**Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации**

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

**Московский технический университет связи и информатики**

Кафедра «Информатики»

**ЛАБОРАТОРНАЯ РАБОТА №7**

**по дисциплине ВвИТ**

«Телебот»

Выполнил студент группы БИН2003 Мельникова Ю.В.

Проверил: Аршинов Е. А.

Москва 2021

**1. Цель работы:**

Создать своего бота в телеграмме, который будет выводить актуально расписание.

**2. Выполнение работы:**



import telebot

from utils.config import TOKEN

from commands.timetable import timetable

bot = telebot.TeleBot(TOKEN)

@bot.message\_handler(commands=["start"])

def start(message):

keyboard = telebot.types.ReplyKeyboardMarkup(True)

keyboard.row('Понедельник', 'Вторник', 'Среда', 'Четверг', 'Пятница', 'Суббота')

bot.send\_message(message.chat.id, 'Выберите день', reply\_markup=keyboard)

@bot.message\_handler(func=lambda message: True)

def rasp(message):

if (message.text.split()[0] in ["Эта\_неделя",'Понедельник', 'Вторник', 'Среда', 'Четверг', 'Пятница', 'Cуббота']):

print(message.text.split()[0])

rasp = timetable(message=message.text)

bot.send\_message(message.chat.id, rasp)

elif (message.text.split()[0] == 'help'):

keyboard = telebot.types.ReplyKeyboardMarkup(True)

keyboard.row('Понедельник', 'Вторник', 'Среда', 'Четверг', 'Пятница', 'Суббота')

text = """

sadasd

"""

bot.send\_message(message.chat.id, text, reply\_markup=keyboard)

@bot.message\_handler(commands=["help"])

def help(message):

keyboard = telebot.types.ReplyKeyboardMarkup(True)

keyboard.row('Понедельник', 'Вторник', 'Среда', 'Четверг', 'Пятница', 'Суббота')

text = """

помощь

"""

bot.send\_message(message.chat.id, 'lol', reply\_markup=keyboard)

bot.infinity\_polling()

from datetime import date

from db.operations import simple\_select

from db.db import conn, cur

week\_day = {'понедельник': range(0, 5),

'вторник': range(5, 10),

'среда': range(10, 15),

'четверг': range(15, 20),

'пятница': range(20, 25),

'суббота': range(25, 30)}

spaces = '&#4448;&#4448;&#4448;&#4448;&#4448;'

long\_lines = '---------------------'

short\_lines = '--------'

days = ['Понедельник (ОП)',

'Вторник (Мотор)',

'Среда (Мотор)',

'Четверг (ОП)',

'Пятница (ОП)',

'Суббота']

days\_with\_spaces = [f'{short\_lines}{day}{short\_lines}' for day in days]

time = ['| 09:30-11:05 |', '| 11:20-12:55 |', '| 13:10-14:45 |', '| 15:25-17:00 |', '| 17:15-18:50 |']

def timetable(message):

message\_array = message.split(' ')

delta = delta\_func()

nech = simple\_select(conn, cur, select\_what=['class\_name'], select\_from='qtimetable', where="week = 'неч'")

ch = simple\_select(conn, cur, select\_what=['class\_name'], select\_from='qtimetable', where="week = 'чет'")

week\_type = {'чет': ch, 'нечет': nech}

if len(message\_array) == 1 and message\_array[0] == '/Эта\_неделя':

rasp = ch if (delta // 7) % 2 != 0 else nech

day = []

for r in range(0, 30):

day.append(rasp[r])

text = rasp\_with\_time(day, 6)

return text

elif len(message\_array) == 1 and message\_array[0] != '/Эта\_неделя':

arg = message\_array[0].lower()

day = []

rasp = ch if (delta // 7) % 2 != 0 else nech

for r in week\_day[arg]:

day.append(rasp[r])

text = rasp\_with\_time(day, 1)

return text

elif len(message\_array) == 2:

arg = message\_array[1]

if arg in week\_day:

day = []

rasp = ch if (delta // 7) % 2 != 0 else nech

for r in week\_day[arg]:

day.append(rasp[r])

text = rasp\_with\_time(day, 1)

return text

elif arg in week\_type:

rasp = week\_type[arg]

day = []

for r in range(0, 30):

day.append(rasp[r])

text = rasp\_with\_time(day, 6, nofw=False)

return text

elif len(message\_array) == 3:

day\_of\_the\_week = message\_array[1]

type\_of\_the\_week = message\_array[2]

rasp = week\_type[type\_of\_the\_week]

day = []

for r in week\_day[day\_of\_the\_week]:

day.append(rasp[r])

text = f'{day\_of\_the\_week.capitalize()} | {type\_of\_the\_week.capitalize()}ная неделя\n{rasp\_with\_time(day, 1, nofw=False)}'

return text

def rasp\_with\_time(pr, mn, nofw=True):

if nofw:

delta = delta\_func()

week\_number = (delta // 7) + 1

week = f'Четная неделя ({week\_number})' if (delta // 7) % 2 != 0 else f'Нечетная неделя ({week\_number})'

text = f'{week}\n'

else:

text = ''

for i, item in enumerate(time \* mn):

if i % 5 == 0 and mn > 1:

text = text + days\_with\_spaces[i//5] + '\n'

if pr[i] != None:

text = text + str(item) + ' ' + pr[i] + '\n'

return text

def delta\_func():

first\_day = date(2021, 8, 30)

today = date.today()

delta = (today - first\_day).days

return delta

def simple\_select(conn, cur, select\_what, select\_from, where=None):

if where:

where\_expression = f'WHERE {where};'

else:

where\_expression = ''

select\_what\_str = ', '.join(select\_what)

cur.execute(f'''

SELECT ({select\_what\_str}) FROM {select\_from}

{where\_expression}

''')

result\_tuples = cur.fetchall()

result\_list = [item[0] if item[0] != 'None' else None for item in result\_tuples]

return result\_list