**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**

**Отчет по лабораторной работе № 6**

**“Создание Телеграм бота”**

**по дисциплине**

**"Введение в ИТ"**

Выполнил: студент гр. БИН2001 Казаков А.А.

Проверил: Аршинов Е.А.

Москва, 2021 г.

1. token = 'Токен бота'
3. db\_connect = { #настройки подключения к базе данных
4. 'database':'schedule',
5. 'user':'postgres',
6. 'password':'0000',
7. 'host':'localhost',
8. 'port':'5432'
9. }
11. buttons = { #Кнопки в меню Бота
12. 'start\_menu' : ['Сегодня', 'Завтра', 'Полное расписание'],
13. 'odd\_menu' : ['Расписание на текущую неделю', 'Расписание на следующую неделю'],
14. 'day\_menu' : ['Понедельник', 'Вторник', 'Среда', 'Четверг', 'Пятница']
15. }

Файл config.py

1. #Определение чётности недели
2. def odd\_date\_check():
3. now = datetime.now()
4. sep = datetime(now.year if now.month >= 9 else now.year - 1, 9, 1)
5. d1 = sep - timedelta(days=sep.weekday())
6. d2 = now - timedelta(days=now.weekday())
7. parity = not bool(((d2 - d1).days // 7) % 2)
8. return parity
9. #Определение текущего дня недели
10. def now\_date():
11. return datetime.today().isoweekday() - 1
13. #Получение текущей даты и чётности недели в указанным сдвигом дней
14. def now\_data(time\_delta = 0):
15. odd = odd\_date\_check() *# чётно нечётно*
16. day = now\_date() *# день недели*
17. day += time\_delta
18. if time\_delta == 7:
19. time\_delta = 0
20. odd = not odd
21. return odd, day
23. #Преобразование данных типа Время в текстовый вид
24. def timedelta\_to\_str(td):
25. return ':'.join(str(td).split(':')[:2])
27. #Преобразование массива данных в более визуально приемлемый вид
28. def map\_rasp(dataset):
29. def f\_map\_rasp(data):
30. p = list()
31. lesson\_delta = timedelta(hours=1, minutes=40)
32. data\_timedelta = datetime.combine(date.min, data[4]) - datetime.min
33. timedelta\_to\_str(data\_timedelta)
34. p.append('\*' + str(['9:30', '11:20', '13:10', '15:25', '17:15'].index(timedelta\_to\_str(data\_timedelta)) + 1) + ')' + '\*')
35. p.append('('+str(timedelta\_to\_str(data[4]))+'-'+ str(timedelta\_to\_str(data\_timedelta + lesson\_delta))+')')
36. p.append(data[2])
37. p.append('**\n**')
38. *#p.append(str(data[4]))*
39. if data[3] == None:
40. p.append('аудитория нн')
41. else:
42. p.append(data[3])
43. if data[6] == None:
44. p.append('преподаватель нн')
45. else:
46. p.append(data[6])
47. return ' '.join(p)
48. *#print(dataset)*
49. return map(f\_map\_rasp, dataset)
51. #Получение расписание на выбранный день из базы данных
52. def get\_rasp(is\_even = None, day = None):
53. *#print(day, is\_even)*
54. if is\_even == None or day == None:
55. is\_even, day = now\_data()
56. cursor.execute('''
57. SELECT timetable.\*, teacher.full\_name
58. FROM bin2001.timetable timetable
59. LEFT JOIN bin2001.teacher teacher
60. ON teacher.subject = timetable.subject
61. WHERE day = %s AND is\_even = %s
62. ORDER BY start\_time
63. ''', (int(day), bool(is\_even)))
64. rt = ['\*' + config.buttons['day\_menu'][day] + ' ' + ("чётная" if is\_even else "нечётная") + '\*']
65. rt.extend(list(map\_rasp(cursor.fetchall())))
66. return '**\n**'.join(rt)

Файл functions.py

В файле main.py импортируются все необходимые библиотеки и файлы config.py и functions.py, происходит обработка сообщений пользователя вызов соответствующих функций и отправка необходимого ответа

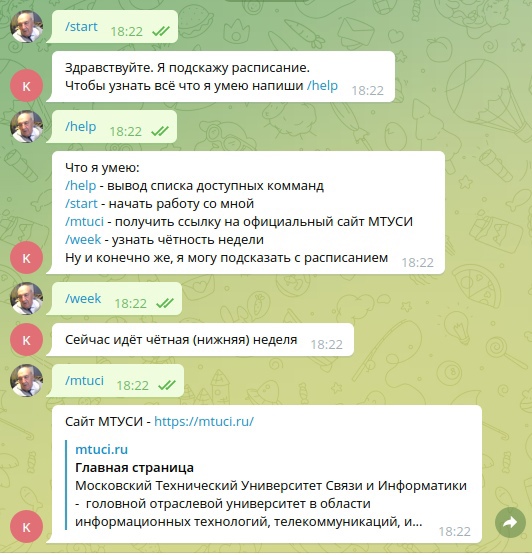


Рисунок 1 - Команды бота

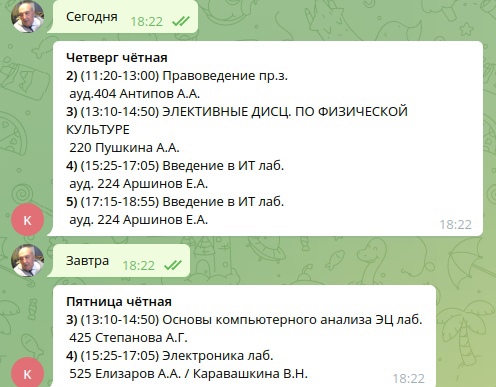


Рисунок 2 - Ввывод расписания на сегодня/завтра

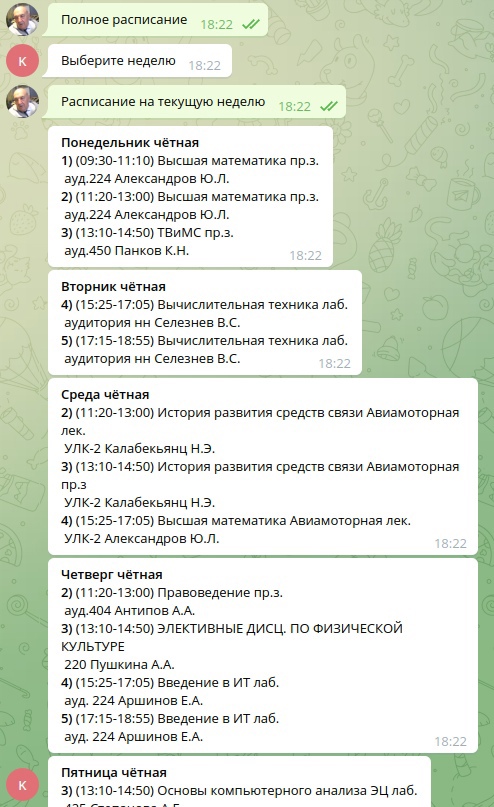


Рисунок 3 - Вывод расписания на текущую неделю

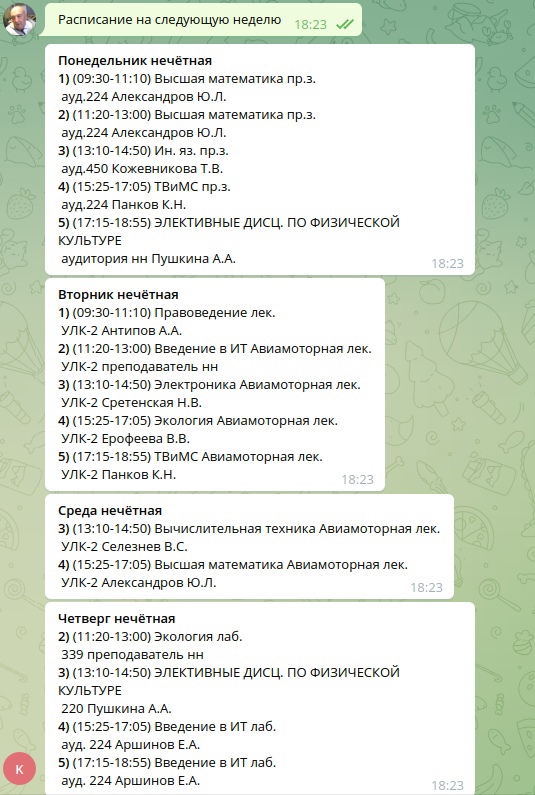


Рисунок 4 - Вывод расписания на следующую неделю