

Боты для телеграмм и их польза в повседневной жизни

Выполнил: Сергеев Ермак,

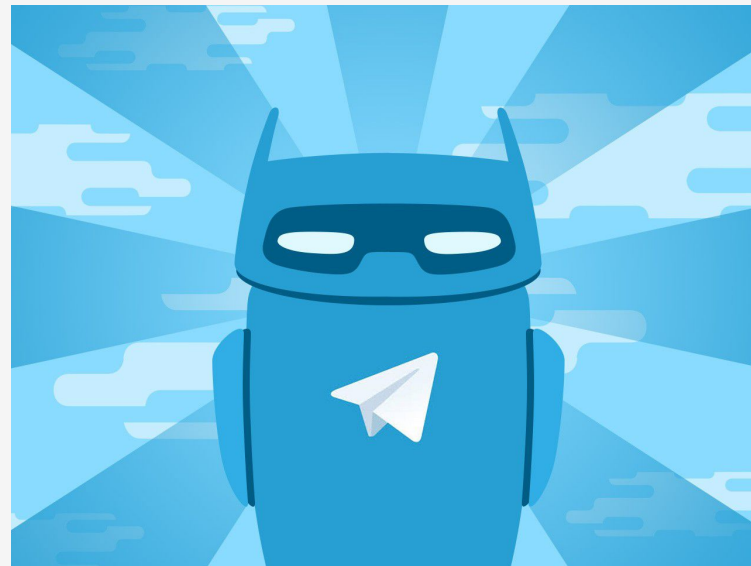
МБОУ «СОШ №17»

Руководитель:

Иванова И.А.

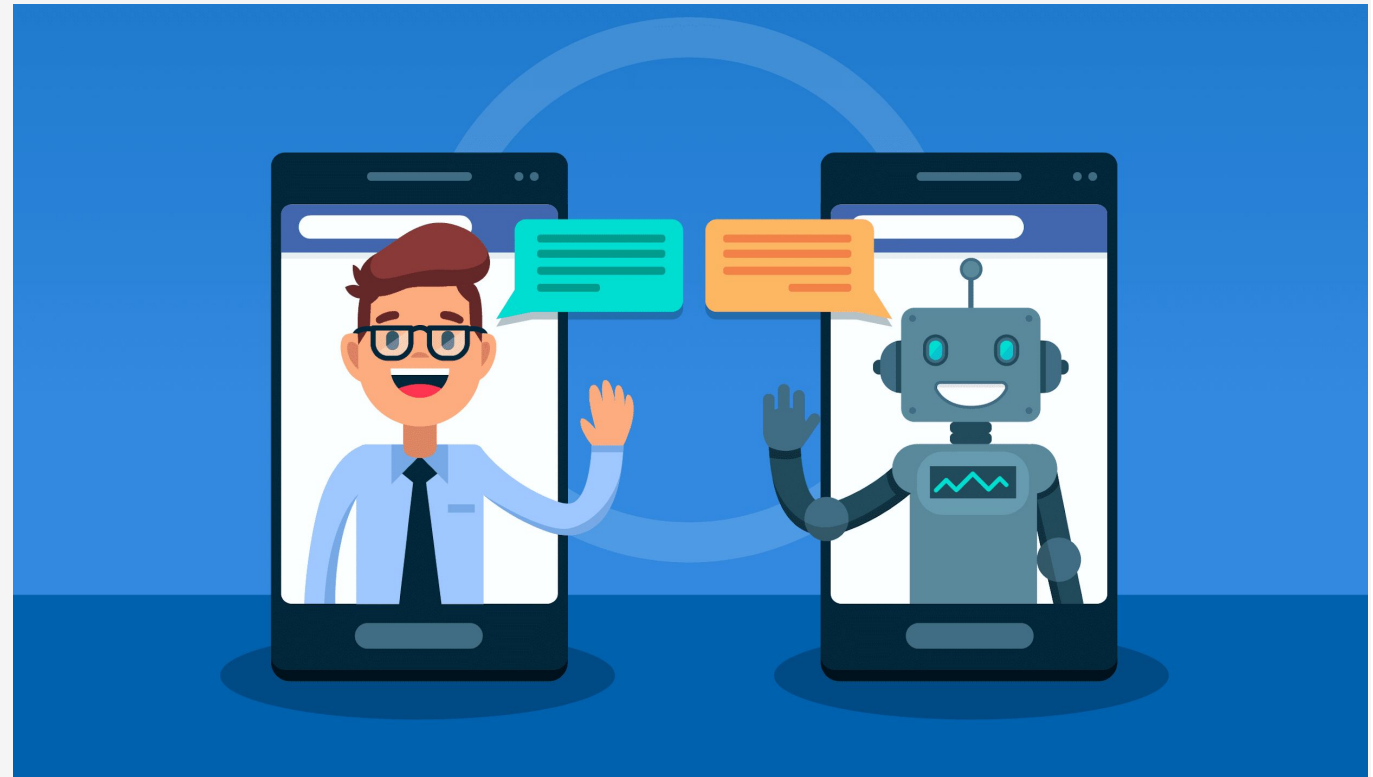
Ведение

Что такое Telegram и компьютерные боты?



Актуальность

- **Автоматизация рутинных задач**
- **Быстрота выполнения**
- **Удобство в использовании**



Цели и задачи

Цель: Написать бота для социальной сети - Telegram

Теоретические задачи:

1. **Определить наилучший язык программирования для этой задачи.**
2. **Установить нужное программное обеспечение на компьютер.**
3. **Найти и изучить нужную литературу.**

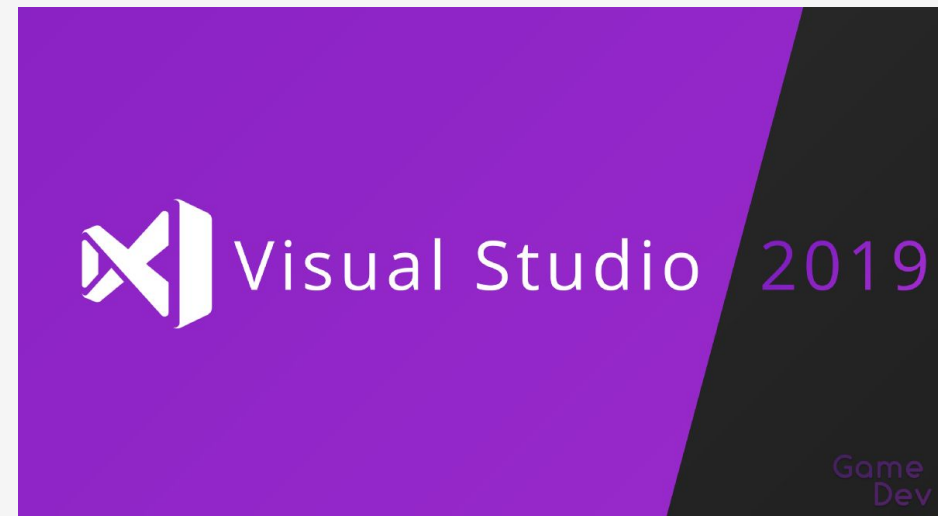
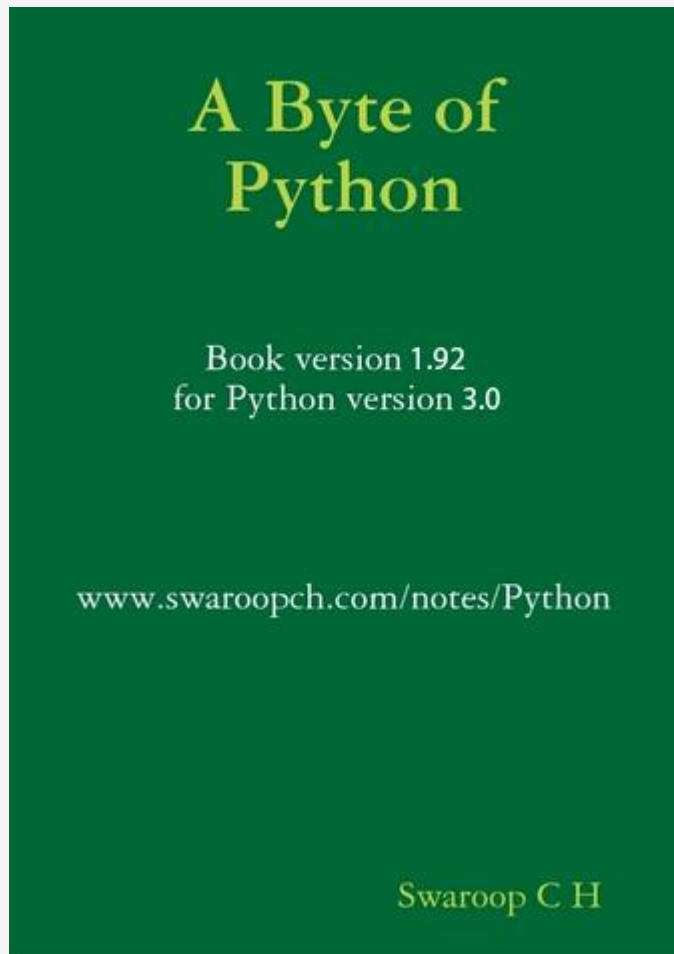
Практические задачи:

1. **Написать первую версию бота, протестировать и выявить все ошибки в его коде.**
2. **Исправить все недочеты, выявленные во время теста первой версии бота.**
3. **Сделать вывод.**

Определение языка программирования

Java Script	Python
Простой синтаксис	Простой синтаксис
Большое количество некачественных библиотек	Позволяет получить все функции с «коробки»
Проблемы работы с типами данных	Прост в изучении
Хорошая распространенность и большое количество литературы	Низкое быстродействие
Высокое быстродействие	Быстрота разработки

Установка программного обеспечения и изучение литературы



Написание первой версии бота и выявление всех ошибок в его выполнении

```
test.py - Visual Studio Code
Файл  Правка  Выделение  Вид  Переход  Выполнить  Терминал  Справка

test.py  X
C: > Python > test.py > ...
1  import telebot
2  from telebot import types
3  import COVID19Py
4
5  covid19 = COVID19Py.COVID19()
6  bot = telebot.TeleBot('1608210334:AAH41YSQEB20SQP44b09U7IW4CZ04vgur8Y')
7  |
```

```
8  # Функция при старте
9  # создание кнопок быстрого действия
10 @bot.message_handler(commands=['start'])
11 def start(message):
12     markup = types.ReplyKeyboardMarkup(resize_keyboard=True, row_width=2)
13     btn1 = types.KeyboardButton('Во всём мире')
14     btn2 = types.KeyboardButton('Украина')
15     btn3 = types.KeyboardButton('Россия')
16     btn4 = types.KeyboardButton('Беларусь')
17     markup.add(btn1, btn2, btn3, btn4)
18
19     send_message = f"<b>Привет {message.from_user.first_name}!</b>\nЧтобы узнать данные про коронавируса напишите " \
20     f"название страны, например: США, Украина, Россия и так далее\n\n"
21     bot.send_message(message.chat.id, send_message, parse_mode='html', reply_markup=markup)
```

Написание первой версии бота и выявление всех ошибок в его выполнении

```
23 # при отправке боту сообщения
24 # stata про некоторые страны
25 @bot.message_handler(content_types=['text'])
26 def mess(message):
27     final_message = ""
28     get_message_bot = message.text.strip().lower()
29     if get_message_bot == "сша":
30         location = covid19.getLocationByCountryCode("US")
31     elif get_message_bot == "украина":
32         location = covid19.getLocationByCountryCode("UA")
33     elif get_message_bot == "россия":
34         location = covid19.getLocationByCountryCode("RU")
35     elif get_message_bot == "беларусь":
36         location = covid19.getLocationByCountryCode("BY")
37     elif get_message_bot == "казакхстан":
38         location = covid19.getLocationByCountryCode("KZ")
39     elif get_message_bot == "италия":
40         location = covid19.getLocationByCountryCode("IT")
41     elif get_message_bot == "франция":
42         location = covid19.getLocationByCountryCode("FR")
43     elif get_message_bot == "германия":
44         location = covid19.getLocationByCountryCode("DE")
45     elif get_message_bot == "япония":
46         location = covid19.getLocationByCountryCode("JP")
47     else:
48         location = covid19.getlatest()
49     final_message = f"<u>Данные по всему миру:</u>\n<b>Заболевших: </b>{location['confirmed'];,}\n<b>Сметрей: </b>{location['deaths'];,}"
```

```
51     if final_message == "":
52         date = location[0]['last_updated'].split("T")
53         time = date[1].split(".")
54         final_message = f"<u>Данные по стране:</u>\nНаселение: {location[0]['country_population'];,}\n" \
55             f"Последнее обновление: {date[0]} {time[0]}\nПоследние данные:\n<b>" \
56             f"Заболевших: </b>{location[0]['latest']['confirmed'];,}\n<b>Сметрей: </b>" \
57             f"{location[0]['latest']['deaths'];,}"
58
59     bot.send_message(message.chat.id, final_message, parse_mode='html')
60
61 # чтобы нон стопился
62 bot.polling(none_stop=True)
```


Вывод и его работа:

