

Факультет РТ Радиотехнический

Кафедра ИУ5 Системы обработки информации и управления

**Отчет по лабораторной работе № 6 по курсу
Базовые компоненты интернет-технологий**

Исполнитель

Студент группы РТ5-31Б _____ Яковенко Ю.С.

“ ____ ” _____ 2021 г.

Проверил

Доцент кафедры ИУ5 _____ Гапанюк Ю.Е.

“ ____ ” _____ 2021 г.

Содержание

| | |
|--|---|
| 1. Описание задания..... | 3 |
| 2. Текст программы..... | 3 |
| 3. Экранные формы с примерами выполнения программы | 5 |

1. Описание задания

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

2. Текст программы

bot.py

```
import telebot
from telebot import types
import config
import dbworker

# Создание бота
bot = telebot.TeleBot(config.TOKEN)

# Начало диалога
@bot.message_handler(commands=['start'])
def cmd_start(message):
    bot.send_message(message.chat.id, 'Выполнение действий над двумя числами')
    dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.STATE_FIRST_NUM.value)
    bot.send_message(message.chat.id, 'Введите первое 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_FIRST_NUM.value)
    bot.send_message(message.chat.id, 'Введите первое число')

# Обработка первого числа
@bot.message_handler(func=lambda message:
dbworker.get(dbworker.make_key(message.chat.id, config.CURRENT_STATE)) ==
config.States.STATE_FIRST_NUM.value)
def first_num(message):
    text = message.text
    if not text.isdigit():
        # Состояние не изменяется, выводится сообщение об ошибке
```

```

    bot.send_message(message.chat.id, 'Пожалуйста введите число!')
    return
else:
    bot.send_message(message.chat.id, f'Вы ввели первое число {text}')
    # Меняем текущее состояние
    dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.STATE_SECOND_NUM.value)
    # Сохраняем первое число
    dbworker.set(dbworker.make_key(message.chat.id,
config.States.STATE_FIRST_NUM.value), text)
    bot.send_message(message.chat.id, 'Введите второе число')

# Обработка второго числа
@bot.message_handler(func=lambda message:
dbworker.get(dbworker.make_key(message.chat.id, config.CURRENT_STATE)) ==
config.States.STATE_SECOND_NUM.value)
def second_num(message):
    text = message.text
    if not text.isdigit():
        # Состояние не изменяется, выводится сообщение об ошибке
        bot.send_message(message.chat.id, 'Пожалуйста введите число!')
        return
    else:
        bot.send_message(message.chat.id, f'Вы ввели второе число {text}')
        # Меняем текущее состояние
        dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.STATE_OPERATION.value)
        # Сохраняем первое число
        dbworker.set(dbworker.make_key(message.chat.id,
config.States.STATE_SECOND_NUM.value), text)
        markup = types.ReplyKeyboardMarkup(row_width=4)
        itembtn1 = types.KeyboardButton('+')
        itembtn2 = types.KeyboardButton('*')
        itembtn3 = types.KeyboardButton('-')
        itembtn4 = types.KeyboardButton('/')
        markup.add(itembtn1, itembtn2, itembtn3, itembtn4)
        bot.send_message(message.chat.id, 'Выберите пожалуйста действие',
reply_markup=markup)

# 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
    # Читаем операнды из базы данных
    v1 = dbworker.get(dbworker.make_key(message.chat.id,
config.States.STATE_FIRST_NUM.value))
    v2 = dbworker.get(dbworker.make_key(message.chat.id,
config.States.STATE_SECOND_NUM.value))
    # Выполняем действие
    fv1 = float(v1)
    fv2 = float(v2)
    res = 0
    if op == '+':
        res = fv1 + fv2
    elif op == '*':
        res = fv1 * fv2
    elif op == '-':
        res = fv1 - fv2
    elif op == '/':
        res = fv1 / fv2
    # Выводим результат
    markup = types.ReplyKeyboardRemove(selective=False)
    bot.send_message(message.chat.id, f'Результат: {v1}{op}{v2}={str(res)}',
reply_markup=markup)
    # Меняем текущее состояние
    dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.STATE_FIRST_NUM.value)
    # Выводим сообщение
    bot.send_message(message.chat.id, 'Введите первое число')

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

```

config.py

```

from enum import Enum

```

```

# Токент бота

```

```

TOKEN = '5086111476:AAE1CPf6uNZi45copOdDghd5miLE-kJAMYU'

```

```

# 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_FIRST_NUM = "STATE_FIRST_NUM"
    STATE_SECOND_NUM = "STATE_SECOND_NUM"
    STATE_OPERATION = "STATE_OPERATION"

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.S_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

```

3. Экранные формы с примерами выполнения программы

