**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: |  | Проверил: |
| студент группы ИУ5-34Б: |  | преподаватель каф. ИУ5 |
| Стукалов Иван Дмитриевич |  | Гапанюк Ю.Е. |
| Подпись и дата: |  | Подпись и дата: |

2022

Разработка простого бота для Telegram с использованием языка Python.

Описание задания:

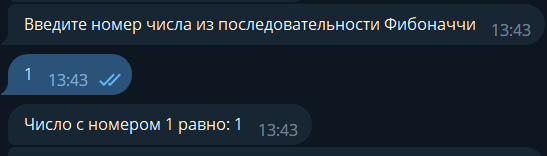
Разработайте простого бота для Telegram. Бот должен использовать функциональность создания кнопок.

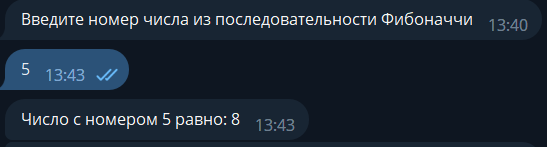
Текст программы:

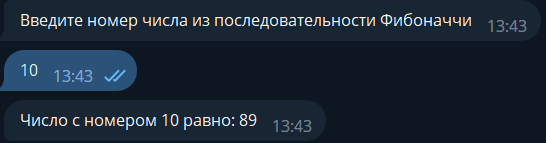
// файл main.py

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:

