**Решение диофантовых уравнений**

Российская федерация,

Государственное бюджетное образовательное учреждение «Школа 1158»

**Создал: Зуйков Андрей**, 10 класс

Научный руководитель: Коваленко Игорь Борисович,

учитель информатики «Школа 1158»

**Научная статья**

**Введение**

Вся современная криптография построена на диофантовых уравнениях. Взлом RSA -- это и есть решение диофантовых уравнений. Взлом любого криптографического алгоритма открытого распределения ключа, цифровой подписи и т.п. сводится к решению какого-то диофантового уравнения. И вся современная криптография держится на том, что разработчики этих криптографических алгоритмов могут верно оценить сложность решения соответствующих (диофантовых в итоге) уравнений. Ведь эти уравнения решаются. Особенность в том, что, изменяя определённые параметры алгоритма, можно сделать так, что решение соответствующего уравнения будет сколь угодно дорогим по сравнению со стоимостью использования алгоритма. Дорогим в прямом смысле слова - это совокупность затраченной машинной мощности - время и потребляемая энергия.

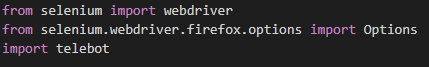
**Цель работы**: на основе теоретического изучения курса программирования разработать программу, которая будет решать диофантовы уравнения.

**Задачи**:

1. Углубленно изучить Python;
2. Изучить основы библиотеки TeleBot;
3. Написать программу для решения диофантовых уравнений;
   1. Разобраться как работает библиотека Selenium
   2. Написать программу получения решений с сайта <https://planetcalc.ru/3303/>
   3. Научиться подставлять значения и получать данные
4. Вывести решение в Telegram

Реализация

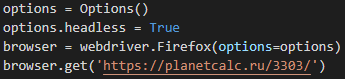
1. Подключение нужных библиотек
   1. Selenium
   2. TeleBot



1. Инициализируем Telegram бота



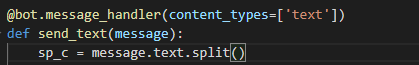
1. Настраиваем geckodriver, через Selenium и переходим на сайт



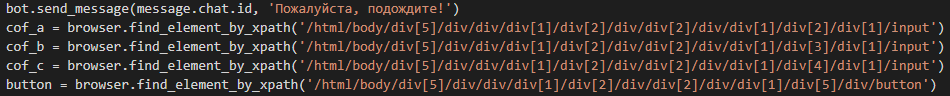
1. Обрабатываем команды /start и /help



1. Обрабатываем сообщения и получаем коэффициенты



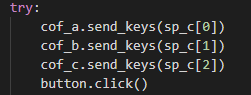
1. Получаем XPATH кнопок и полей ввода коэффициентов



1. Очищаем поля ввода



1. Отправляем значения и нажимаем на кнопку



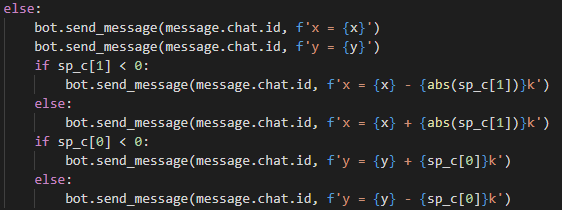
1. Получаем значения

****

1. Проверяем на наличие решений



1. Если найдено решение, отправляем решение и подставляем в формулу множеств. Отправляем множества



1. Если введены неправильные значения выводим ошибку и запускаем бота

