

Создание telegram-бота для решения квадратных уравнений на языке программирования Python

Выполнила: Андреева Дарья

Научный руководитель: Дедюхина Е.Н.



Актуальность

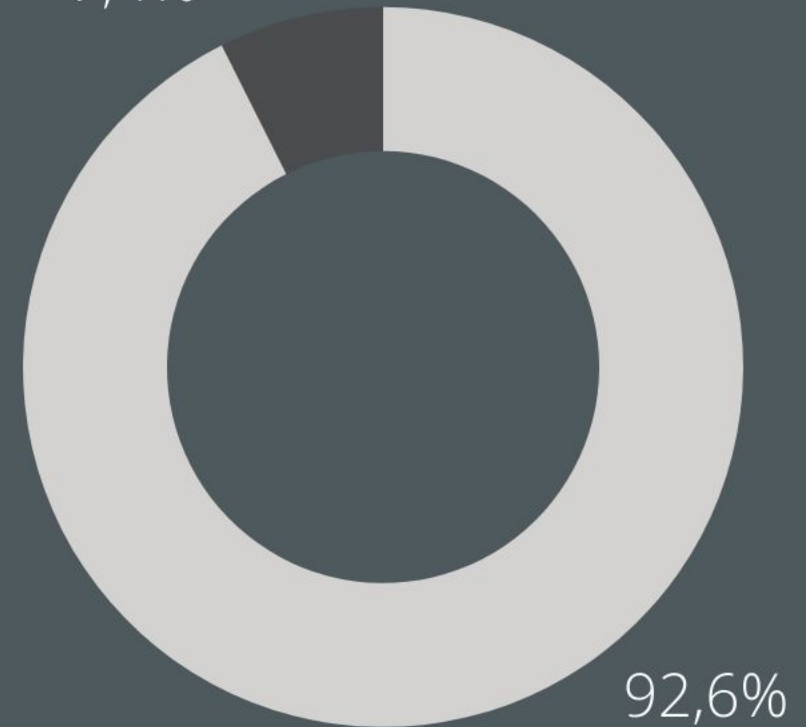
В настоящее время большинство подростков пользуются мессенджером Telegram

Создание ботов в Telegram – актуальная и развивающаяся сфера программирования

Язык Python быстр и удобен в работе, этот язык больше всего подходит для написания ботов

Квадратные уравнения применяются в решении многих задач по математике

Результаты опроса: 7,4%



Цель:

Освоение языка программирования Python, создание telegram-бота для самопроверки учащимися решения квадратных уравнений

Задачи:

- 1) Провести опрос с целью выявления проблемы
- 2) Изучить язык Python
- 3) Написать базовый алгоритм программы решения квадратных уравнений на языке Python
- 4) Изучить алгоритмы написания telegram-ботов, принципы работы API
- 5) Написать бота, провести его тестирование и опрос среди учеников

Ход работы

1.

Я изучила основы языка Python, для написания необходимых программ, после чего приступила к написанию программы по решению любого вида квадратного уравнения

Программа:

```
1 import math
2
3 print("Введите коэффициенты для уравнения")
4 print("ax^2 + bx + c = 0:")
5
6 a = float(input("a = "))
7 b = float(input("b = "))
8 c = float(input("c = "))
9
10 discr = b ** 2 - 4 * a * c
11 print("Дискриминант D = %.2f" % discr)
12
13 if discr > 0:
14     x1 = (-b + math.sqrt(discr)) / (2 * a)
15     x2 = (-b - math.sqrt(discr)) / (2 * a)
16     print("x1 = %.2f \nx2 = %.2f" % (x1, x2))
17
18 elif discr == 0:
19     x = -b / (2 * a)
20     print("x = %.2f" % x)
21
22 else:
23     print("Корней нет")
```

Ход работы

2.

Далее необходимо было изучить алгоритм написания бота и принципы работы API. Для создания токена любого бота используется главный бот Telegram:



Ход работы

3.

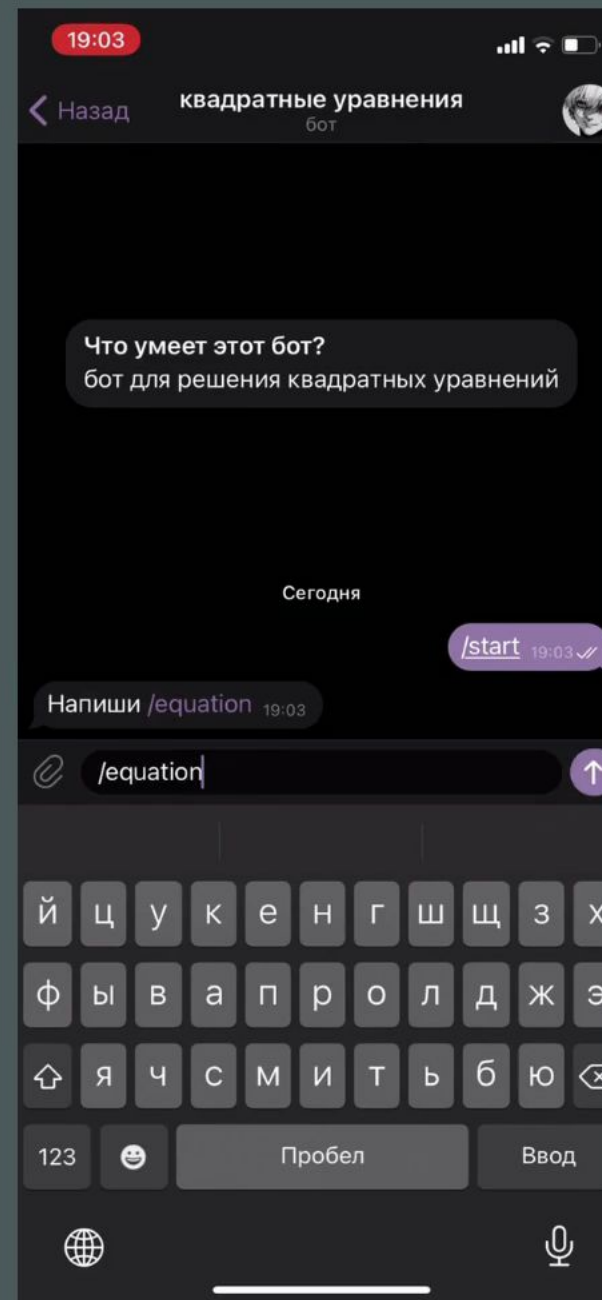
После создания токена я приступила к написанию бота.

Код с подробными пояснениями:

```
1 # Библиотеки
2 import telebot
3 import math
4
5 # Токен
6 bot = telebot.TeleBot('XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX')
7
8 # Начинаем читать что присылает пользователь
9 @bot.message_handler(content_types=['text'])
10 def start(message):
11     # Проверяем введена ли команда
12     if message.text == '/equation':
13         bot.send_message(message.from_user.id, "Введите коэффициенты для уравнения")
14         bot.send_message(message.from_user.id, "ax^2 + bx + c = 0:")
15         bot.send_message(message.from_user.id, "Введите a:")
16         bot.register_next_step_handler(message, get_a) # Считываем "a" и переходим к следующему шагу
17     # Просим ввести команду
18     else:
19         bot.send_message(message.from_user.id, 'Напиши /equation')
20 # Получаем и сохраняем "a"
21 def get_a(message):
22     global a
23     a = float(message.text)
24
25     # Просим ввести "b" и переходим к следующему шагу
26     bot.send_message(message.from_user.id, 'Введите b:')
27     bot.register_next_step_handler(message, get_b)
28
29 # Получаем и сохраняем "b"
30 def get_b(message):
31     global b
32     b = float(message.text)
```

```
34     # Просим ввести "c" и переходим к следующему шагу
35     bot.send_message(message.from_user.id, 'Введите c:')
36     bot.register_next_step_handler(message, get_c)
37
38 # Получаем и сохраняем "c"
39 def get_c(message):
40     global c
41     c = float(message.text)
42
43     # Начинаем вычисления и сразу выводим дискриминант
44     discr = b ** 2 - 4 * a * c
45     bot.send_message(message.from_user.id, "Дискриминант D = %.2f" % discr)
46
47     # Сверяем дискриминант с условиями
48     if discr > 0:
49         x1 = (-b + math.sqrt(discr)) / (2 * a)
50         x2 = (-b - math.sqrt(discr)) / (2 * a)
51         bot.send_message(message.from_user.id, "x1 = %.2f \nx2 = %.2f" % (x1, x2))
52     elif discr == 0:
53         x = -b / (2 * a)
54         bot.send_message(message.from_user.id, "x = %.2f" % x)
55     else:
56         bot.send_message(message.from_user.id, "Корней нет")
57
58 # Связывается с сервером телеграма для запроса сообщений 24/7
59 bot.polling(none_stop=False, interval=0 )
```

Демонстрация продукта



Заключение



Проект достиг своей цели,
получилось написать
функционирующего telegram-бота