**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторным работам №5

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

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-31Б |  | преподаватель каф. ИУ5 |
| Баринов А. А. |  | Гапанюк Ю. Е. |
| Подпись и дата: |  | Подпись и дата: |
|  |  |  |

Москва, 2023 г.

Задание:

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

Текст программы

App.py

from loader import bot, storage

async def on\_shutdown(dp):

await bot.close()

await storage.close()

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

from aiogram import executor

from handlers import dp

executor.start\_polling(dp, on\_shutdown=on\_shutdown)

loader.py

import asyncio

from aiogram import Bot, Dispatcher, types

from aiogram.contrib.fsm\_storage.memory import MemoryStorage

import config

loop=asyncio.get\_event\_loop()

bot = Bot(token=config.BOT\_TOKEN, parse\_mode=types.ParseMode.HTML)

storage = MemoryStorage()

dp = Dispatcher(bot, storage=storage, loop=loop)

handlers.py

from app import bot

from aiogram import types

from loader import dp

from aiogram.types import Message, ReplyKeyboardRemove

from aiogram.dispatcher.filters import Command, Text

from keyboards import menu

from Test import Test

from aiogram.dispatcher import FSMContext

import re

@dp.message\_handler(Command('start'))

async def Start(message: Message):

await message.answer('Введите команду /menu для выбора функции')

@dp.message\_handler(Command('menu'))

async def Vibor\_funkcii(message: Message):

await message.answer('Выберите функцию', reply\_markup=menu)

@dp.message\_handler(Text(equals=['Перевести число']), state=None)

async def Perevod(message: types.Message):

await message.answer('Введите натуральное число', reply\_markup=ReplyKeyboardRemove())

await Test.Q1.set()

@dp.message\_handler(state=Test.Q1)

async def answer\_q1(message: types.Message, state: FSMContext):

matched=re.match(r'^-?[0-9]\*$', message.text)

if not matched:

await message.reply('Число должно быть целым')

await message.answer('Пожалуйста, введите число повторно')

return

if message.text=='-':

await message.reply('- это не число')

await message.answer('Пожалуйста, введите число повторно')

return

if int(message.text) == 0 or int(message.text) == -0:

await message.reply('0 в любой системе счисления 0')

await message.answer('Пожалуйста, введите число повторно')

return

await state.update\_data(chislo=message.text.title())

await message.answer('Введите начальную систему счисления числа')

await Test.next()

@dp.message\_handler(state=Test.Q2)

async def answer\_q2(message: types.Message, state: FSMContext):

if message.text.isdigit() == False:

await message.reply('Система счисления должна быть наутральным числом и не равна 1')

await message.answer('Пожалуйста, введите начальную систему счисления числа повторно')

return

if int(message.text) == 1 or int(message.text) == 0:

await message.reply('Система счисления должна быть наутральным числом и не равна 1')

await message.answer('Пожалуйста, введите начальную систему счисления числа повторно')

return

y=int(message.text)

data = await state.get\_data()

answer1 = data.get('chislo')

x=int(answer1)

x1 = str(abs(x))

for i in range(1, len(x1) + 1):

if int(x1[i - 1]) >= y:

await message.reply('Такое число не может быть записано в данной системе счисления')

await message.answer('Пожалуйста, введите начальную систему счисления числа повторно')

return

await state.update\_data(SS1=message.text.title())

await message.answer('Введите СС, в которую нужно перевести число')

await Test.next()

@dp.message\_handler(state=Test.Q3)

async def answer\_q3(message: types.Message, state: FSMContext):

if message.text.isdigit() == False:

await message.reply('Система счисления должна быть наутральным числом и не равна 1')

await message.answer('Пожалуйства введите СС, в которую нужно перевести число повторно')

return

if int(message.text) == 1 or int(message.text) == 0:

await message.reply('Система счисления должна быть наутральным числом и не равна 1')

await message.answer('Пожалуйства введите СС, в которую нужно перевести число повторно')

return

data = await state.get\_data()

answer2 = data.get('SS1')

if int(message.text)==int(answer2):

await message.reply('Вы ввели одинаковые системы счисления')

await message.answer('Пожалуйства введите СС, в которую нужно перевести число повторно')

return

await state.update\_data(SS2=message.text.title())

data = await state.get\_data()

answer1 = data.get('chislo')

answer2 = data.get('SS1')

answer3 = data.get('SS2')

x = int(answer1)

y = int(answer2)

z = int(answer3)

x1 = str(abs(x))

x1 = x1[::-1]

x10 = abs(x)

if y == 10:

s = ''

while x10 != 0:

s = str(x10 % z) + str(s)

x10 //= z

if y != 10:

s = 0

for i in range(0, len(x1)):

s += int(x1[i]) \* (y \*\* i)

if y != 10 and z != 10:

s1 = ''

while s != 0:

s1 = str(s % z) + str(s1)

s //= z

s = s1

x1 = str(abs(x))

for i in range(1, len(x1) + 1):

if int(x1[i - 1]) >= y:

s = ('Такое число не может быть записано в данной системе счисления')

break

elif i == int(len(x1)):

s = s

if x < 0:

s = '-' + str(s)

text = str(s)

await message.answer(text=text)

await state.finish()

Test.py

from aiogram.dispatcher.filters.state import StatesGroup, State

class Test(StatesGroup):

Q1 = State()

Q2 = State()

Q3 = State()

Calculator.py

x = int(input())

y = int(input())

z = int(input())

s = 0

x1 = str(x)

x1 = x1[::-1]

if y == 10:

s = ''

while x != 0:

s = str(x % z) + str(s)

x //= z

if y != 10:

s = 0

for i in range(0, len(x1)):

s += int(x1[i]) \* (y \*\* i)

if y != 10 and z != 10:

s1 = ''

while s != 0:

s1 = str(s % z) + str(s1)

s //= z

s = s1

if y == z:

s = ('Вы ввели одинаковые системы счисления')

if y == 0 or z == 0:

s = ('Некорректные данные')

x1 = str(x)

for i in range(1, len(x1) + 1):

if int(x1[i - 1]) >= y:

s = ('Некорректные данные')

break

elif i == int(len(x1)):

s = s

print(s)

keyboards.py

from aiogram.types import ReplyKeyboardMarkup, KeyboardButton

menu = ReplyKeyboardMarkup(

keyboard=[

[

KeyboardButton(text='Перевести число')

]

],

resize\_keyboard=True

)

BOT\_TOKEN = "\*\*\*\*\*\*\*\*\*\*"

admin\_id = \*\*\*\*\*\*\*\*\*\*

Результат выполнения



