**ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ ГОРОДА МОСКВЫ**

**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ГОРОДА МОСКВЫ**

**«ШКОЛА № 1504»**

Тренажер решения уравнения основной школы

Работа

Ученицы 5 класса «Э»

Сойчик Аливии Николаевны

Руководитель проекта

Твердохлеб Юлия Алексеевна

Москва

2025 г.

**Обоснование выбора, актуальность**

Мой проект — это практическая и исследовательская работа на тему решения линейных уравнений основной школы.

Продуктом моего проекта является тренажер решения уравнений. Это Web – приложение, которое помогает школьникам потренироваться решать линейные уравнения. По задумке, моё приложение не будет решать за школьника уравнение, а в место этого разделит процесс решения на этапы и выдаст подсказку по решению для каждого этапа.

Пятиклассникам только начинающим осваивать уравнения будет полезным иметь под рукой тренажер для решения уравнений. Задумка в том, что он не решит за них уравнения, а только подскажет шаги его решения (алгоритм).

При разработке продукта проекта – тренажера решения равнений, я сделала выбор в пользу разработки Web приложения. Мой выбор связан с тем, что это удобный способ запустить приложение на любом устройстве (телефон, планшет, компьютер) где есть браузер. Также для разработки приложения я выбрала язык программирования Python, так как это распространённый язык программирования с большим количеством примеров, и легкий для понимания для моего уровня.

**Постановка цели**

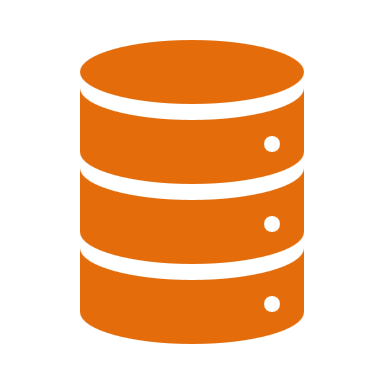
Целью проекта является разработка Web приложения “Тренажер решения уравнений”.

Данное Web приложение предназначено для школьников 5-7 классов, желающих научится быстро решать уравнения, изучив алгоритм их решения по шагам. Цель приложения, не решить за школьника уравнение, а провести его по всем этапам решения, с подсказками на каждом этапе и проверки полученного им результата на каждом этапе.

При выборе ресурсов и средств разработки я выбрала разработку Web – приложения, так как его можно запустить практически на любом устройстве (телефон, компьютер, планшет) где есть интернет, что будет очень удобно для школьников. Web – приложение запускается в браузере. Браузер – это программа для отображения Web сайтов. Наше Web-приложение это, по сути, маленький сайт. Хранятся сайты на серверах, значит в процессе разработки нам необходимо найти место (сервер), где мы будем хранить наше Web-приложение. Для хранения приложения я остановилась на сервисе хранения исходных кодов <https://github.com/>. Данный сервис предоставляет нам возможность хранить исходный код нашего Web – приложения, и будет служить сервером для его загрузки.



Ресурсы Web приложения



Ресурсы Web приложения

Пользователи

Интернет

Ресурсы Web приложения

Сервер

Рис. 1. Схема работы Web-приложений

При выборе языка программирования я остановилась на языке Python. с использованием промежуточной среды Brython для Web приложений. Библиотека Brython переводит программы написанные на Python в JavaScript прямо в браузере. Все что нам нужно – это подключить библиотеку Brython и загрузить на странницу наше приложение на Python.

Также для быстрой разработки интерфейса я использовала уже готовые модули для среды Brython. Вот они:

1. Модуль SimpleEquation – этот модуль содержит компонент - поле ввода простого уравнения. Этот компонент удобен тем, что он проверят правильность ввода уравнения за нас. Также используя его очень удобно организовать обход по элементам уравнения и выделения цветом частей уравнения.

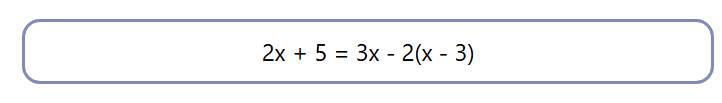


Рис.2. Компонент SimpleEquation на странице Web приложения

1. Модуль Controls – из этого модуля мы будем использовать компоненты Button и Label. Button – это кнопка, при нажатии на которую мы запустим нужные нам действия. Label – это текстовое поле, куда мы будем выводить информацию про тот или иной этап.



Рис.3. Компоненты Button на странице Web приложения

**Этапы**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Этап** | **Задача** | **Промежуточный результат** |
| 1 | Выбрать алгоритм решения уравнений | Изучить литература и информацию про решения уравнений и выбрать оптимальный алгоритм | См. Приложение 1. |
| 2 | Выбор языка программирования, ресурсов и среды разработки | Определить на каком языке мне будет удобно работать, выбрать ресурсы, библиотеки, хостинг для Web приложения. | Мой выбор пал на язык Python, библиотеку для Web приложения Brython. Также для хостинга приложения был выбран сервис <https://github.com/> |
| 3 | Разбить разработку приложения на части | Продумать структуру приложения.  Разбить задачу на под-алгоритмы | См. Приложения 2-3 |
| 4 | Разработка алгоритмов для каждой части задачи | Продумать каждую часть задачи и нарисовать блок схемы для них | См. Приложения 4-11 |
| 5 | Разработка Web приложения | Написание кода на языке Python для каждой части задачи, и написание основанного кода для взаимодействия всех частей. | См. Приложения 12-13 |
| 6 | Тестирование Web приложения | Протестировать каждую часть задачи и приложение целиком. | Тестирование выполнено |

**Подведение итогов**

Изучив информацию по решению уравнений, был выбран алгоритм представленный в приложении 1. Было разработано Web приложение “Тренажер решения уравнений”, которое доступно по адресу <https://alivia-school.github.io/project-equations/>.

Данное приложение может иметь практическое применение среди школьников 5-7 классов, для изучения алгоритма решения линейных уравнений и как помощник для их решений в будущем.

­­­

**Приложение 1**. Блок схема алгоритма решения уравнений

Нет

Уравнение содержит скобки?

Да

Раскрыть скобки

В левой части только неизвестные, а в правой только числа?

Да

Нет

Выполнить перенос

Нет

Требуется упрощение?

Да

Упростить

Да

Да

Неизвестное найдено?

Нет

Вычислить неизвестное

**Приложение 2**. Структура приложения

|  |  |  |
| --- | --- | --- |
| Папка или файл | | Описание |
| assets/ | | Папка с ресурсами для Web приложения |
| libs/ | | Папка со сторонними библиотеками для Web приложения |
| py/ | | Пака с нашими файлами .py для Web приложения |
|  | main.py | Это основной файл приложения. В нем мы объявляем компоненты нашего интерфейса, описываем действия при событиях (нажатие кнопок), и описываем основную логики этапов решения уравнения. |
|  | functions.py | Это файл с функциями в которых выполняются алгоритмы для каждой части задачи (например определение нужен ли тот или иной этап, алгоритмы выделения частей уравнения для этапов, расчет результата для каждого этапа). |
|  | tips.py | Файл с текстами для пояснений к каждому этапу. |
| index.html | | Файл разметки для Web приложения. Именно он загружается браузером и в нем прописано что браузер должен загрузить и выполнить для работы нашего Web приложения. |

**Приложение 3.** Список частей задачи

1. Для первого этапа (Раскрыть скобки)
   1. Определить, нужно ли раскрывать скобки в уравнении
   2. Выделить скобки в уравнении
   3. Раскрыть скобки в уравнении, чтобы проверить наш результат с результатом пользователя
2. Для второго этапа (Перенести все неизвестные компоненты влево, а известные вправо)
   1. Определить, нужно ли выполнять перенос в уравнении
   2. Выделить в уравнении красным то, что нужно переносить влево, а то, что вправо выделить синим
   3. Выполнить перенос в уравнении, чтобы проверить наш результат с результатом пользователя
3. Для третьего этапа (Упростить обе части уравнения)
   1. Определить, нужно ли упрощать уравнение
   2. Выделить знаки, с помощью которых мы будем упрощать уравнение
   3. Выполнить упрощение уравнения, чтобы проверить наш результат с результатом пользователя
4. Для четвертого этапа (Вычислить неизвестное)
   1. Определить, нужно ли вычислять неизвестное в уравнении
   2. Вычислить неизвестное, чтобы проверить наш результат с результатом пользователя

**Приложение 4.** Схема алгоритма определения нужно ли раскрывать скобки в уравнении

ЕстьСкобки(Уравнение)

Взять Элемент из Уравнение

Нет

Элемент.Тип == Скобка

Да

False

True

**Приложение 5.** Схема алгоритма раскрытия скобок в уравнении

РаскрытьСкобки(Уравнение)

Поместить Элемент в Результат

ЭлементПередСкобкой = Элемент

Нет

Элемент стоит перед скобкой?

Нет

Да

Умножить ЭлементПередСкобкой на Элемент и поместить в Результат

Нет

Мы внутри скобок?

Снять флаг что мы внутри скобок

Да

Нет

Элемент.Значение == ‘)‘

Выставить флаг что мы внутри скобок

Да

Элемент.Значение == ‘(‘

Взять Элемент из Уравнение

Результат = “”

Результат

**Приложение 6.** Схема алгоритма проверки нужен ли перенос элементов в уравнении

НуженПеренос(Уравнение)

True

Да

Нет

Элемент.Тип = Число

Элемент.Тип = Неизвестное

Взять Элемент из Уравнение.ЛеваяЧасть

Взять Элемент из Уравнение.ПраваяЧасть

Нет

Да

False

True

**Приложение 7.** Схема алгоритма переноса элементов в уравнении

Нет

Да

Добавить Элемент в РезультатЛевый с противоположным знаком

Да

Элемент.Тип == Число

Взять Элемент из Уравнение.ЛеваяЧасть

Добавить Элемент в РезультатЛевый

Да

Нет

Элемент.Тип = Неизвестное

РезультатПравый = «»

РезультатЛевый = «»

Перенос(Уравнение)

Взять Элемент из Уравнение.ЛеваяЧасть

Да

Добавить Элемент в РезультатПравый

РезультатЛевый + «=» + РезультатПравый

Нет

Элемент.Тип == Число

Элемент.Тип == Неизвестное

Взять Элемент из Уравнение.ПраваяЧасть

Добавить Элемент в РезультатПравый

Нет

**Приложение 8.** Схема алгоритма проверки нужно ли упрощение уравнения

Да

True

Колво(Уравнение.ЛеваяЧасть) >=3

НуженоУпрощение(Уравнение)

False

True

Колво(Уравнение.ПраваяЧасть) >=3

Нет

Да

Нет

**Приложение 9.** Схема алгоритма упрощения уравнения

РезультатПравый = 0

РезультатЛевый = 0

Нет

Да

Взять Элемент из Уравнение.ЛеваяЧасть

Упрощение(Уравнение)

Элемент.Тип == Неизвестное

Добавить Элемент.ЗначениеПередНеизвестным в РезультатЛевый учитывая знак перед ним

Взять Элемент из Уравнение.ПраваяЧасть

Элемент.Тип == Число

Да

Нет

Добавить Элемент.Значение в РезультатПравый учитывая знак перед ним

КакСтрока(РезультатЛевый) + Уравнение.Неизвестное + «=» + КакСтрока(РезультатПравый)

**Приложение 10.** Схема алгоритма проверки нужно ли вычисление неизвестного

НуженоВычислениеНеизвестного(Уравнение)

Нет

Уравнение.ЛеваяЧасть[0]. Значение == «-»

Да

True

Нет

Уравнение.ПраваяЧасть[0]. ЗначениеПередНеизвестным > 1

Да

False

True

**Приложение 11.** Схема алгоритма вычисления неизвестного

Нет

Да

ЗначениеПравойЧасти % ЗначениеЛевойЧасти == 0

ЗначениеПравойЧасти= Уравнение.ЛеваяЧасть[0].Значение

ЗначениеПравойЧасти = -Уравнение.ЛеваяЧасть[1].Значение

Нет

Да

Уравнение.ПраваяЧасть[0]. Значение == «-»

ЗначениеЛевойЧасти= Уравнение.ЛеваяЧасть[0].ЗначениеПередНеизвестным

ЗначениеЛевойЧасти = -Уравнение.ЛеваяЧасть[1].ЗначениеПередНеизвестным

ВычислениеНеизвестного(Уравнение)

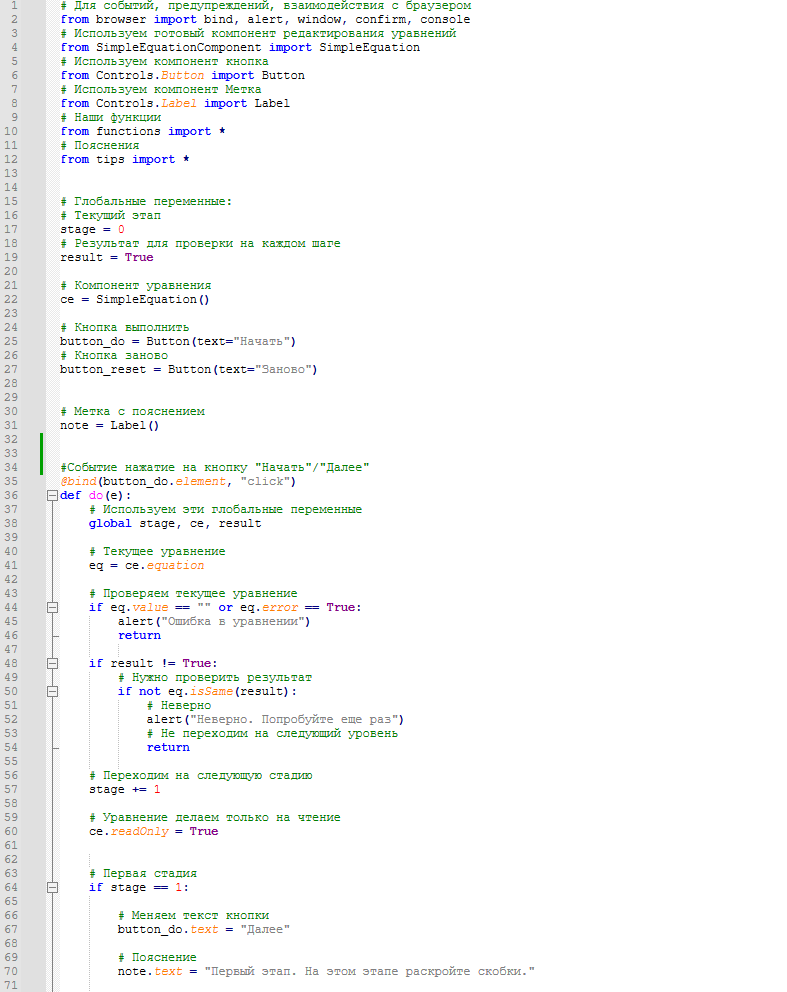
Нет

Да

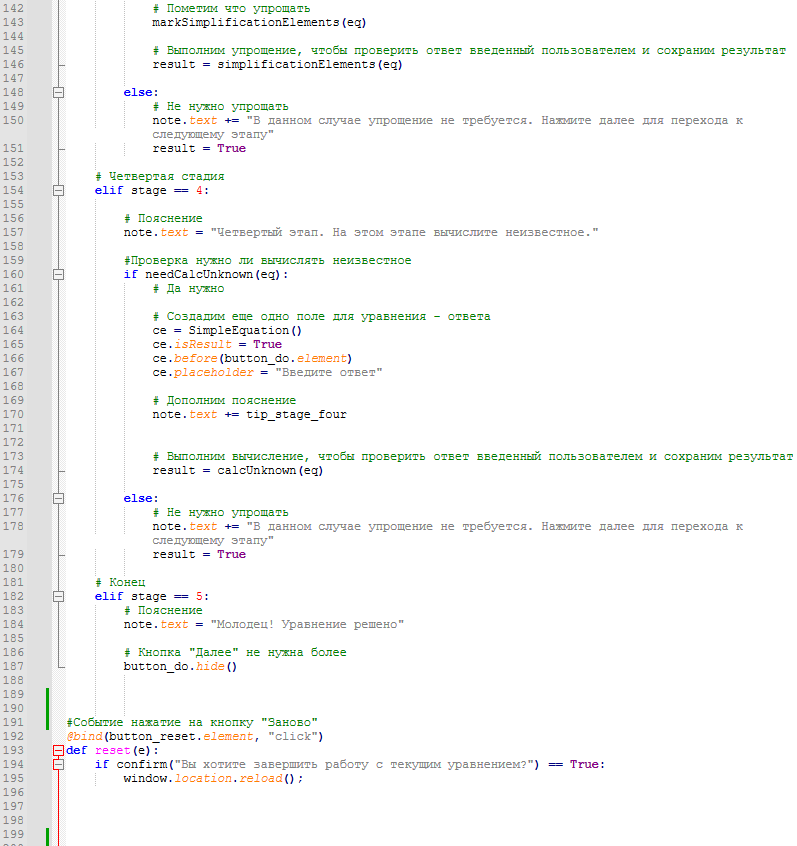
Уравнение.ЛеваяЧасть[0]. Значение == «-»

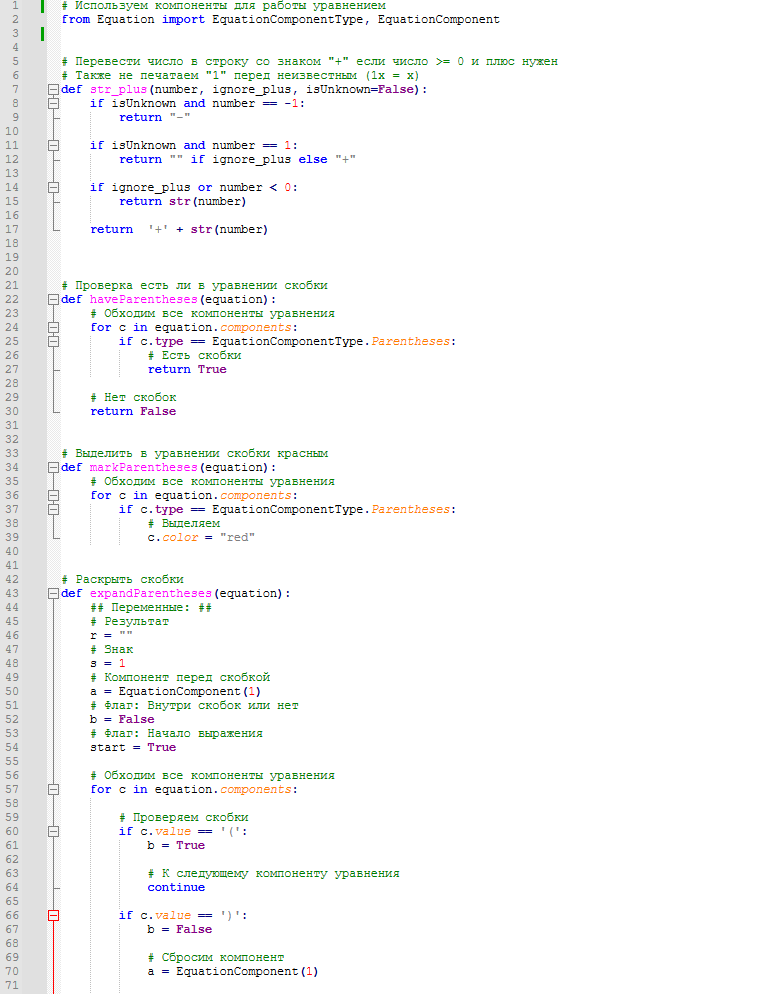
Уравнение.Неизвестное + «=» + КакСтрока(ЗначениеПравойЧасти) + «/» КакСтрока (ЗначениеЛевойЧасти)

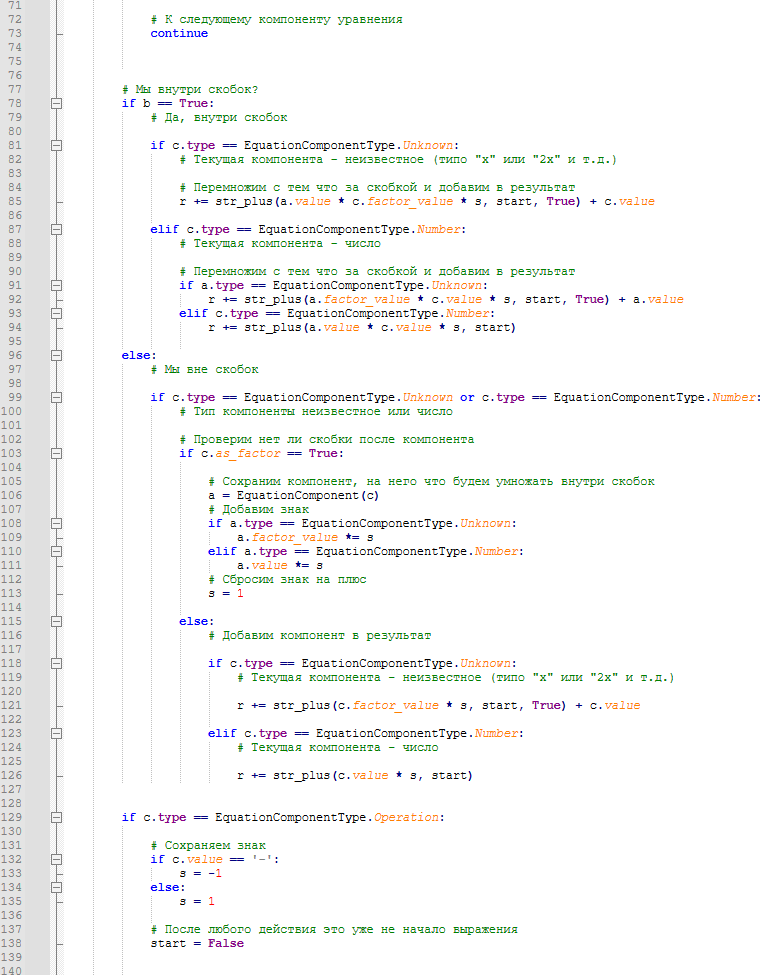
Уравнение.Неизвестное + «=» + КакСтрока(ЗначениеПравойЧасти / ЗначениеЛевойЧасти)

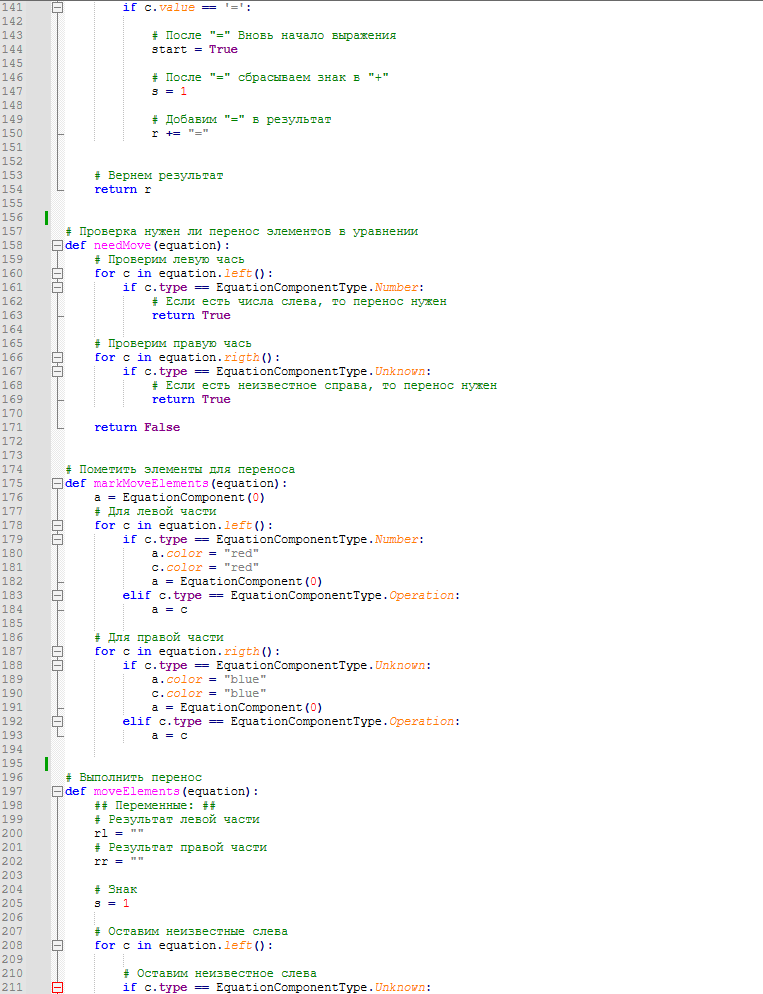
**Приложение 12.** Листинг модуля main.py

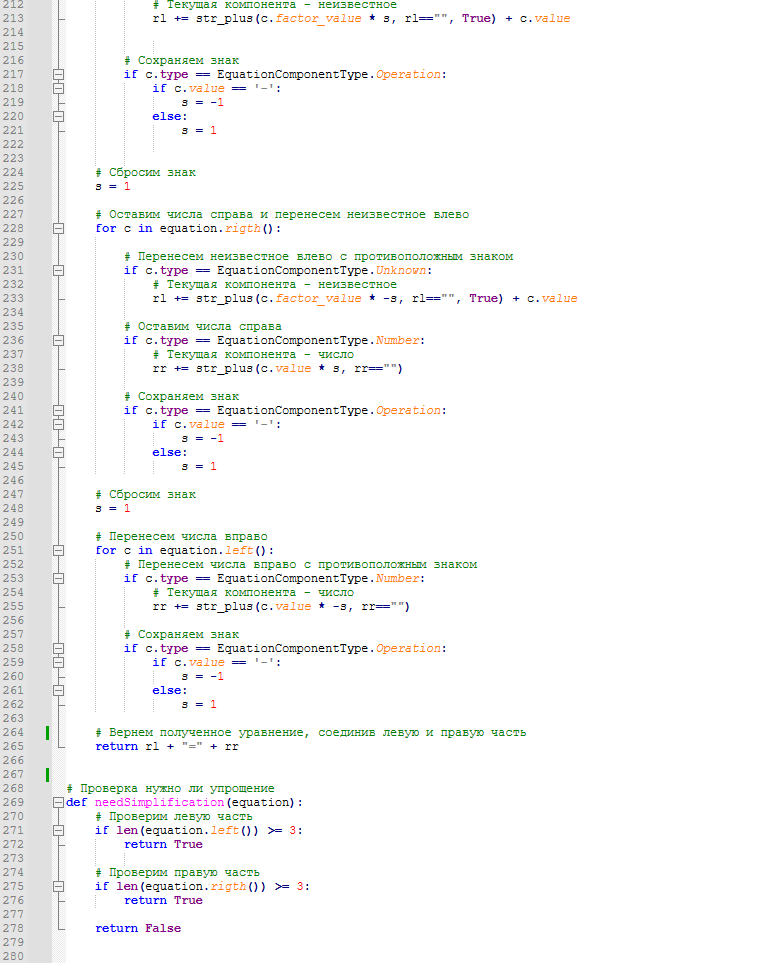
****

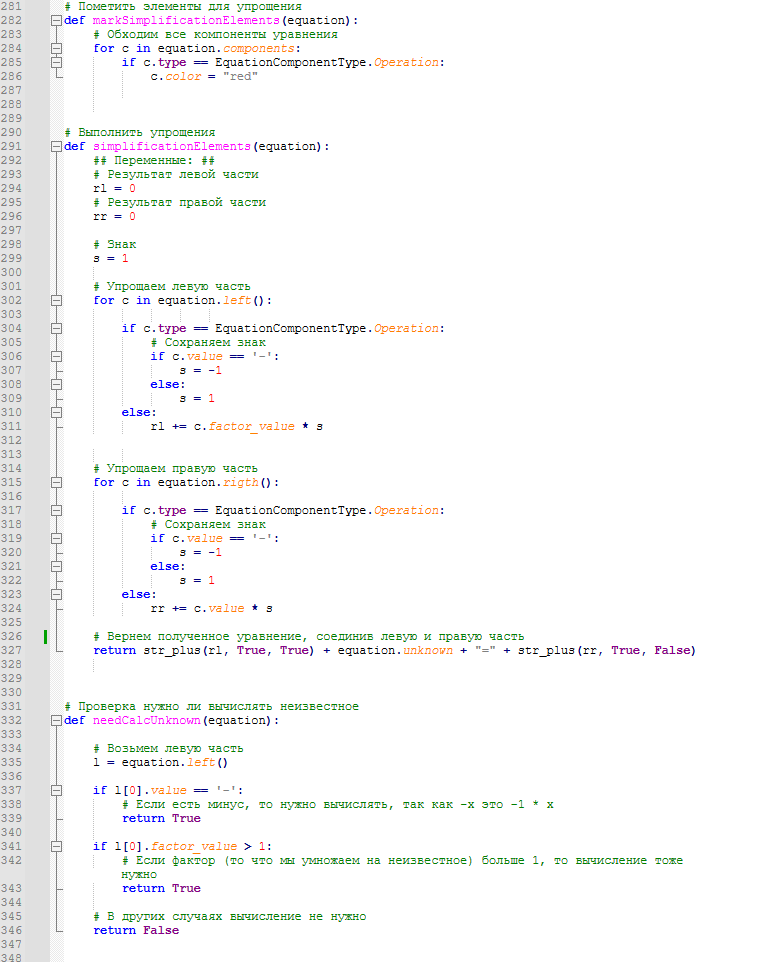
****

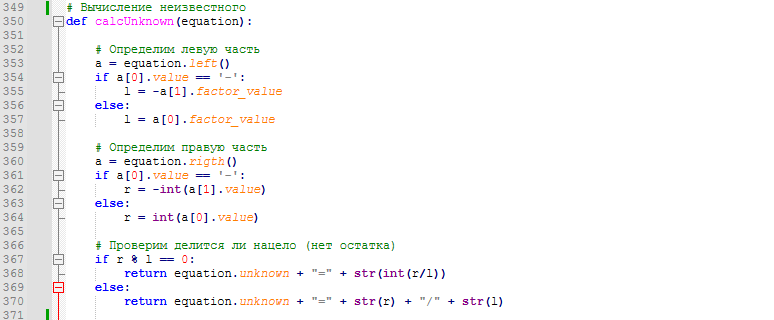
**Приложение 13.** Листинг модуля function.py

****

****

****

****

****

**Список использованных материалов**

1. Джейсон Бриггс. Python для детей, 2017
2. Brython: <https://brython.info>
3. Как решать линейные уравнения: <https://math-prosto.ru/ru/pages/linear_equations/solution_linear_equations>
4. Как решать линейные уравнения — формулы и примеры решения простейших уравнений: <https://skysmart.ru/articles/mathematic/reshenie-prostyh-linejnyh-uravnenij>