

**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 «Математическая кибернетика и информационные
технологии»**

**Лабораторная работа № 7
Создание telegram-бота с
расписанием
по дисциплине
«Введение в информационные технологии»**

**Выполнил: студент гр. БВТ2201
Шамсутдинов Р.Ф.**

Проверил:

Москва, 2023 г

Импортируем нужные модули в бот файл

```
Help
logic.py database.py bot.py x cfg.py
laba_7 > bot.py > ...
1 import asyncio
2 import logging
3
4 from aiogram import Bot, Dispatcher, Router, types
5 from aiogram.utils.keyboard import InlineKeyboardBuilder, KeyboardBuilder, ReplyKeyboardBuilder
6 from aiogram.filters import Command
7 from aiogram.types import Message
8
9 from cfg import Bot_Token
10 from database import get_schedule
11 from logic import get_date, sort_data, which_day_logic
12
```

Создаем два хендлера команд, первый для команды старт, и второй для команды командс, которая выводит клавиатуру с кнопками с периодами для которых вывести расписание

```
@router.message(Command(commands=["start"]))
async def command_start_handler(message):
    await message.answer(f"Здравствуйте, <b>{message.from_user.full_name}</b>! Чтобы узнать больше о боте, воспользуйтесь командой /help")

@router.message(Command(commands=["Commands"]))
async def command_Commands_handler(message):
    kb = [
        [
            types.KeyboardButton(text="Понедельник"),
            types.KeyboardButton(text="Вторник")
        ],
        [
            types.KeyboardButton(text="Среда"),
            types.KeyboardButton(text="Четверг")
        ],
        [
            types.KeyboardButton(text="Пятница"),
            types.KeyboardButton(text="Суббота")
        ],
        [
            types.KeyboardButton(text="Расписание на текущую неделю", ),
            types.KeyboardButton(text="Расписание на следующую неделю")
        ]
    ]
    keyboard = types.ReplyKeyboardMarkup(
        keyboard=kb,
        resize_keyboard=True,
        input_field_placeholder="Выберите период, на который хотите узнать расписание"
    )
    await message.answer(f"На какой период вывести расписание?", reply_markup=keyboard)
```

Создаем еще четыре хендлера, три для команд (/week, /mtuci, /help) и один для сообщений

```
@router.message(Command(commands=['week']))
async def command_week_handler(message):
    week = get_date()
    await message.answer(week)

@router.message(Command(commands=['mtuci']))
async def command_mtuci_handler(message):
    await message.answer('https://mtuci.ru/')

@router.message(Command(commands=['help']))
async def command_help_handler(message):
    ans = 'Приветствую вас, я - бот для просмотра расписания, написанный на aiogram.' + '\n'
    ans += 'Структура проекта состоит из 4 файлов, cfg.py содержит токен и пароль от бд, в database.py находится подключение к бд и работа с ней,\n'
    ans += 'в logic.py написаны вспомогательные функции для работы с данными и датой и в файле bot.py собраны все основные функции,\n'
    ans += 'также импортированы модули из предыдущих файлов, для запуска бота нужно запустить файл bot.py \n'
    ans += 'Нажав /Commands появится клавиатура с днями недели, нажмите на интересующий вас период времени и получите расписание на него,\n'
    ans += '/mtuci выведет ссылку на сайт МТУСИ, /week подскажет какая сейчас неделя'
    await message.answer(ans)

@router.message()
async def msg_handler(message):
    try:
        if message.text in ['Понедельник', 'Вторник', 'Среда', 'Четверг', 'Пятница', 'Суббота', 'Расписание на текущую неделю', 'Расписание на следующую неделю']:
            await command_schedule_handler(message)
        else:
            await message.answer('Извините, я вас не понял')
    except TypeError:
        await message.answer("Nice try!")
```

Создаем еще уже главный хендлер который будет выводить расписание, также создаем мейн функцию и запускаем бота с помощью конструкции иф нейм равно мейн

```
85 @router.message()
86 async def command_shedule_handler(message):
87
88     week = get_date()
89     data = sort_data(get_schedule())
90     msg = message.text
91     ans = which_day_logic(week, data, msg)
92
93     await message.answer(ans)
94
95
96 async def main():
97
98     dp = Dispatcher()
99     dp.include_router(router)
100     bot = Bot(TOKEN, parse_mode="HTML")
101     await dp.start_polling(bot)
102
103
104 if __name__ == "__main__":
105     logging.basicConfig(level=logging.INFO)
106     asyncio.run(main())
```

Файл для работы с базами данных:
импортируем нужный модуль, подключаемся с его помощью к
бд и создаем заготовки для заполнения таблиц в бд

```
database.py X bot.py M cfg.py
7 > database.py > ...
1 import psycopg2
2 from cfg import database_Password
3
4 conn = psycopg2.connect(database="schedule_bot", user="postgres",
5                           password=database_Password, host="localhost", port="5432")
6
7 cursor = conn.cursor()
8
9 subjects = (
10     ('English',), ('Higher mathematics',), ('German',), ('Introduction into IT',),
11     ('The basics of DevOps',), ('Gaming sports',), ('Physics',), ('History',),
12     ('Mathematical basis of databases',), ('Project Practice',)
13 )
14
15 teachers = (
16     ('Лапаев Л. Л.', 'English',),
17     ('Шаймарданова Л. К.', 'Higher mathematics',),
18     ('Зелкина Ю. М.', 'German',),
19     ('Фурлетов Ю. М.', 'Introduction into IT',),
20     ('Королёв И. В.', 'Gaming sports',),
21     ('Вальковский С. Н.', 'Physics',),
22     ('Скляр Л. Н.', 'History',),
23     ('Изотова А. А.', 'Mathematical basis of databases',),
24     ('Городничев М. Г.', 'The basics of DevOps',),
25     ('Потапченко П. Д.', 'Project Practice',)
26 )
```

```
28 timetables = (
29     ('Нечётная', 'Понедельник', 'English', '412', '13:10',),
30     ('Нечётная', 'Понедельник', 'Higher mathematics', '314', '15:25',),
31     ('Нечётная', 'Понедельник', 'German', '301', '17:15',),
32     ('Нечётная', 'Вторник', 'Introduction into IT', '203', '09:30',),
33     ('Нечётная', 'Вторник', 'The basics of DevOps', '302', '11:20',),
34     ('Нечётная', 'Среда', 'Gaming sports', 'спортзал', '11:20',),
35     ('Нечётная', 'Среда', 'Higher mathematics', '514', '13:10',),
36     ('Нечётная', 'Среда', 'Physics', '226', '15:25',),
37     ('Нечётная', 'Четверг', 'Physics', '301', '09:30',),
38     ('Нечётная', 'Четверг', 'History', '318', '11:20',),
39     ('Нечётная', 'Четверг', 'Gaming sports', 'спортзал', '13:10',),
40     ('Нечётная', 'Пятница', 'History', '227', '09:30',),
41     ('Нечётная', 'Пятница', 'Mathematical basis of databases', '535', '11:20',),
42     ('Нечётная', 'Пятница', 'Mathematical basis of databases', '401', '13:10',),
43     ('Нечётная', 'Пятница', 'Mathematical basis of databases', '519', '15:25',),
44     ('Чётная', 'Понедельник', 'Higher mathematics', '514', '11:20',),
45     ('Чётная', 'Понедельник', 'Higher mathematics', '308', '13:10',),
46     ('Чётная', 'Понедельник', 'Gaming sports', 'спортзал', '15:25',),
47     ('Чётная', 'Понедельник', 'English', '418', '17:15',),
48     ('Чётная', 'Вторник', 'Gaming sports', 'спортзал', '09:30',),
49     ('Чётная', 'Вторник', 'History', '318', '11:20',),
50     ('Чётная', 'Вторник', 'Physics', '340', '13:10',),
51     ('Чётная', 'Среда', 'The basics of DevOps', '414', '13:10',),
52     ('Чётная', 'Среда', 'The basics of DevOps', '302', '15:25',),
53     ('Чётная', 'Пятница', 'Project Practice', '211', '11:20',),
54     ('Чётная', 'Пятница', 'Introduction into IT', '203', '13:10',),
55     ('Чётная', 'Пятница', 'Introduction into IT', '208', '15:25',),
56 )
57
```


Функции создания таблиц

```

57
58
59 def create_table_subject():
60     cursor.execute('CREATE TABLE subject(name varchar(255) primary key);')
61     conn.commit()
62
63 def create_table_timetable():
64     cursor.execute('CREATE TABLE timetable (id serial primary key, week varchar(255),\
65     day varchar(255), subject varchar(255) references subject(name),\
66     room_num varchar(255), start_time varchar(255));')
67     conn.commit()
68
69 def create_table_teacher():
70     cursor.execute('CREATE TABLE teacher (id serial primary key, full_name varchar(1023),\
71     subject varchar(255) references subject(name));')
72     conn.commit()
73
74

```

Функции заполнения таблиц

```

74
75 def insert_into_subject(tuples):
76     cursor.executemany('INSERT INTO subject (name) VALUES (%s)', tuples)
77     conn.commit()
78
79 def insert_into_teacher(tuples):
80     cursor.executemany('INSERT INTO teacher (full_name, subject) VALUES (%s, %s)', tuples)
81     conn.commit()
82
83 def insert_into_timetable(tuples):
84     cursor.executemany('INSERT INTO timetable (week, day, subject, room_num, start_time) VALUES (%s, %s, %s, %s, %s)', tuples)
85     conn.commit()
86

```

Функция получения данных из таблиц и вызов нужных функций

```
conn.commit()

def get_schedule():
    cursor.execute('select t.id, t.week, t.day, t.subject, t.room_num, t.start_time,\n                    te.full_name from timetable t join subject s on s.name=t.subject join teacher te on s.name=te.subject order by 1;')
    records = list(cursor.fetchall())
    return records

if __name__ == '__main__':
    # create_table_subject()
    # create_table_timetable()
    # create_table_teacher()
    # insert_into_subject(subjects)
    # insert_into_teacher(teachers)
    # insert_into_timetable(timetables)
    # print(get_schedule())
    pass
```

Файл с логикой:

Функция для получения типа недели

```
laba_7 > logic.py > sort_data
1  import datetime
2
3
4  def get_date():
5
6      starting_date = datetime.date(2023,5,8)
7
8      current_datetime = datetime.datetime.now().date()
9
10     passed_days = (current_datetime - starting_date).days
11
12     week = 'Проблема с определением четности недели'
13     if 0 <= (passed_days % 14) <= 6:
14         week = 'Нечётная'
15     elif 7 <= (passed_days % 14) <= 13:
16         week = 'Чётная'
17
18     return week
19
```

Функция сортировки данных

```
20
21 def sort_data(schedule):
22     some = {
23         'Нечётная':
24             {
25                 'Понедельник': [],
26                 'Вторник': [],
27                 'Среда': [],
28                 'Четверг': [],
29                 'Пятница': [],
30                 'Суббота': []
31             },
32         'Чётная' :
33             {
34                 'Понедельник': [],
35                 'Вторник': [],
36                 'Среда': [],
37                 'Четверг': [],
38                 'Пятница': [],
39                 'Суббота': []
40             }
41     }
42     for item in schedule:
43
44         some[item[1]][item[2]].append(item[3::])
45
46     return some
47
```

Одна из самых важных функций для правильного вывода расписания на определенный период

```
48
49 def which_day_logic(week, schedule, msg):
50
51     result = '\n'
52     days_of_week = ['Понедельник', 'Вторник', 'Среда', 'Четверг', 'Пятница', 'Суббота']
53
54     if msg in days_of_week:
55
56         result += msg + '\n'
57         result += ('-----') + '\n'
58         day = schedule[week][msg]
59         count_par = len(day)
60
61         if count_par == 0:
62             result += ('Пар нет') + '\n'
63
64
65         elif count_par > 0:
66             for para in range(len(day)):
67                 result += (' '.join(day[para])) + '\n'
68             result += ('-----') + '\n'
69     elif msg == 'Расписание на текущую неделю':
70
```

```
71
72     result += (msg) + '\n'
73
74     for day_of_week in days_of_week:
75         result += (day_of_week) + '\n'
76         result += ('-----') + '\n'
77         day = schedule[week][day_of_week]
78         count_par = len(day)
79
80         if count_par == 0:
81             result += ('Пар нет') + '\n'
82
83         elif count_par > 0:
84             for para in range(count_par):
85                 result += (' '.join(day[para])) + '\n'
86
87         result += ('-----') + '\n' + '\n'
```

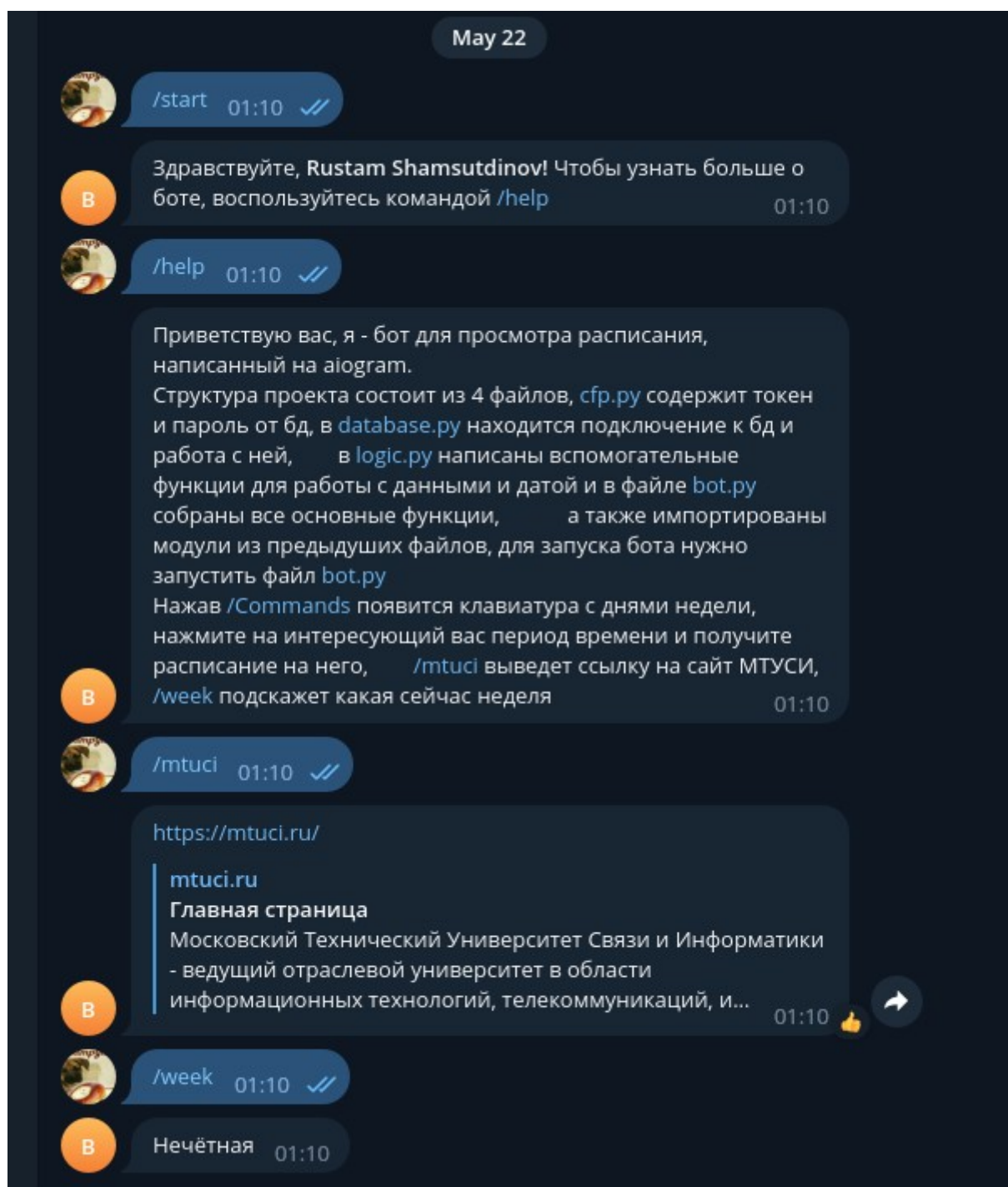


```


87
88 elif msg == 'Расписание на следующую неделю':
89
90     if week == 'Чётная':
91         week = 'Нечётная'
92
93     elif week == 'Нечётная':
94         week = 'Чётная'
95
96     result += (msg) + '\n'
97     for day_of_week in days_of_week:
98         result += (day_of_week) + '\n'
99         result += ('-----') + '\n'
100         day = schedule[week][day_of_week]
101         count_par = len(day)
102
103         if count_par == 0:
104             result += ('Пар нет') + '\n'
105
106         elif count_par > 0:
107             for para in range(count_par):
108                 result += (' '.join(day[para])) + '\n'
109             result += ('-----') + '\n' + '\n'
110
111     return result
112

```


Демонстрация работы бота



The screenshot shows a Telegram chat window with a dark theme. At the top, there's a header bar with a search icon and a notification bell. The chat history includes several messages from a bot named "/commands". The bot has sent messages for Monday, Tuesday, Wednesday, Thursday, and Friday, each containing a schedule of courses. The user "В" (V) has responded to the Monday message with "Извините, я Вас не понял" (Excuse me, I don't understand you). Below the chat history, there's a section titled "Выберите период, на который хотите узнать расписание" (Select the period for which you want to know the schedule). This section contains two columns of buttons: one for the current week ("Расписание на текущую неделю") and one for the next week ("Расписание на следующую неделю"). Each column has five buttons labeled with days of the week: Понедельник, Вторник, Среда, Четверг, and Суббота.



Расписание на текущую неделю

01:13 

Расписание на текущую неделю

Понедельник

English 412 13:10 Лапаев Л. Л.
Higher mathematics 314 15:25 Шаймарданова Л. К.
German 301 17:15 Зелкина Ю. М.

Вторник

Introduction into IT 203 09:30 Фурлетов Ю. М.
The basics of DevOps 302 11:20 Городничев М. Г.

Среда


Gaming sports спортзал 11:20 Королёв И. В.
Higher mathematics 514 13:10 Шаймарданова Л. К.
Physics 226 15:25 Вальковский С. Н.

Четверг



Physics 301 09:30 Вальковский С. Н.
History 318 11:20 Скляр Л. Н.
Gaming sports спортзал 13:10 Королёв И. В.

Пятница

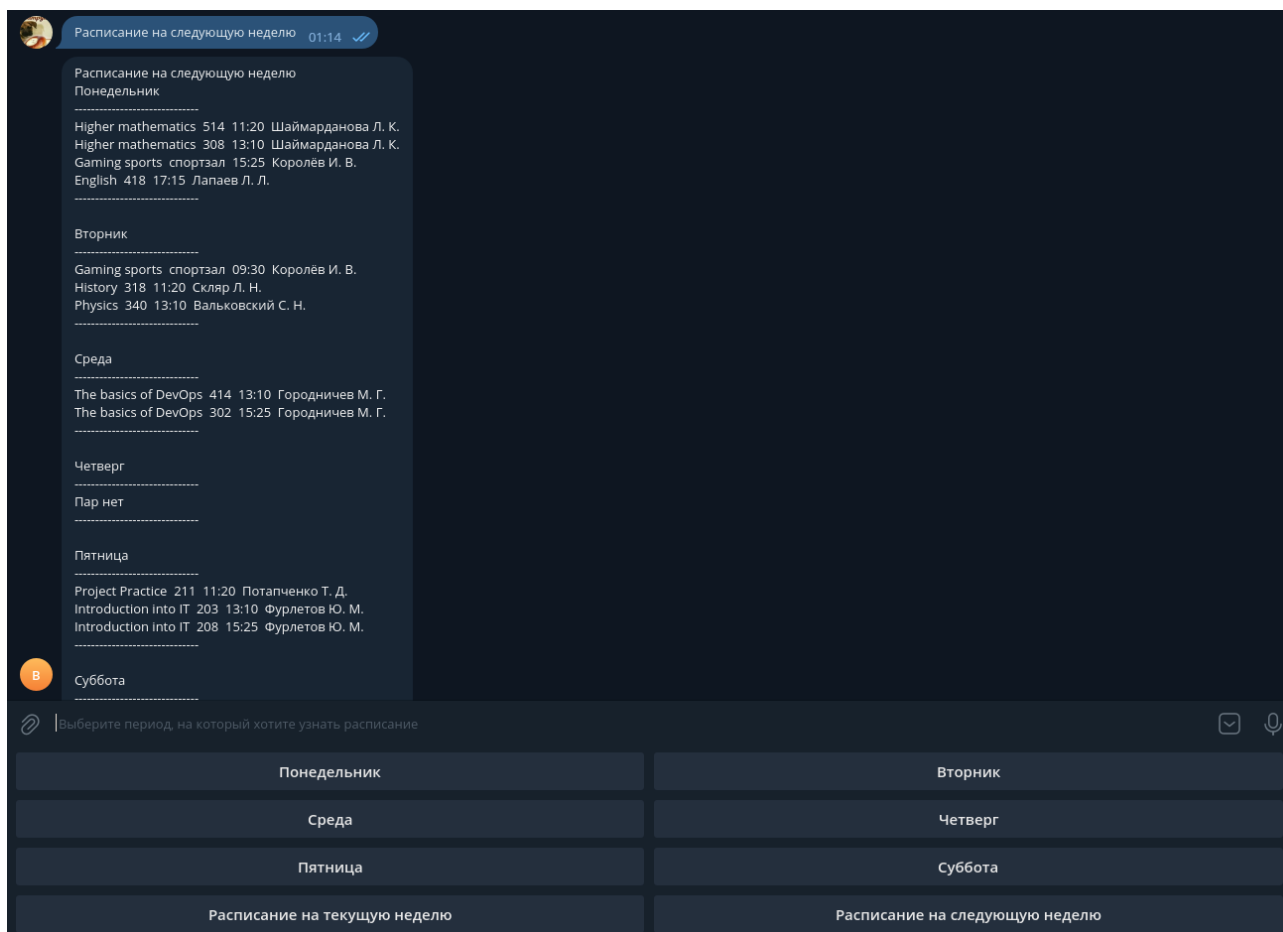
History 227 09:30 Скляр Л. Н.
Mathematical basis of databases 535 11:20 Изотова А. А.
Mathematical basis of databases 401 13:10 Изотова А. А.
Mathematical basis of databases 519 15:25 Изотова А. А.



Выберите период, на который хотите узнать расписание

Понедельник	Вторник
Среда	Четверг
Пятница	Суббота
Расписание на текущую неделю	Расписание на следующую неделю



Вывод: проделав данную работу, я познакомился с такой библиотекой как `aiogram`, научился создавать телеграм бота для вывода расписания с удобными командами и клавиатурой, также я попробовал себя в роли архитектора и попытался разбить код на несколько файлов, каждый из которых содержал в себе функции, который выполняли определенный ряд задач.

https://github.com/Sypoo1/VVIT/tree/main/laba_7