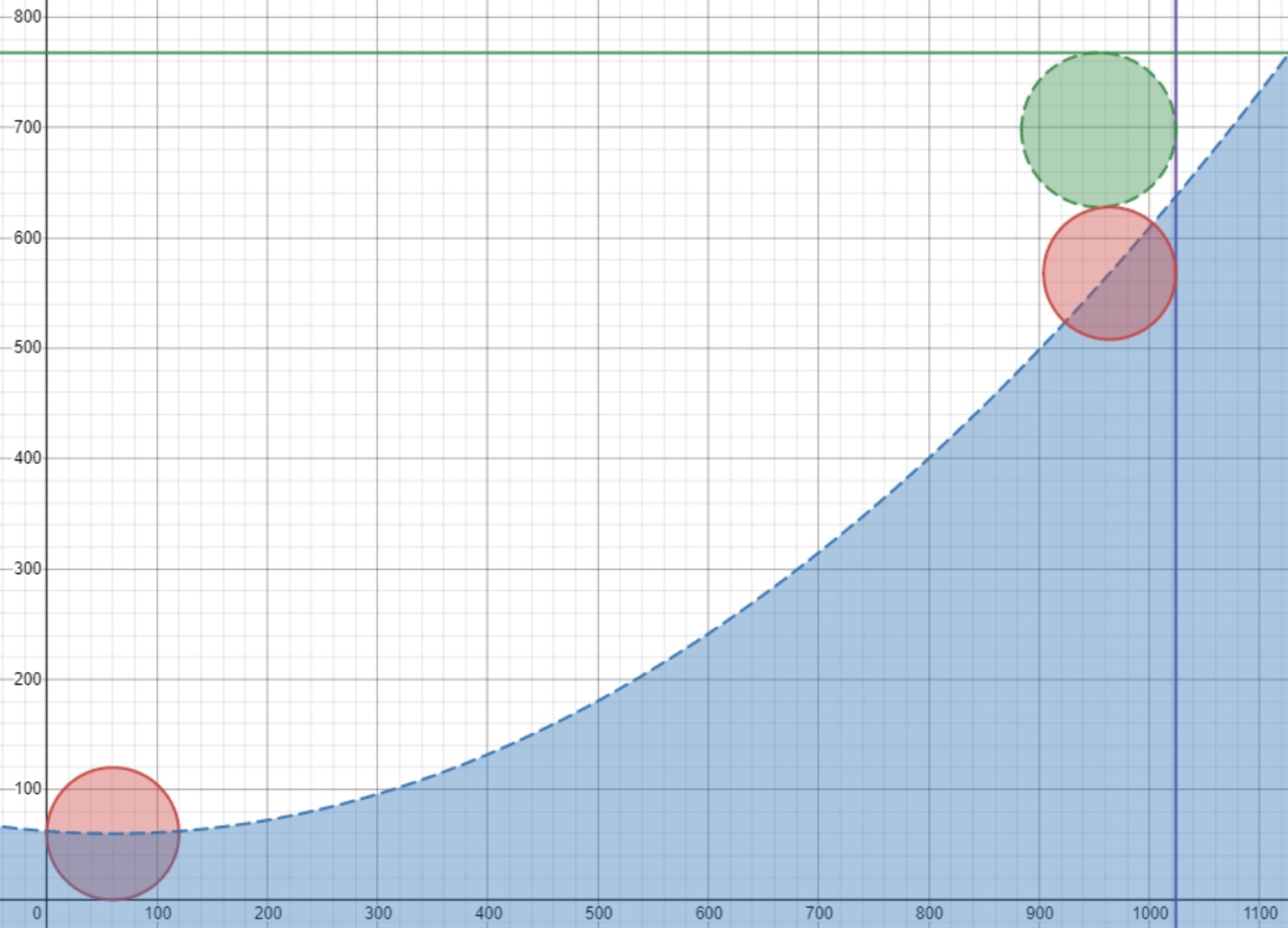
(«Мышь» не попала в «Нору», но коснулась правой стороны окна)

Для построения уравнения была использована общая формула построения параболы на графике: , где – параметр, влияющий на сближение ветвей параболы, – параметр, влияющий на смещение параболы вдоль оси , – параметр, влияющий на смещение параболы вдоль .

Всего возможно 4 варианта построения парабол из начальной точки в левом нижнем углу экрана, удовлетворяющих указанным условиям: см. ниже Рисунки 6-9.



**Рисунок 6. Ситуация №1 траектории «Мыши»**



**Рисунок 7. Ситуация №2 траектории “Мыши”**

Рассчитаем уравнения траекторий ситуаций №1 и №2 (Рисунки 6 и 7)

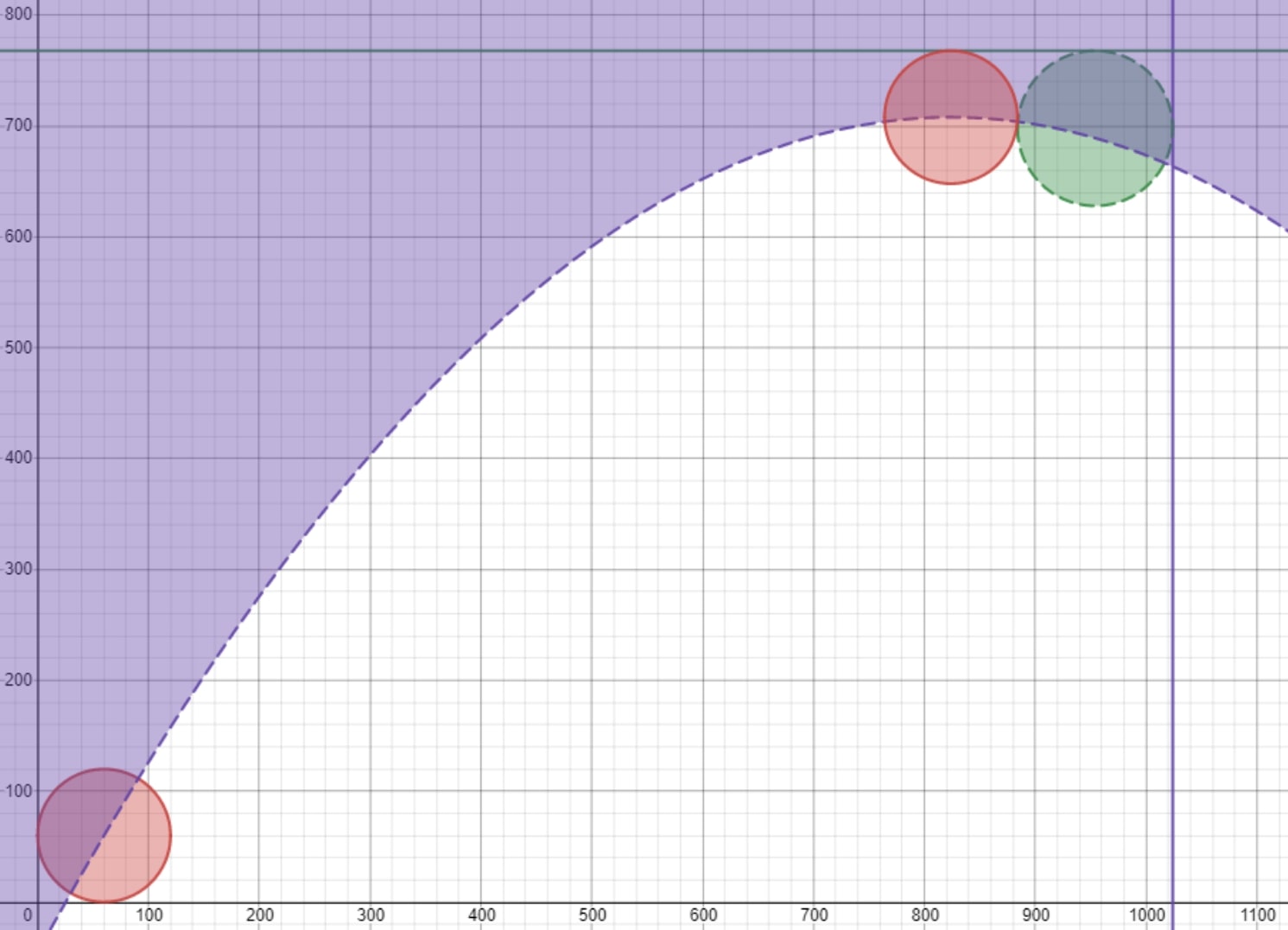
Так как в этих ситуациях вершина параболы находиться в одной точке, то формула таких парабол будет следующая:

Чтобы рассчитать , нужно подставить координаты критических точек и выразить коэффициенты.

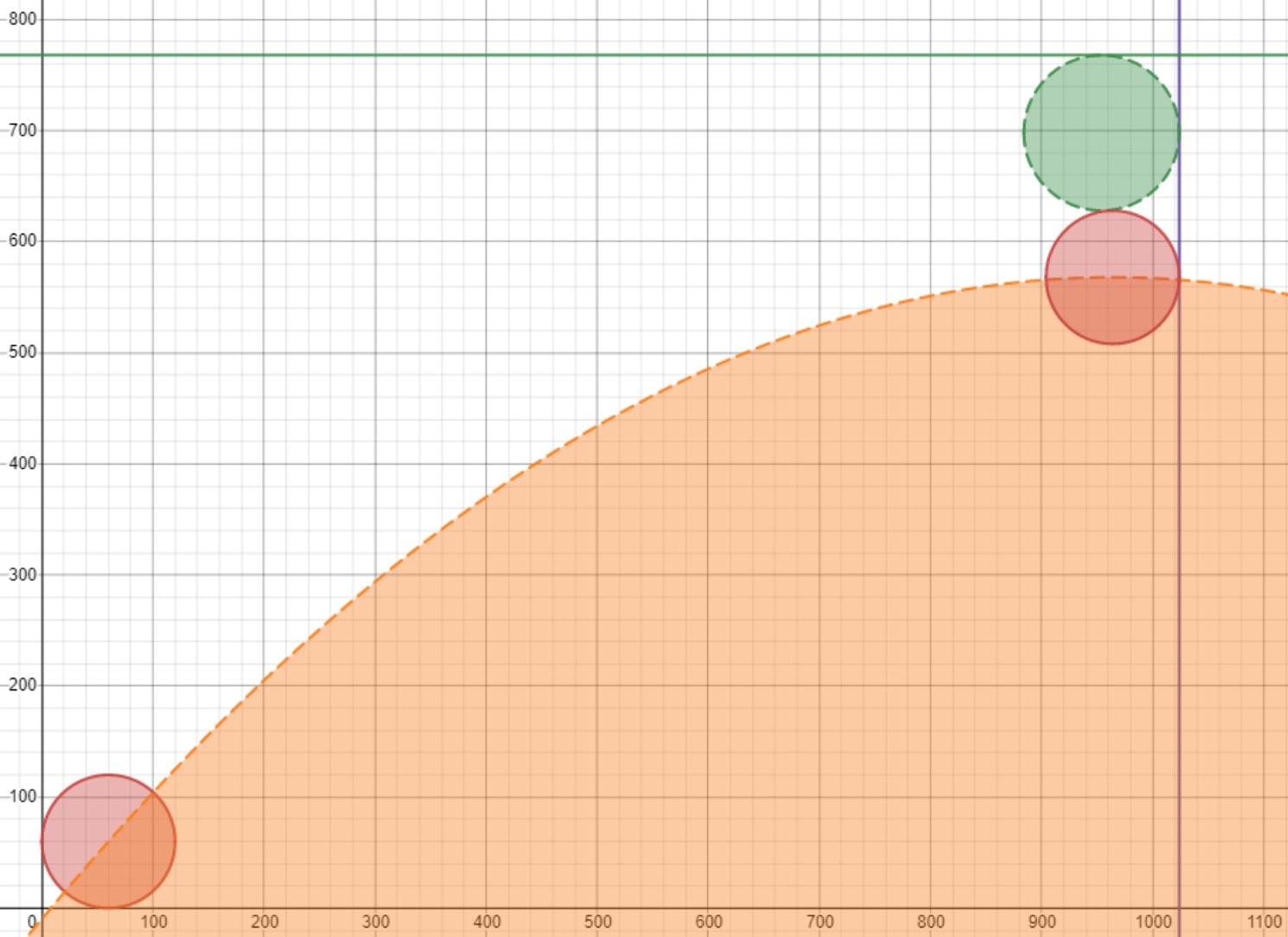
* **Ситуация №1**
* **Ситуация №2**

Также получим производную от найденной функции.

Проделаем те же действия с ситуациями №3 и №4 (см. Рисунки 8 и 9).



**Рисунок 8. Ситуация №3 траектории «Мыши»**



**Рисунок 9. Ситуация №4 траектории «Мыши»**

В отличие от первых двух ситуаций, у ситуаций №3 и №4 разные вершины парабол.

Поэтому для ситуации №3 будет формула:

А для ситуации №4:

Подставим в каждую формулу изначальные координаты мыши и критических точек и выразим коэффициенты.

* **Ситуация №3**
* **Ситуация №4**

Приведенные выше формулы для расчета коэффициентов k1, k2, k3, k4 позволяют узнать их граничные значения: максимум для k2 и k3, минимум для k1 и k4. Для генерации траекторий используются случайные значения от или от

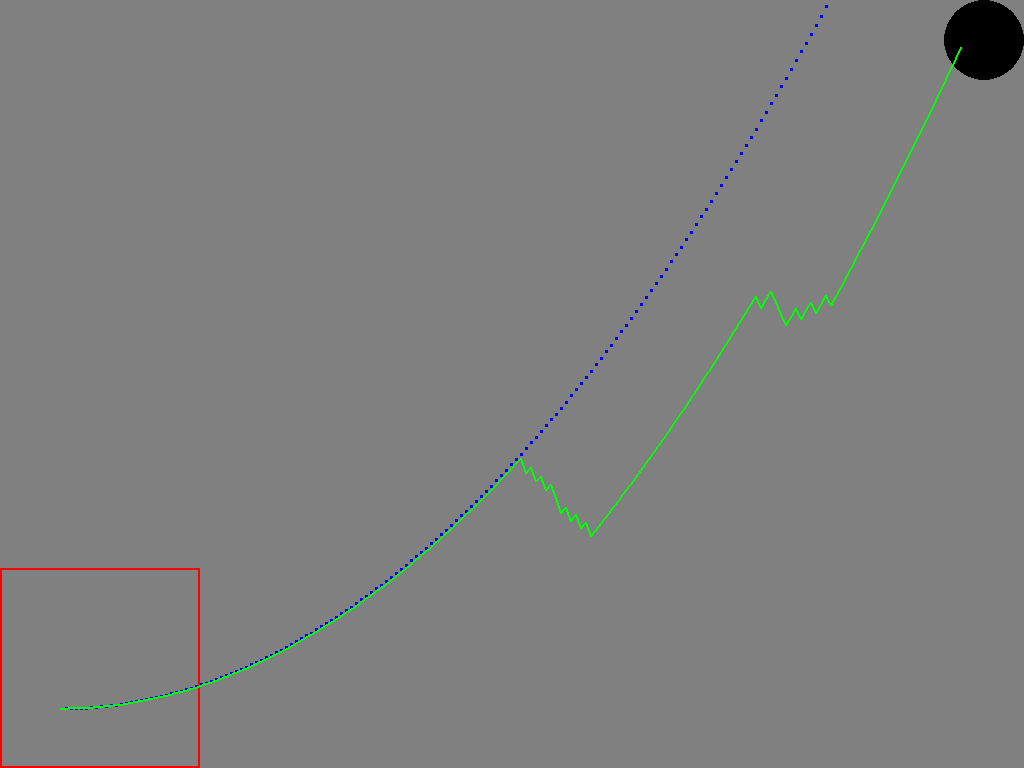
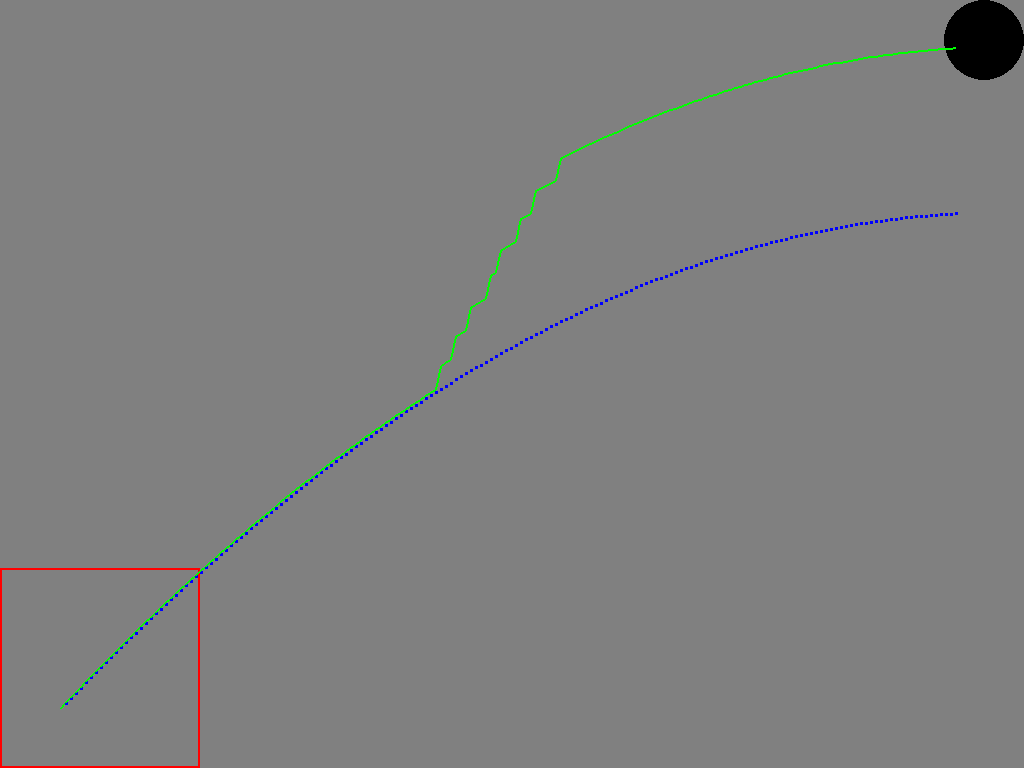
Имея формулы производной, можно рассчитывать приращение координаты Y «Мыши» при приращении её X координаты на шаг.

Выведенные формулы имеют в качестве параметров размер экрана и объектов («Мыши» и «Норы»). Поэтому их можно использовать при любом разрешении экрана и любых размерах объектов.

### Интерпретация результатов эксперимента

После завершения эксперимента программа создает директорию с названием эксперимента, датой и временем начала эксперимента. В ней создаются:

* Директория “log\_img”, в которой хранятся файлы изображений траекторий: зеленой (управляем испытуемым) и синей (заданной программой) (см. Рисунок 10).



**Рисунок 10. Примеры снимок траекторий**

* Директория “log\_txt”, где хранятся файлы с табличным представлением координат траекторий c одного опыта (см. Таблица 2).

Таблица 2. Пример файла с координатами траекторий

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Траектории | Разрешение окна: 1024x768 | | | Частота (в секундах): 0.25 | |
| Зеленая | | | Синяя | | |
| X | | Y | X | | Y |
| 215 | | 510 | 215 | | 511 |
| 370 | | 347 | 370 | | 349 |
| 525 | | 217 | 525 | | 220 |
| 680 | | 260 | 680 | | 125 |
| 835 | | 78 | 835 | | 63 |

* Файл “main\_log”, в котором храниться главный отчет по проведенному эксперименту, а именно количество добравшихся и пропавших мышей и отчет по каждому проведенному опыту: номер опыта, попала или не попала «мышь в нору», длины траекторий, их разница и координата касания с «норой» (см. Таблица 3).

Таблица 3. Пример файла с главным отчетом по проведенному эксперименту

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Мыши: | Добравшиеся | | Пропавшие | | Разрешение окна: 1024x768 | | |
|  | 2 | | 2 | |  | Финиш | |
| Опыт | Попал? | Длина: | Синий | Зеленый | Разница | X | Y |
| 1 | True |  | 1374 | 998 | 376 | 945 | 81 |
| 2 | False |  | 1233 | 978 | 255 | 925 | 42 |
| 3 | False |  | 1314 | 1144 | 170 | 950 | 80 |
| 4 | True |  | 1364 | 1143 | 221 | 930 | 53 |

Полный алгоритм работы программы «Мыши» в блок-схемах представлен в Приложении 4 и 5.

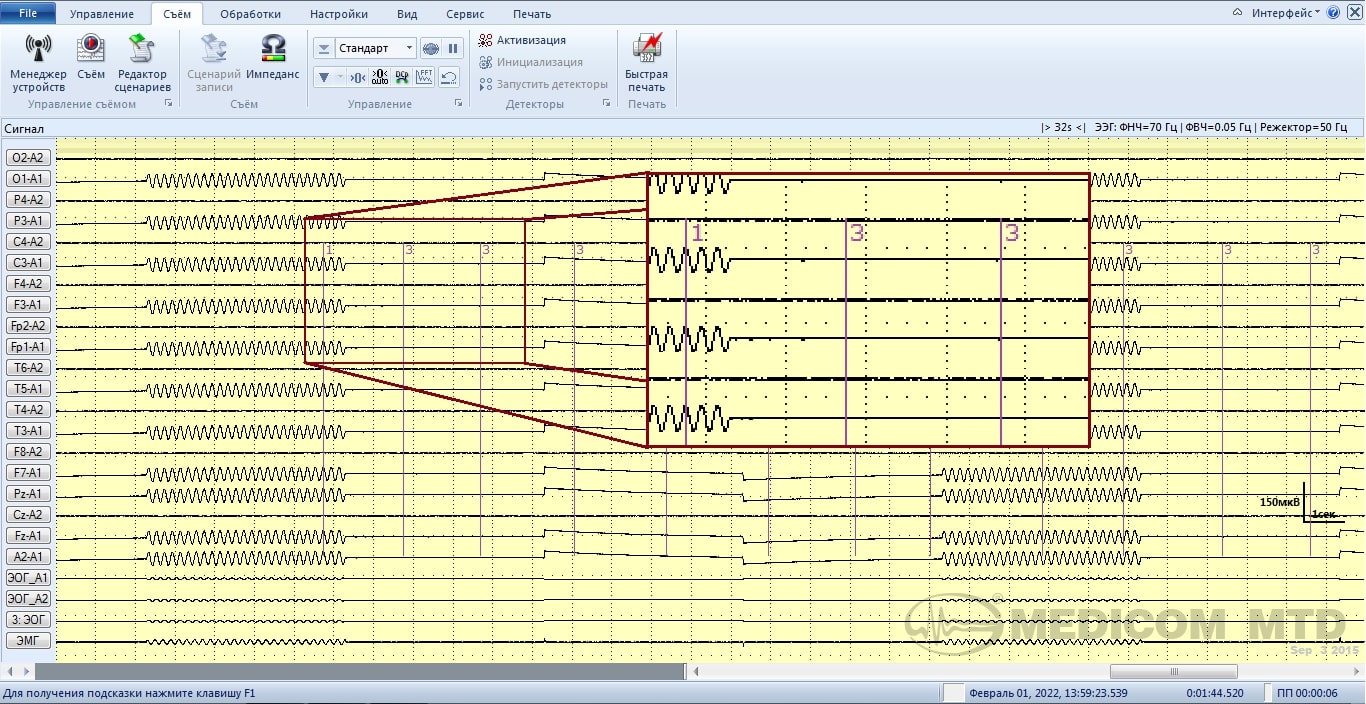
## 4. Реализация интеграции приложения с электроэнцефалограф-регистратором.

В процессе эксперимента записывается электроэнцефалограмма (ЭЭГ) испытуемого. Для полной картины эксперимента необходимо видеть этапы опыта непосредственно на ЭЭГ.



**Рисунок 11. Программное обеспечение и оборудование для ЭЭГ.**

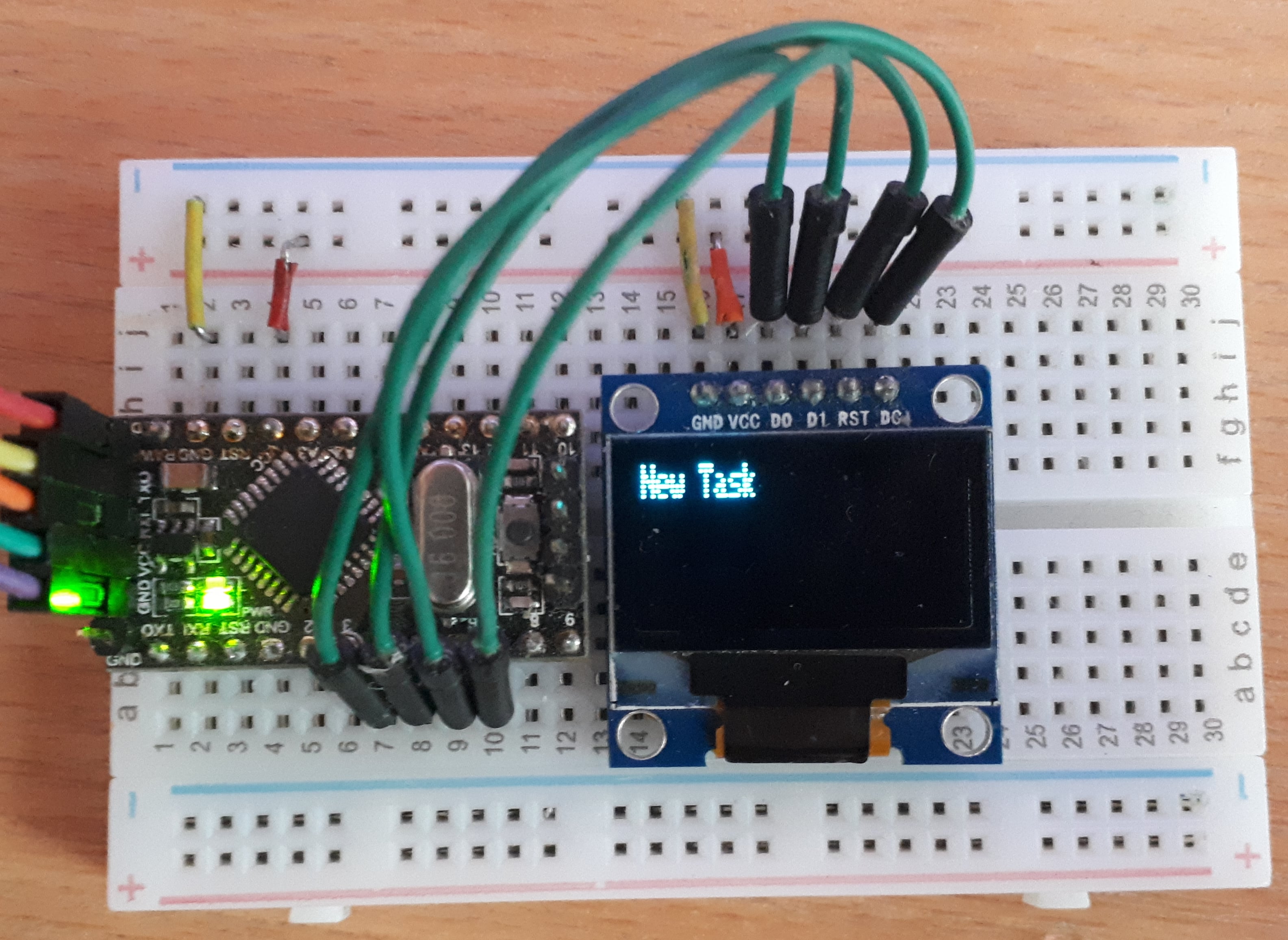
В связи с этим биологи попросили добавить в приложение интеграцию с электроэнцефалограф-регистратором «Энцефалан-ЭЭГР-19/26» (см. рис. 11): приложение должно посылать определенный набор байтов по USB протоколу в моменты начала выполнения эксперимента, начала нового опыта «Мыши» и начало нового опыта «Задачки». В начале эксперимента приложение должно поставить метку “1”, в началах новых опытов с “Задачками” – “2”, в началах новых опытов с “Мышами” – “3” (см. Рисунок 12).



**Рисунок 12. Требуемый вид электроэнцефалограммы**

Так как у меня не было возможности работать с «Энцефаланом» напрямую, для разработки и тестирования функционала пришлось создать эмулирующее устройство, которое примерно повторяло функционал модуля выставления меток на ЭЭГ.

Для этого я использовал программируемый модуль Arduino Pro Mini[8] с OLED дисплеем (см. Рисунок 13).



**Рисунок 13. Эмулирующее устройство**

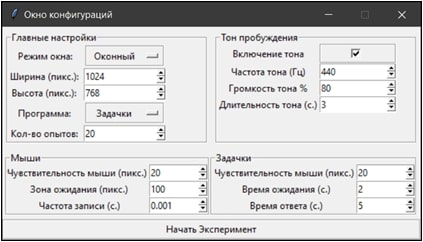
Это устройство, получает определенные байты с порта USB и отображает на дисплее соответствующую информацию, а именно:

* При получении байта “1” на экране отображается сообщение “Start”, обозначающее начала эксперимента.
* При получении байта “2” – сообщение “New Mouse”, обозначающее начала нового опыта в подпрограмме «Мыши».
* При получении байта “3” – сообщение “New Task”, обозначающее начала нового опыта в подпрограмме «Задачки».

Таким образом, я смог отладить свое приложение, не имея доступа к конечному устройству в процессе разработки, и минимизировать количество тестовых запусков приложения на месте (всего до 3-х запусков).

## Реализация графического интерфейса для настройки программ и сборка в одно приложение.

Для обеспечения большего удобства использования было решено связать две вышеперечисленные программы в одну, добавив графический интерфейс для их настройки (см. Рисунок 14).



**Рисунок 14. Вид конфигурационного окна.**

Для реализации этой задачи я использовал библиотеку Tkinter, с помощью которой можно создавать простые графические интерфейсы.

Окно конфигурации позволяет задавать общие настройки для программ такие как:

* Режим отображения: оконный или полноэкранный режим
* Разрешение отображения: ширина и высота окна в пикселях
* Выбор программы для выполнения: «Задачки» или «Мыши»
* Количество опытов в одном эксперименте
* Тон пробуждения: включить или отключить
* Частота тона в герцах
* Громкость тона в процентах
* Длительность тона в секундах

А также определенные настройки для соответствующих программ.

* Для программы «Задачки»:
  + Чувствительность мыши (кол-во пикселей за один щелчок колесика)
  + Время ожидания перед опытом в секундах
  + Время ответа в опыте
* Для программы «Мыши»:
  + Чувствительность мыши (кол-во пикселей за один щелчок колесика)
  + Зона ожидания (сторона квадратной зоны запрета управления в пикселях)
  + Частота записи координат траекторий в секундах

Заданные настройки сохраняются в файл и используются при следующем запуске программы.

Полный алгоритм работы общей программы в блок-схемах представлен в Приложении 1.

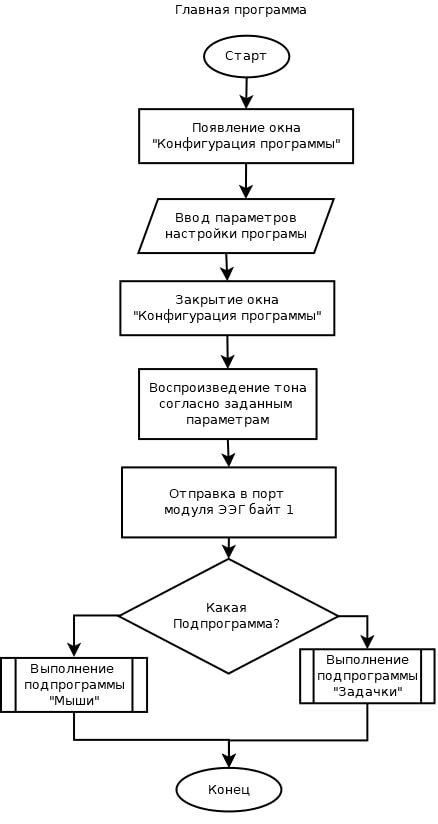
# Заключение

Данный проект реализован. На сегодняшний день было поставлено несколько экспериментов с использованием этого программного обеспечения.

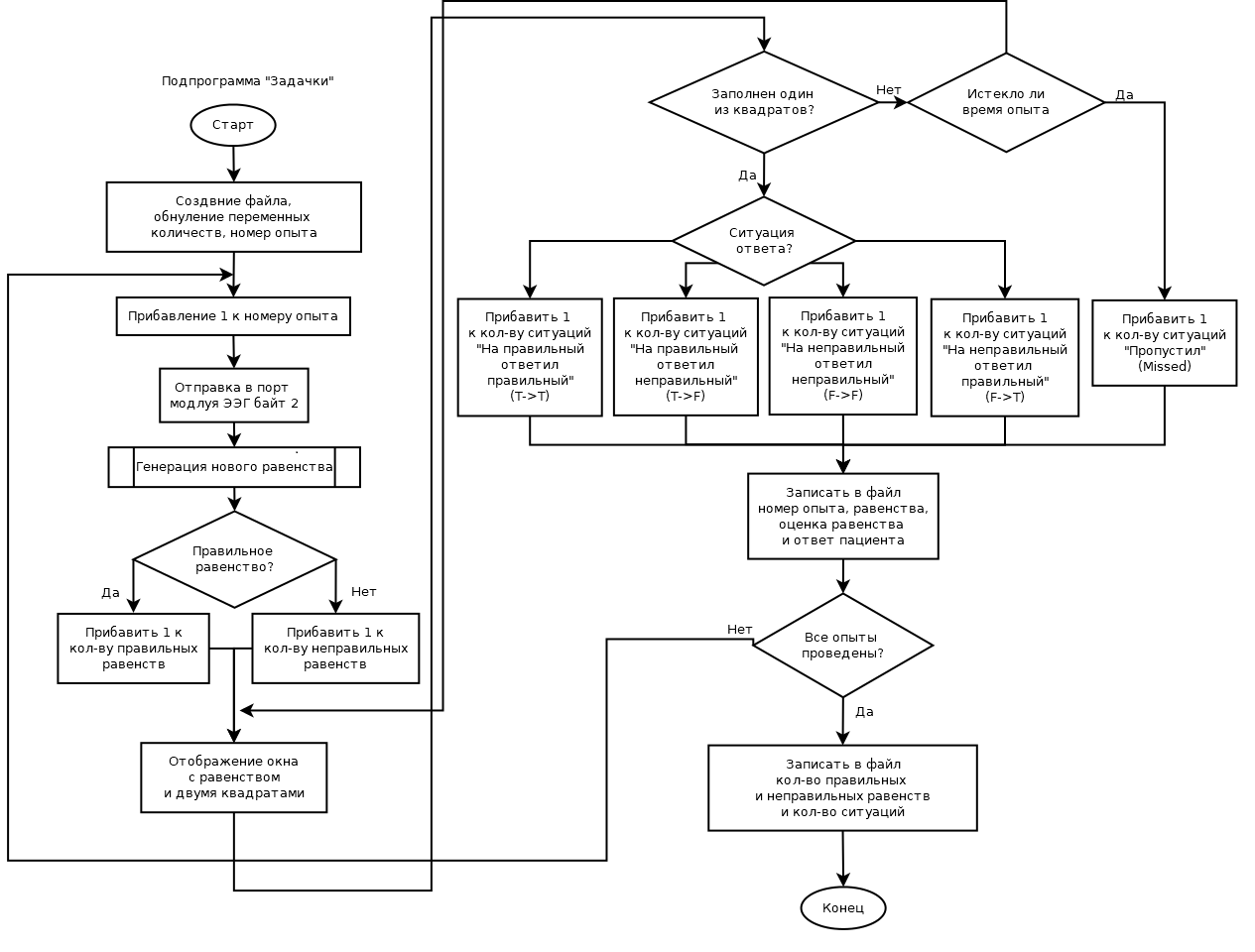
Исследовательский проект биологов вышел на стадию массовых экспериментов с участием добровольцев, поэтому программный продукт еще будет развиваться по мере появлений новых требований для повышения точности результатов экспериментов.

**Файлы проекта** с исходным кодом программы, исполняемым файлом, текстом работы и мультимедийной презентацией доступны по ссылке: <http://www.github.com/ClausSolov7939/SleepResearch>

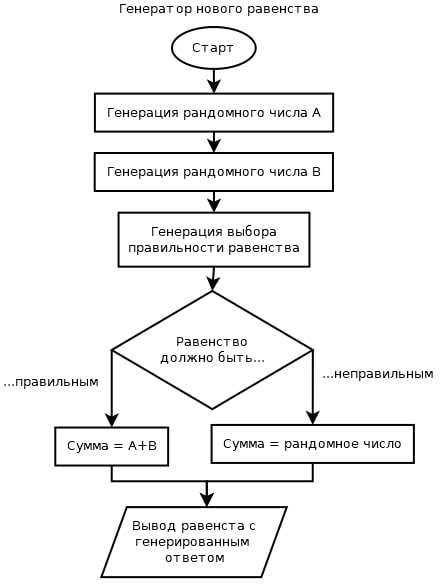
# Приложения



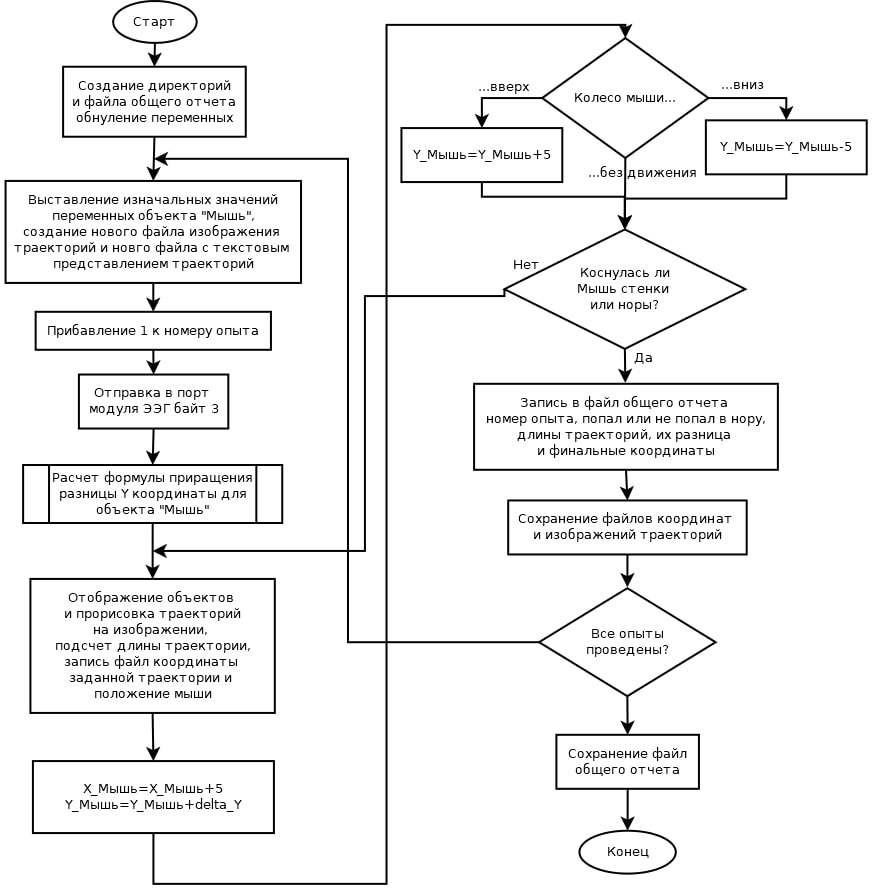
**Приложение 1. Алгоритм работы общей программы**

****

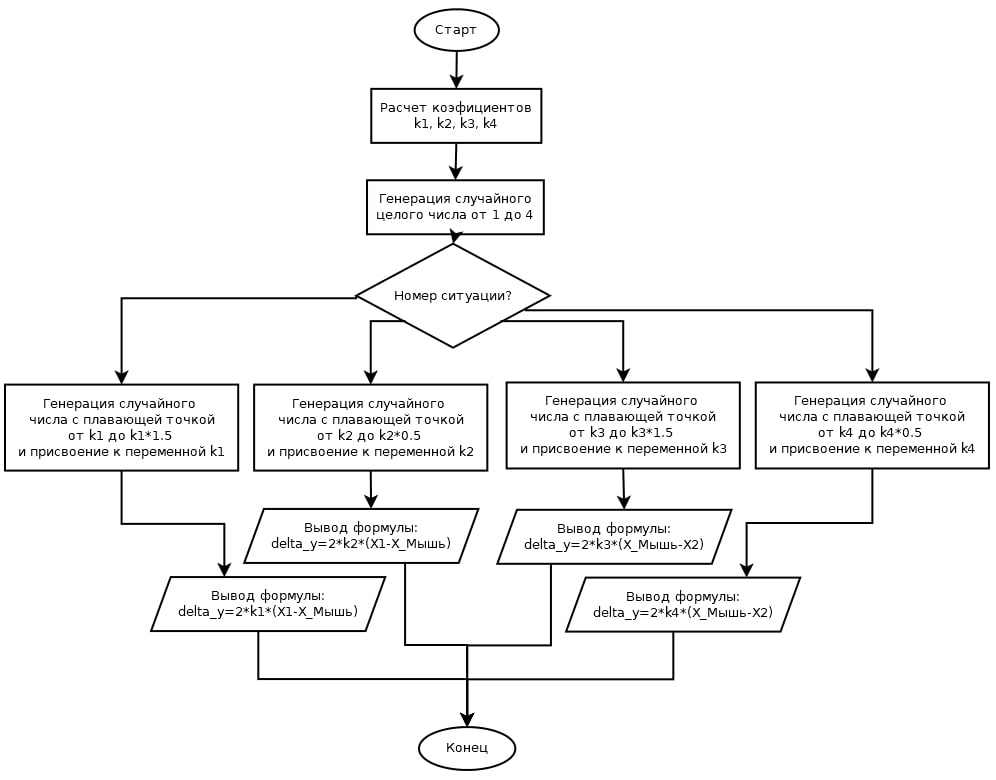
**Приложение 2. Алгоритм работы программы «Задачки»**



**Приложение 3. Алгоритм расчета новых равенств для программы «Задачки»**



**Приложение 4. Алгоритм работы программы «Мыши»**



**Приложение 5. Алгоритм расчета формулы новых траекторий для программы «Мыши»**

# Библиографический список

1. Язык программирования Python3: Официальный сайт. – URL: <https://www.python.org/>
2. Графическая библиотека Pygame: Официальный сайт документации. – URL: <https://www.pygame.org/docs/>
3. Библиотека графических интерфейсов пользователя Tkinter: Официальный сайт документации. – URL: <https://docs.python.org/3/library/tkinter.html>
4. Библиотека интерфейсов операционной системы OS: Официальный сайт документации. – URL: <https://docs.python.org/3/library/os.html>
5. Библиотека для работы со временем Time: Официальный сайт документации. – URL: <https://docs.python.org/3/library/time.html>
6. Библиотека для работы с внешними устройствами Serial: Официальный сайт документации. – URL: <https://pyserial.readthedocs.io/>
7. "Энцефалан-ЭЭГ":Официальный сайт. – URL**:**

<http://medicom-mtd.com/htm/Study/EEG/study_eeg_temp.html>

1. "Arduino":Официальный сайт. – URL**:** <https://www.arduino.cc>