

**Московский государственный технический
университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №6
“ Разработка бота для Telegram с использованием языка Python”

Выполнил:
студент группы ИУ5-34Б:
Стукалов Иван Дмитриевич
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.
Подпись и дата:

2022

Разработка простого бота для Telegram с использованием языка Python.

Описание задания:

Разработайте простого бота для Telegram. Бот должен использовать функциональность создания кнопок.

Текст программы:

```
// файл main.py
import telebot
from telebot import types

bot = telebot.TeleBot(TOKEN) // токен скрыт

# Создание бота
bot = telebot.TeleBot(TOKEN)

@bot.message_handler(func=lambda message: True)
def echo_all(message):
    # Идентификатор диалога
    chat_id = message.chat.id

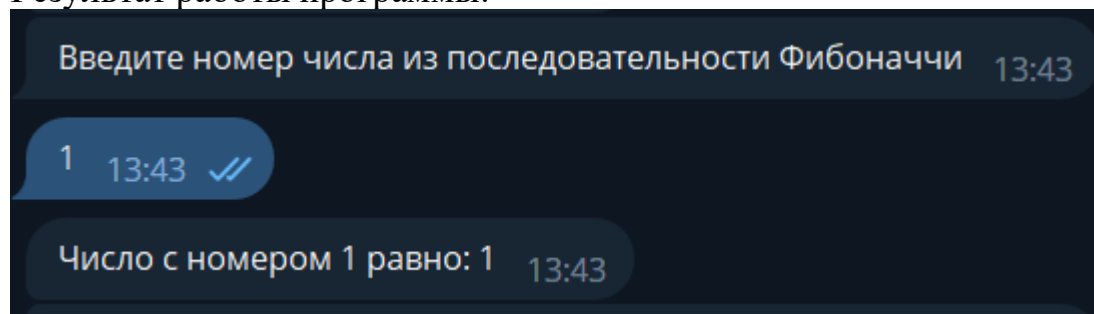
    # Текст, введенный пользователем, то есть текст с кнопки
    text = message.text

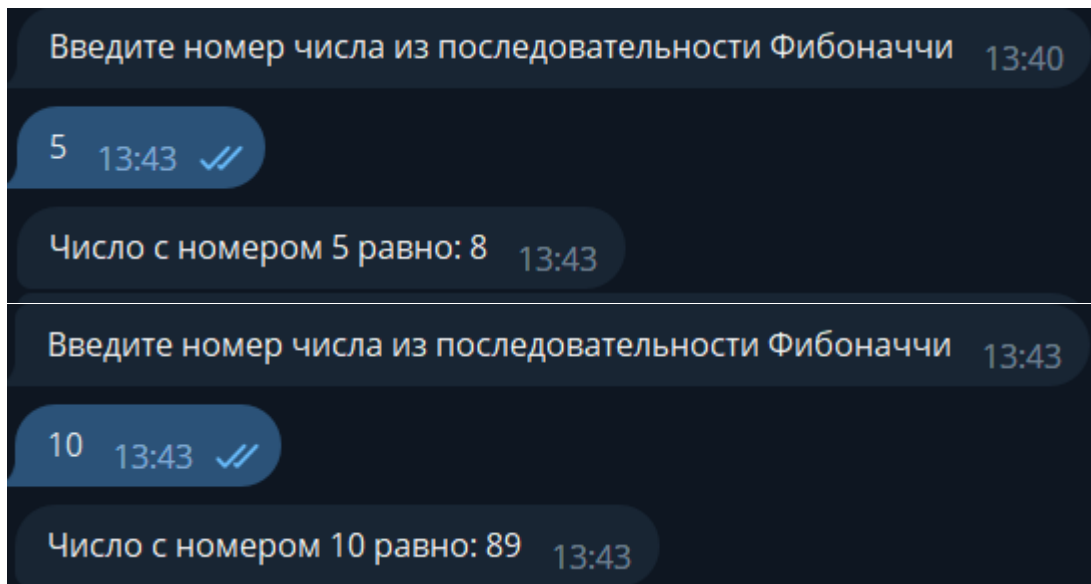
    if not text.isdigit(): # если введено не число
        bot.send_message(chat_id, 'Введите номер числа из последовательности Фибоначчи')

    if text.isdigit(): # если введено число
        a = 0
        b = 1
        for i in range(int(text)):
            c = a + b
            a = b
            b = c
        bot.send_message(message.chat.id, f'Число с номером {text} равно: {b}')
        bot.send_message(chat_id, 'Введите номер числа из последовательности Фибоначчи')

bot.infinity_polling()
```

Результат работы программы:





Разработка бота на основе конечного автомата для Telegram с использованием языка Python

Описание задания:

Разработайте бота для Telegram. Бот должен реализовывать конечный автомат из трех состояний.

Текст программы:

// файл config.py

```
from enum import Enum

# Токент бота
//токен скрыт

# Файл базы данных Vedis
db_file = "db.vdb"

# Ключ записи в БД для текущего состояния
CURRENT_STATE = "CURRENT_STATE"

# Состояния автомата
class States(Enum):
    STATE_START = "STATE_START" # Начало нового диалога
    STATE_NUM = "STATE_NUM"
    STATE_OPERATION = "STATE_OPERATION"
```

//файл dbworker.py

```
from vedis import Vedis
import config

# Чтение значения
def get(key):
    with Vedis(config.db_file) as db:
        try:
            return db[key].decode()
        except KeyError:
            # в случае ошибки значение по умолчанию - начало диалога
```

```

        return config.States.STATE_START.value

# Запись значения
def set(key, value):
    with Vedis(config.db_file) as db:
        try:
            db[key] = value
            return True
        except:
            # тут желательно как-то обработать ситуацию
            return False

# Создание ключа для записи и чтения
def make_key(chatid, keyid):
    res = str(chatid) + '__' + str(keyid)
    return res

```

// файл main.py

```

import telebot
from telebot import types
import config
import dbworker

bot = telebot.TeleBot(config.TOKEN)
bot_list = []

# Начало диалога
@bot.message_handler(commands=['start'])
def cmd_start(message):
    bot.send_message(message.chat.id, 'Я умею выполнять действия над массивом!')
    bot.send_message(message.chat.id, 'Когда закончите заполнять массив, введите "end"')
    dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE), config.States.STATE_NUM.value)
    bot.send_message(message.chat.id, 'Введите число')

# По команде /reset будем сбрасывать состояния, возвращаясь к началу диалога
@bot.message_handler(commands=['reset'])
def cmd_reset(message):
    bot.send_message(message.chat.id, 'Сбрасываем результаты предыдущего ввода.')
    dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE), config.States.STATE_NUM.value)
    bot.send_message(message.chat.id, 'Введите число')

# Обработка числа
@bot.message_handler(func=lambda message: dbworker.get(dbworker.make_key(message.chat.id, config.CURRENT_STATE)) == config.States.STATE_NUM.value)
def num_to_list(message):
    text = message.text
    if not text.isdigit(): # если введено не число
        if text == 'end':
            bot.send_message(message.chat.id, 'Конец ввода')
            dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE), config.States.STATE_OPERATION.value)
            bot.send_message(message.chat.id, 'Введенный массив:')
            for i in bot_list:

```

```

        bot.send_message(message.chat.id, i)

        markup = types.ReplyKeyboardMarkup()
        item_btn_1 = types.KeyboardButton('max')
        item_btn_2 = types.KeyboardButton('min')
        item_btn_3 = types.KeyboardButton('sum')
        item_btn_4 = types.KeyboardButton('count')
        item_btn_5 = types.KeyboardButton('avg')
        markup.add(item_btn_1, item_btn_2, item_btn_3, item_btn_4,
item_btn_5)
        bot.send_message(message.chat.id, 'Выберите пожалуйста действие',
reply_markup=markup)
        return
    else:
        # Состояние не изменяется, выводится сообщение об ошибке
        bot.send_message(message.chat.id, 'Пожалуйста введите число!')
        return
    else:
        float_text = float(text)
        bot.send_message(message.chat.id, f'Вы ввели число {text}')
        # Сохраняем число
        bot_list.append(float_text)

        dbworker.set(dbworker.make_key(message.chat.id,
config.States.STATE_NUM.value), bot_list)
        bot.send_message(message.chat.id, 'Введите следующее число')

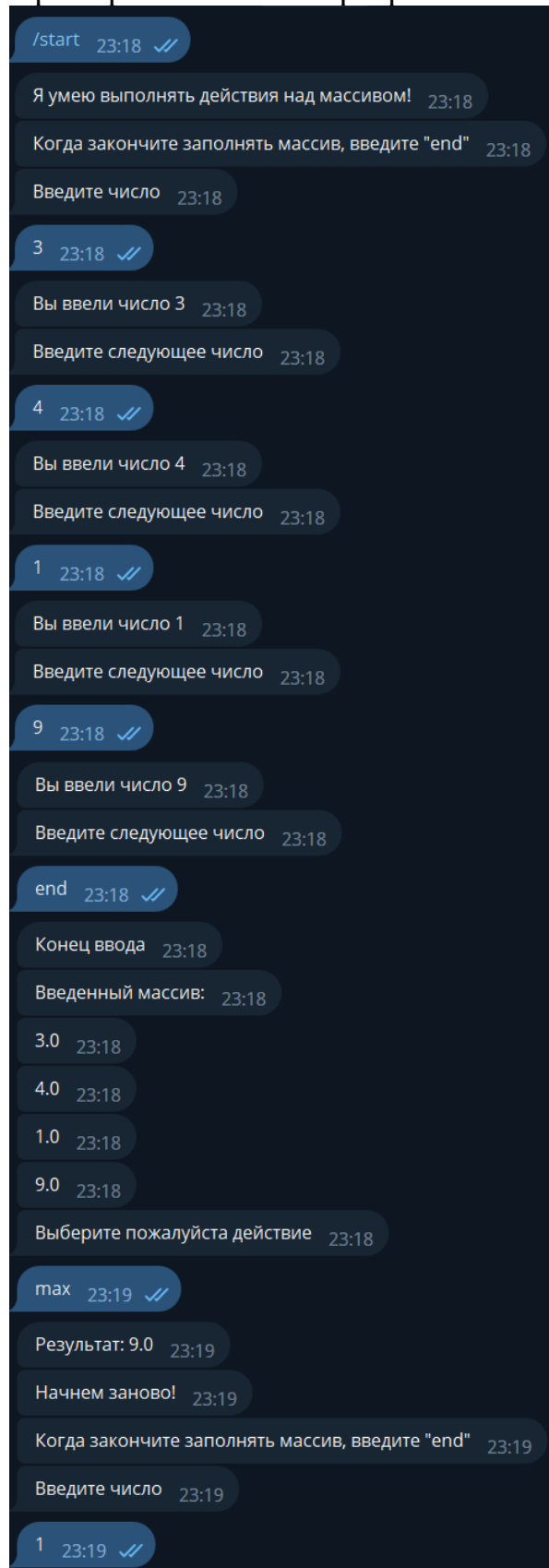
# Выбор действия
@bot.message_handler(func=lambda message: dbworker.get(
    dbworker.make_key(message.chat.id, config.CURRENT_STATE)) ==
config.States.STATE_OPERATION.value)
def operation(message):
    # Текущее действие
    op = message.text
    # Выполняем действие
    res = 0
    if op == 'max':
        res = max(bot_list)
    elif op == 'min':
        res = min(bot_list)
    elif op == 'sum':
        res = sum(bot_list)
    elif op == 'count':
        res = len(bot_list)
    elif op == 'avg':
        res = sum(bot_list) / len(bot_list)

    # Выводим результат
    markup = types.ReplyKeyboardRemove(selective=False)
    bot.send_message(message.chat.id, f'Результат: {str(res)}',
reply_markup=markup)
    # Меняем текущее состояние
    dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.STATE_NUM.value)
    bot_list.clear()
    # Выводим сообщение
    bot.send_message(message.chat.id, 'Начнем заново!')
    bot.send_message(message.chat.id, 'Когда закончите заполнять массив,
введите "end"')
    bot.send_message(message.chat.id, 'Введите число')

```

```
if __name__ == '__main__':  
    bot.infinity_polling()
```

Пример выполнения программы:



Вы ввели число 1 23:19

Введите следующее число 23:19

2 23:19 ✓✓

Вы ввели число 2 23:19

Введите следующее число 23:19

5 23:19 ✓✓

Вы ввели число 5 23:19

Введите следующее число 23:19

end 23:19 ✓✓

Конец ввода 23:19

Введенный массив: 23:19

1.0 23:19

2.0 23:19

5.0 23:19

Выберите пожалуйста действие 23:19

min 23:19 ✓✓

Результат: 1.0 23:19

Начнем заново! 23:19

Когда закончите заполнять массив, введите "end" 23:19

Введите число 23:19

5 23:19 ✓✓

Вы ввели число 5 23:19

Введите следующее число 23:19

6 23:19 ✓✓

Вы ввели число 6 23:19

Введите следующее число 23:19

7 23:19 ✓✓

Вы ввели число 7 23:19

Введите следующее число 23:19

end 23:19 ✓✓

Конец ввода 23:19

Введенный массив: 23:19

5.0 23:19

6.0 23:19

7.0 23:19

Выберите пожалуйста действие 23:19

sum 23:19 ✓✓

Результат: 18.0 23:19

Начнем заново! 23:19

Когда закончите заполнять массив, введите "end" 23:19

Введите число 23:19

10 23:19 ✓✓

Вы ввели число 10 23:19

Введите следующее число 23:19

19 23:19 ✓✓

Вы ввели число 19 23:19

Введите следующее число 23:19

4 23:19 ✓✓

Вы ввели число 4 23:19

Введите следующее число 23:19

end 23:19 ✓✓

Конец ввода 23:19

Введенный массив: 23:19

10.0 23:19

19.0 23:19

4.0 23:19

Выберите пожалуйста действие 23:19

count 23:19 ✓✓

Результат: 3 23:19

Начнем заново! 23:19

Когда закончите заполнять массив, введите "end" 23:19

Введите число 23:19

33 23:19 ✓✓

Вы ввели число 33 23:19

Введите следующее число 23:19

15 23:19 ✓✓

Вы ввели число 15 23:19

Введите следующее число 23:19

87 23:19 ✓✓

Вы ввели число 87 23:19

Введите следующее число 23:19

end 23:19 ✓✓

Конец ввода 23:19

Введенный массив: 23:19

33.0 23:19

15.0 23:19

87.0 23:19

Выберите пожалуйста действие 23:19

avg 23:20 ✓✓

Результат: 45.0 23:20