

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

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

Курс «Казовые компоненты интернет технологий»

Отчет по лабораторной работе №6  
«Разработка простого бота в Telegram с использованием языка Python»

Выполнил:  
студент группы ИУ5-33Б:

Лебедева С.К.

Руководитель:  
преподаватель каф.  
ИУ5

Гапанюк Ю.Е.

Москва, 2022 г.

## Цель лабораторной работы:

Изучение разработки ботов в Telegram.

## Задание:

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

## Текст программы:

### bot.py

```
import telebot
import config
import dbworker
import random

bot = telebot.TeleBot(config.TOKEN)

@bot.message_handler(commands=['start'])
def start_message(message):
    bot.send_message(message.chat.id, 'Я умею бросать кубики различной
гранности!')

    dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.STATE_FIRST_NUM.value)
    bot.send_message(message.chat.id, 'Введите количество кубиков (одинаковой
гранности)')

@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 количество кубиков (одинаковой
гранности)!')

# 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
```

```

elif int(text) <= 0:
    # Состояние не изменяется, выводится сообщение об ошибке
    bot.send_message(message.chat.id, 'Пожалуйста введите число, большее
нуля!')
    return
else:
    bot.send_message(message.chat.id, f'Вы ввели количество кубиков {text}')
    # Меняем текущее состояние
    dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.STATE_SECOND_NUM.value)
    # Сохраняем число кубиков
    dbworker.set(dbworker.make_key(message.chat.id,
config.States.STATE_FIRST_NUM.value), text)
    bot.send_message(message.chat.id, 'Введите значение модификатора')

@bot.message_handler(commands=['test'])
def start_message(message):
    markup = telebot.types.InlineKeyboardMarkup()
    markup.add(telebot.types.InlineKeyboardButton(text='d4', callback_data=4))
    markup.add(telebot.types.InlineKeyboardButton(text='d6', callback_data=6))
    markup.add(telebot.types.InlineKeyboardButton(text='d8', callback_data=8))
    markup.add(telebot.types.InlineKeyboardButton(text='d10', callback_data=10))
    markup.add(telebot.types.InlineKeyboardButton(text='d10 по десяткам',
callback_data=1))
    markup.add(telebot.types.InlineKeyboardButton(text='d12', callback_data=12))
    markup.add(telebot.types.InlineKeyboardButton(text='d20', callback_data=20))
    bot.send_message(message.chat.id, text="Какой кубик хотите бросить?",
reply_markup=markup)
# 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 значение модификатора
{text}')
        # Меняем текущее состояние
        dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.STATE_COUNTING.value)
        # Сохраняем первое число
        dbworker.set(dbworker.make_key(message.chat.id,
config.States.STATE_SECOND_NUM.value), text)
        markup = telebot.types.ReplyKeyboardMarkup(row_width=2)
        itembtn1 = telebot.types.KeyboardButton('d4')
        itembtn2 = telebot.types.KeyboardButton('d6')
        itembtn3 = telebot.types.KeyboardButton('d8')

```

```

        itembtn4 = telebot.types.KeyboardButton('d10')
        itembtn5 = telebot.types.KeyboardButton('d10 по десяткам')
        itembtn6 = telebot.types.KeyboardButton('d12')
        itembtn7 = telebot.types.KeyboardButton('d20')
        markup.add(itembtn1, itembtn2, itembtn3, itembtn4, itembtn5, itembtn6,
itembtn7)

        bot.send_message(message.chat.id, 'Выберите, пожалуйста, кубик',
reply_markup=markup)

@bot.message_handler(func=lambda message:
dbworker.get(dbworker.make_key(message.chat.id, config.CURRENT_STATE)) ==
config.States.STATE_COUNTING.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 = int(v1)
    fv2 = int(v2)
    res = 0
    r = 0
    answer = ''
    s = op[1:]
    #size = 0
    if len(s) > 2:
        for i in range(0, fv1):
            r = 10 * random.randint(1, 10)
            res = res + r
            if i==0:
                answer = answer + str(r)
            else:
                answer = answer + ' ' + str(r)
        res = res + fv2
    else:
        for i in range(0, fv1):
            r = random.randint(1, int(s))
            res = res + r
            if i==0:
                answer = answer + str(r)
            else:
                answer = answer + ' ' + str(r)
        res = res + fv2
    # Выводим результат
    markup = telebot.types.ReplyKeyboardRemove(selective=False)
    bot.send_message(message.chat.id, f'Броски: {answer} \n Результат:
{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 количество кубиков')

    ...

@bot.message_handler(content_types=['text'])
def send_text(message):
    if message.text.lower() == 'привет':
        bot.send_message(message.chat.id, 'Ещё раз привет!')
    elif message.text.lower() == 'пока':
        bot.send_message(message.chat.id, 'Пока!')
    ...

@bot.callback_query_handler(func=lambda call: True)
def get_number(call):

    bot.answer_callback_query(callback_query_id=call.id, text='Кубик брошен!')
    answer = ''
    line = ''

    if call.data == '1':
        res = 10 * random.randint(1, 10)
        line = 'd10 по десяткам'
        answer = res
    else:
        res = random.randint(1, int(call.data))
        line = 'd'+str(call.data)
        answer = res

    bot.send_message(call.message.chat.id, 'Брошен кубик ' + line)
    bot.send_message(call.message.chat.id, answer)
    bot.edit_message_reply_markup(call.message.chat.id, call.message.message_id)

bot.polling()

```

## config.py

```

from enum import Enum

# Токен бота
TOKEN = '5889577871:AAEQXcaoSqW-g4dVb95mzD-eUno4fuAy8wI '

# 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_COUNTING = 'STATE_COUNTING'
```

## dbworker.py

```
from vedis import Vedis
import config

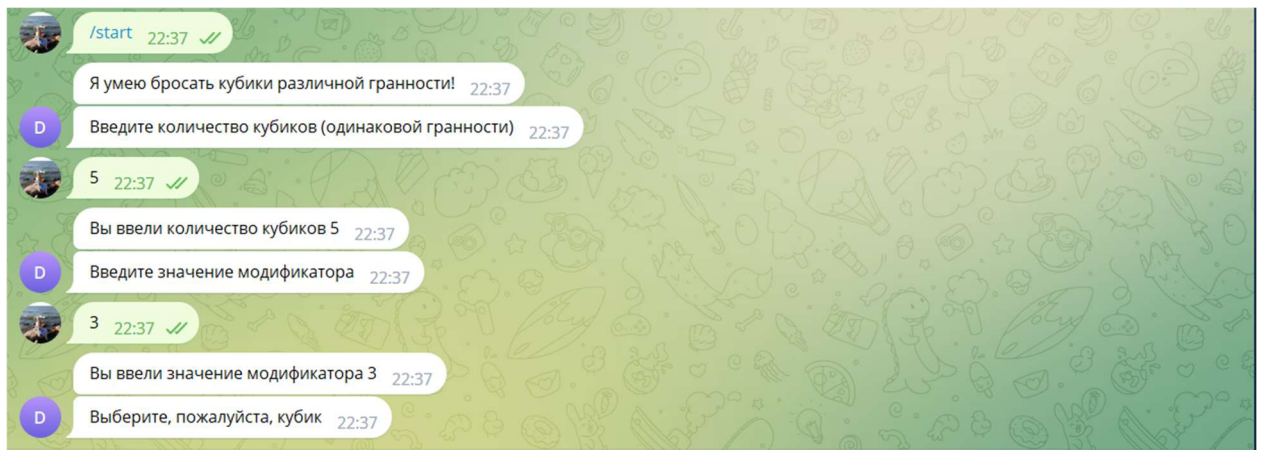
# Чтение значения
def get(key):
    with Vedis(config.db_file) as db:
        try:
            return db[key].decode()
        except KeyError:
            # в случае ошибки значение по умолчанию - начало диалога
            return config.States.S_START.value

# Запись значения
def set(key, value):
    with Vedis(config.db_file) as db:
        try:
            db[key] = value
            return True
        except:
            return False

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

## Примеры выполнения:

```
PS C:\Users\sophi\Documents\Python\leb_lab_6> python bot.py
█
```



Написать сообщение...

d4	d6
d8	d10
d10 по десяткам	d12
d20	

