**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Радиотехнический»

Кафедра «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №6

«Разработка простого бота для Telegram с использованием языка Python.»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы РТ5-31Б: |  | преподаватель кафедры ИУ5 |
| Чиканчи В. А. |  | Гапанюк Ю. Е. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2024 г.

Постановка задачи

Разработайте простого бота для Telegram.

Текст программы

**Файл «my\_bot.py»:**

from aiogram import Bot, Dispatcher, types

from aiogram.filters import Command

from aiogram.types import Message

import json

import os

BOT\_TOKEN = '7882305908:AAFuWhjoT9tOKTCNLLnE6AJJjR4Z4nxU3cE'

# Создаем объекты бота и диспетчера

bot = Bot(token=BOT\_TOKEN)

dp = Dispatcher()

battle\_servers = {}

with **open**(os.path.**join**("users\_settings.json"),"r", encoding="utf-8") as file:

    users\_settings = json.**load**(file)

def **users\_settings\_change**():

    with **open**(os.path.**join**("users\_settings.json"),"w") as file:

        json.**dump**(users\_settings,file)

def **create\_battle\_server**(player1, player2):

    for i in range(10):

        if not i in battle\_servers:

            battle\_servers[i] = {

                "player1": player1,

                "player2": player2,

                "number\_player1": -1,

                "number\_player2": -1,

                "ans\_player1": -1,

                "ans\_player2": -1,

            }

            users\_settings[player1]["battle\_server\_id"] = i

            users\_settings[player2]["battle\_server\_id"] = i

def **create\_buttons**():

    # markup = types.ReplyKeyboardMarkup(resize\_keyboard=True)

    # btn1 = types.KeyboardButton("Поиск противника")

    # btn2 = types.KeyboardButton("Статус")

    markup = [

        [types.KeyboardButton(text="Поиск противника"), types.KeyboardButton(text="Статус")]]

    markup = types.ReplyKeyboardMarkup(keyboard=markup,resize\_keyboard=True)

    return markup

async def **battling\_players**(player: str,text: str, id\_server: int):

    if text.**isdigit**():

        if player == battle\_servers[id\_server]["player1"]:

            number\_player = "number\_player1"

            ans\_player = "ans\_player1"

            enemy = battle\_servers[id\_server]["player2"]

            number\_enemy = "number\_player2"

            ans\_enemy = "ans\_player2"

        else:

            number\_player = "number\_player2"

            ans\_player = "ans\_player2"

            enemy = battle\_servers[id\_server]["player1"]

            number\_enemy = "number\_player1"

            ans\_enemy = "ans\_player1"

        ans = int(text)

        if 0<=ans<=10:

            if battle\_servers[id\_server][number\_player] == -1:

                if battle\_servers[id\_server][number\_enemy] == -1:

                    await bot.**send\_message**(player, "Ожидаем число вашего противника.")

                else:

                    await bot.**send\_message**(player, "Ваш противник уже загадал число. Игра началась.")

                    await bot.**send\_message**(enemy, "Игра началась.")

                battle\_servers[id\_server][number\_player] = ans

            elif battle\_servers[id\_server][number\_enemy] == -1:

                await bot.**send\_message**(player, "Противник ещё не загадал число.")

            elif battle\_servers[id\_server][ans\_player] == -1:

                if battle\_servers[id\_server][ans\_enemy] == -1:

                    await bot.**send\_message**(player, "Ожидаем ответа противника.")

                battle\_servers[id\_server][ans\_player] = ans

                if battle\_servers[id\_server][ans\_player] != -1 and battle\_servers[id\_server][ans\_enemy] != -1:

                    if battle\_servers[id\_server][ans\_player] == battle\_servers[id\_server][number\_enemy] and battle\_servers[id\_server][ans\_enemy] == battle\_servers[id\_server][number\_player]:

                        await bot.**send\_message**(player, "Вы оба победили.", reply\_markup=**create\_buttons**())

                        await bot.**send\_message**(enemy, "Вы оба победили.", reply\_markup=**create\_buttons**())

                        users\_settings[player]["wins"]+=1

                        users\_settings[enemy]["wins"]+=1

                        users\_settings[player]["status"]="registrated"

                        users\_settings[enemy]["stasus"]="registrated"

                        del battle\_servers[id\_server]

**users\_settings\_change**()

                    elif battle\_servers[id\_server][ans\_player] == battle\_servers[id\_server][number\_enemy]:

                        await bot.**send\_message**(player, "Вы победили.", reply\_markup=**create\_buttons**())

                        await bot.**send\_message**(enemy, "Вы проиграли.", reply\_markup=**create\_buttons**())

                        users\_settings[player]["wins"]+=1

                        users\_settings[enemy]["loses"]+=1

                        users\_settings[player]["status"]="registrated"

                        users\_settings[enemy]["stasus"]="registrated"

**users\_settings\_change**()

                        del battle\_servers[id\_server]

                    elif battle\_servers[id\_server][ans\_enemy] == battle\_servers[id\_server][number\_player]:

                        await bot.**send\_message**(player, "Вы проиграли.", reply\_markup=**create\_buttons**())

                        await bot.**send\_message**(enemy, "Вы победили.", reply\_markup=**create\_buttons**())

                        users\_settings[player]["loses"]+=1

                        users\_settings[enemy]["wins"]+=1

                        users\_settings[player]["status"]="registrated"

                        users\_settings[enemy]["stasus"]="registrated"

                        del battle\_servers[id\_server]

**users\_settings\_change**()

                    else:

                        await bot.**send\_message**(player, "Вы оба не угадали. Игра продолжается.")

                        await bot.**send\_message**(enemy, "Вы оба не угадали. Игра продолжается.")

                        battle\_servers[id\_server][ans\_enemy] = -1

                        battle\_servers[id\_server][ans\_player] = -1

        else:

            await bot.**send\_message**(player, "Число должно быть от 0 до 10.")

    else:

        await bot.**send\_message**(player, "Это не число.")

# Этот хэндлер будет срабатывать на команду "/start"

**@dp.message**(Command(commands=["start"]))

async def **process\_start\_command**(message: Message):

    if not str(message.from\_user.id) in users\_settings:

        users\_settings[str(message.from\_user.id)] = {

            "id": str(message.from\_user.id),

            "name": "",

            "status": "registrating",

            "battle\_server\_id": -1,

            "wins": 0,

            "loses": 0

        }

        await message.**answer**(f"Добрый день.\nВведите своё имя.")

    else:

        if users\_settings[str(message.from\_user.id)]["status"] == "battling":

            return

        name = users\_settings[str(message.from\_user.id)]["name"]

        await message.**answer**(f"Привет, {name}", reply\_markup=**create\_buttons**())

**@dp.message**(Command(commands=["battle"]))

async def **process\_battle\_command**(message: Message):

    if str(message.from\_user.id) in users\_settings:

        if users\_settings[str(message.from\_user.id)]["status"] == "battling":

            return

        await message.**answer**("Начинается поиск. Чтобы отменить, напишите /cansel.")

        for user in users\_settings:

            if users\_settings[user]["status"] == "searching":

                name1 = users\_settings[str(message.from\_user.id)]["name"]

                name2 = users\_settings[user]["name"]

                await message.**answer**(f"Ваш противник: {name2}. Загадайте число")

                await bot.**send\_message**(users\_settings[user]["id"], f"Ваш противник: {name1}. Загадайте число")

**create\_battle\_server**(str(user),str(message.from\_user.id))

                users\_settings[str(message.from\_user.id)]["status"] = "battling"

                users\_settings[user]["status"] = "battling"

                break

        else:

            users\_settings[str(message.from\_user.id)]["status"] = "searching"

    else:

        await message.**answer**("Сначала напишите /start, чтобы пройти регистрацию.")

**@dp.message**(Command(commands=["cansel"]))

async def **process\_cancel\_command**(message: Message):

    if str(message.from\_user.id) in users\_settings:

        if users\_settings[str(message.from\_user.id)]["status"] == "searching":

            users\_settings[str(message.from\_user.id)]["status"] = "registrated"

            await message.**answer**("Поиск отменён.", reply\_markup=**create\_buttons**())

**@dp.message**(Command(commands=["status"]))

async def **process\_status\_command**(message: Message):

    if str(message.from\_user.id) in users\_settings:

        if users\_settings[str(message.from\_user.id)]["status"] == "battling":

            return

        await message.**answer**(

            'Вот ваш статус:\n'

            f'побед: {users\_settings[str(message.from\_user.id)]["wins"]}\n'

            f'поражений {users\_settings[str(message.from\_user.id)]["loses"]}', reply\_markup=**create\_buttons**()

        )

    else:

        await message.**answer**("Сначала напишите /start, чтобы пройти регистрацию.")

# Этот хэндлер будет срабатывать на команду "/help"

**@dp.message**(Command(commands=['help']))

async def **process\_help\_command**(message: Message):

    markup = **create\_buttons**()

    await message.**answer**(

        "Нажмите на кнопку (Поиск противника), чтобы начать игру\n",

        "Игроки должны выбрать число, которое хотят загадать. Потом они начинают пытаться отгадать число противника.", reply\_markup=markup

    )

# Этот хэндлер будет срабатывать на любые ваши текстовые сообщения,

# кроме команд "/start" и "/help"

**@dp.message**()

async def **send\_echo**(message: Message):

    try:

        if users\_settings[str(message.from\_user.id)]["status"] == "registrating":

            users\_settings[str(message.from\_user.id)]["name"] = message.text

            users\_settings[str(message.from\_user.id)]["status"] = "registrated"

            markup = **create\_buttons**()

            await message.**reply**(text="Регистрация завершена", reply\_markup=markup)

**users\_settings\_change**()

    except:

        await message.**reply**(text="Напишите /start, чтобы пройти регистрацию")

        return

    if users\_settings[str(message.from\_user.id)]["status"] == "battling":

        id\_server = users\_settings[str(message.from\_user.id)]["battle\_server\_id"]

        await **battling\_players**(str(message.from\_user.id), message.text, id\_server)

            #print(battle\_servers[id\_server])

            #print(users\_settings)

    elif message.text == "Поиск противника" and users\_settings[str(message.from\_user.id)]["status"] != "battling":

        await **process\_battle\_command**(message)

    elif message.text == "Статус" and users\_settings[str(message.from\_user.id)]["status"] != "battling":

        await **process\_status\_command**(message)

    else:

        if users\_settings[str(message.from\_user.id)]["status"] == "battling":

            return

**print**(users\_settings[str(message.from\_user.id)]["status"])

        name = users\_settings[str(message.from\_user.id)]["name"]

        markup = **create\_buttons**()

        await message.**reply**(text=f"{name}: {message.text}", reply\_markup=markup)

**Файл «main.py»:**

from my\_bot import \*

if \_\_name\_\_ == '\_\_main\_\_':

    dp.**run\_polling**(bot)

Экранные формы с примерами выполнения программы

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание