

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский Авиационный Институт» (Национальный
Исследовательский Университет)

Институт: №8 «Информационные технологии и прикладная математика»
Кафедра: 806 «Вычислительная математика и программирование»

Реферат
по курсу «Фундаментальная информатика»
I семестр
Тема: «Язык программирования
Python на примере разработки
Telegram-бота для отслеживания
приёма лекарств»

Группа:	М8О-109Б-22
Студент:	Федоров А. А.
Преподаватель:	Сысоев М. А.
Оценка:	
Дата:	

Москва, 2022

Содержание

- 1) История Python и сферы применения
- 2) Основные преимущества и недостатки языка
- 3) Telegram-бот для отслеживания приёма лекарств
- 4) Примеры работы бота
- 5) Выводы
- 6) Список литературы

1. История Python и сферы применения

Python — высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ. Язык является полностью объектно-ориентированным в том плане, что всё является объектами. Необычной особенностью языка является выделение блоков кода пробельными отступами. Синтаксис ядра языка минималистичен, за счёт чего на практике редко возникает необходимость обращаться к документации. Сам же язык известен как интерпретируемый и используется в том числе для написания скриптов. Недостатками языка являются зачастую более низкая скорость работы и более высокое потребление памяти написанных на нём программ по сравнению с аналогичным кодом, написанным на компилируемых языках, таких как С или С++.

Разработку Python начал Гвидо ван Россум в декабре 1989 года. Для ОС Амоеба требовался расширяемый скриптовый язык. На досуге Гвидо начал писать Python, позаимствовав некоторые наработки из языка ABC.

Где применяется сейчас?

Он применяется во многих проектах и в качестве основного языка, и для создания расширений и интеграции приложений. А ещё на Python пишут прототипы будущих программ.

Три основные сферы использования языка — это веб-разработка, Big Data и Machine Learning и написание скриптов:

- **Веб-разработка.**

Сегодня Python активно используется в веб-разработке наряду с такими языками, как PHP, JavaScript, Java и C#. Существенный плюс Python состоит в том, что он подходит для небольших бизнесов, у которых нет отдела разработки на 50 человек и которые

реализуют относительно несложные проекты. Поэтому к нему стали все чаще обращаться.

Преимущества использования Python для бэкенда по сравнению с PHP — простота обучения, улучшенная структура, легкодоступные инструменты для отладки языка, универсальность и наличие дружественного комьюнити.

Для веб-разработки на Python понадобится знание фреймворков. Наиболее популярные — Django и Flask.

- **Data Science и Machine Learning.**

Python в том числе используют и для анализа данных, и чтобы писать алгоритмы ML-программ и в аналитических приложениях. Среди популярных фреймворков и библиотек для ML программисты выделяют scikit-learn и TensorFlow.

- **Написание скриптов.**

Скриптинг — написание скриптов для автоматизации простых задач. Например, это могут быть дополнительные модули или плагины к уже готовому программному обеспечению. Также Python используют для автоматизации тестирования ПО.

Помимо этого, Python применяется для написания и разработки программных и мобильных приложений, во встроенных системах, системном администрировании и даже в геймдеве

2. Основные преимущества и недостатки языка

Преимущества:

- Простота синтаксиса
Код языка чистый и понятный, без лишних символов и выражений.
- Расширяемость и гибкость
Не зря один из слоганов языка — это «Just Import!» Python можно легко расширить для взаимодействия с другими программными системами или встроить в программы в качестве компонента. Он очень и очень гибкий. Это даёт больше возможностей для использования языка в разных сферах.
- Интерпретируемость и кроссплатформенность.
- Стандартизированность.

- Open Source.
- Широта применения.

Недостатки:

- **Низкая производительность.**

Python требует высоких вычислительных мощностей серверов и компьютеров. Это делает его не таким быстрым, как хотелось бы, он отстаёт от других языков по части производительности. В условиях сильного развития мощностей это не так заметно, как раньше, но всё равно даёт о себе знать. Чтобы нивелировать эту проблему, разработчики обращаются к C для реализации проблемного участка кода.

- **Динамическая типизация.**

Python относится к языкам с динамической типизацией, что делает его невероятно гибким при разработке. Но при этом он потребляет много ресурсов и имеет низкую скорость выполнения программы.

- **Синтаксис и семантика.**

Хотя многие считают синтаксис преимуществом Python, есть ряд архитектурных решений, которые могут сбивать с толку программистов, привыкших работать на других языках. Но это станет недостатком в использовании Python только для опытного программиста, а не для новичка.

3. Telegram-бот для отслеживания приёма лекарств

Основное предназначение Telegram-бота – это отслеживание приёма лекарств. Пользователь может добавить необходимое лекарство и бот каждый день на протяжении указанного срока (в днях) будет присылать сообщение-напоминание о том, что нужно принять указанный препарат в установленном количестве.

Бот находится по адресу: https://t.me/pill_reminder_sp_bot

У бота предусмотрены следующие команды (иные сообщения игнорируются моим обработчиком, ошибочные использования заданных команд не с тем форматом также предусмотрены)

- 1) Для старта работы и предоставления руководства по использованию бота

/start

- 2) Для добавления препарата

/add {Название препарата}, {Количество препарата в один приём с единицами измерения}, {Время}, {Количество дней приёма}
Таким образом, каждый день в указанное время бот будет отправлять сообщение о необходимости приёма данного лекарства.

Пример команды: /add Парацетамол, 1 таблетка, 10:00, 3

- 3) Для просмотра текущего списка принимаемых препаратов

/show

- 4) Для удаления лекарства из списка принимаемых препаратов

/del {Уникальный номер из списка препаратов, который можно запросить при помощи команды /show}

Пример команды: /del 1

Реализация каждой из команд:

- 1) /start

```
11 async def handle_start(message: Message):
12     await message.answer(f'Привет, {message.from_user.first_name}! Я помогу вам пить лекарства вовремя\n\n' +
13                          'Мои команды:\n' +
14                          '• Для добавления препарата используйте команду \n' +
15                          '/add {Название препарата}, {Количество препарата в один приём с единицами измерения}, '
16                          '{Время}, {Количество дней приёма}\n' +
17                          'Таким образом, каждый день в указанное время я буду вам напоминать о приёме данного лекарства\n' +
18                          'Пример команды: /add Парацетамол, 1 таблетка, 10:00, 3\n' +
19                          '• Для просмотра текущего списка принимаемых препаратов используйте команду /show\n' +
20                          '• Для удаления лекарства из списка принимаемых препаратов используйте команду '
21                          '/del {Уникальный номер из списка препаратов, который можно запросить при помощи команды /show}\n' +
22                          'Пример команды: /del 1')
23
```

2) /add

```
63 async def handle_add_reminder(message: Message, command: CommandObject):
64     try:
65         parts_of_text = command.args.split(',')
66         parts_of_text = [x.strip() for x in parts_of_text]
67         pill_name = parts_of_text[0]
68         pill_dose = parts_of_text[1]
69         pill_time = parts_of_text[2]
70         pill_days = int(parts_of_text[3])
71         user_id = message.from_user.id
72         current_pill_id = None
73         if (user_id not in users) or (users[user_id]['pills'] == {}):
74             users[user_id] = {
75                 'pills': {
76                     1: [pill_name, pill_dose, pill_time, pill_days]
77                 }
78             }
79             current_pill_id = 1
80         else:
81             current_pill_id = max(users[user_id]['pills'].keys()) + 1
82             users[user_id]['pills'][current_pill_id] = [pill_name, pill_dose, pill_time, pill_days]
83
84         await message.answer("Препарат успешно добавлен")
85
86     pill_time_hour, pill_time_minute = (int(x) for x in pill_time.split(":"))
87     for i in range(pill_days):
88         now = datetime.datetime.now()
89         next_reminder = datetime.datetime(now.year, now.month, now.day, pill_time_hour, pill_time_minute, now.second)
90         if (next_reminder > now):
91             delay = (next_reminder - now).seconds
92         else:
93             next_reminder += datetime.timedelta(days=1)
94             delay = (next_reminder - now).seconds
95         await asyncio.sleep(delay)
96         if (current_pill_id in users[user_id]['pills']):
97             users[user_id]['pills'][current_pill_id][3] -= 1
98             await message.answer(f'! {message.from_user.first_name}, пришло время принять препарат "{pill_name}", количество: {pill_dose} !')
99         else:
100             break
101     if current_pill_id in users[user_id]['pills']:
102         del users[user_id]['pills'][current_pill_id]
103     except Exception as e:
104         await message.answer(f'Команда /add использована неверно!')
```

3) /show

```
47 async def handle_show(message: Message):
48     current_answer = 'Принимаемые вами препараты:\n'
49     current_answer += '<b>Формат: Уникальный номер) Название препарата, Количество препарата в один приём с единицами измерения, Время, Количество дней приёма</b>\n\n'
50     user_id = message.from_user.id
51     if user_id in users:
52         if len(users[user_id]['pills']):
53             for id in users[user_id]['pills']:
54                 current_answer += f'{id}) ' + ', '.join(str(x) for x in users[user_id]['pills'][id]) + '\n'
55         else:
56             current_answer = 'Нет принимаемых препаратов 🙄'
57     else:
58         current_answer = 'Нет принимаемых препаратов 🙄'
59     await message.answer(current_answer, parse_mode='HTML')
```

4) /del

```

25 async def handle_delete(message: Message, command: CommandObject):
26     try:
27         pill_id_to_del = int(command.args)
28         user_id = message.from_user.id
29         current_answer = ' '
30         if user_id in users:
31             if (('pills' in users[user_id]) and
32                 (users[user_id]['pills'] != {})):
33                 if (pill_id_to_del in users[user_id]['pills']):
34                     del users[user_id]['pills'][pill_id_to_del]
35                     current_answer = "Препарат успешно удален"
36                 else:
37                     current_answer = "Указанный уникальный номер принимаемого препарата отсутствует в списке!"
38             else:
39                 current_answer = 'Нет принимаемых препаратов 🙄'
40         else:
41             current_answer = 'Нет принимаемых препаратов 🙄'
42         await message.answer(current_answer)
43     except Exception as e:
44         await message.answer("Команда /del использована неверно!")

```

Реализация обработчика команд:

```

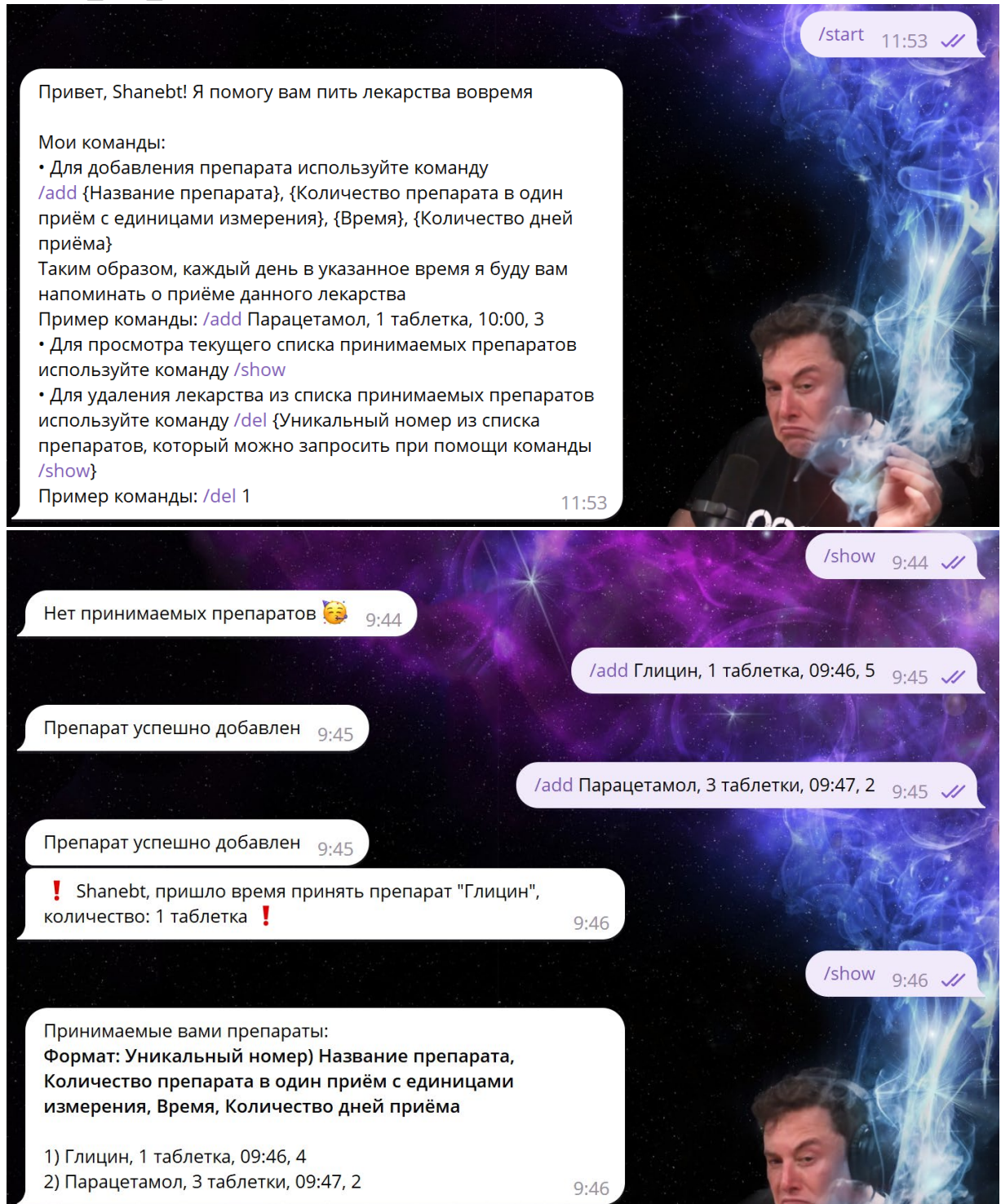
107 async def start():
108     bot = Bot(token=token)
109     dp = Dispatcher()
110     dp.message.register(handle_start, Command(commands='start'))
111     dp.message.register(handle_add_reminder, Command(commands='add'))
112     dp.message.register(handle_show, Command(commands='show'))
113     dp.message.register(handle_delete, Command(commands='del'))
114     try:
115         await dp.start_polling(bot)
116     finally:
117         await bot.session.close()
118
119 if __name__ == '__main__':
120     asyncio.run(start())
121

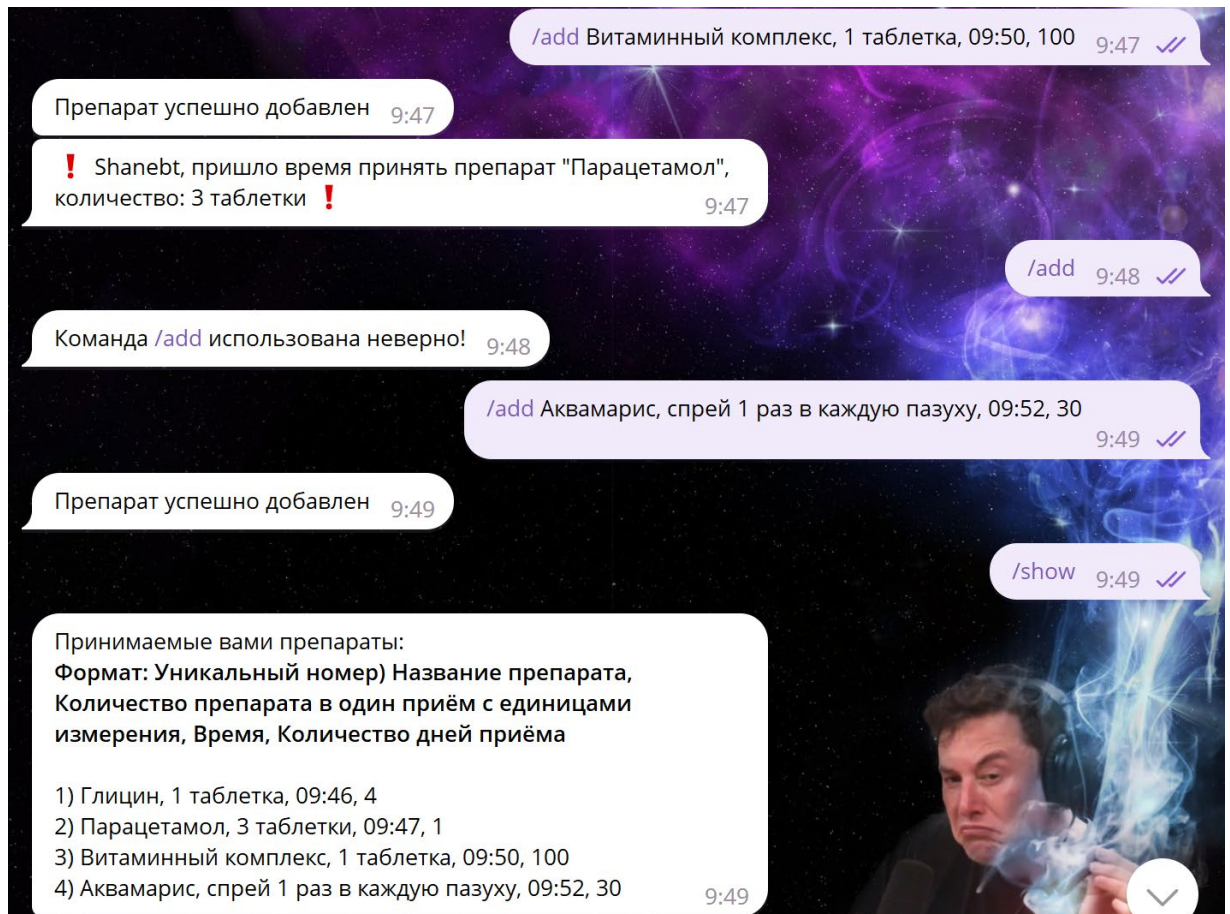
```

4. Примеры работы бота

Сценарии работы бота представлены на скриншотах. В данном случае представлены примеры, когда время уведомления находится рядом с

текущим временем, т. к. это быстрее с точки зрения отладки и демонстрации. По сути же это ничем не отличается, если бы уведомление надо было доставлять в следующий день, а не в этот же, обе ситуации предусмотрены в моём коде в функции `handle_add_reminder()`





! Shanebt, пришло время принять препарат "Витаминный комплекс", количество: 1 таблетка !

9:50

/add Аквамарис, спрей 1 раз в каждую пазуху, 18:01, 30

9:50 ✓✓

Препарат успешно добавлен 9:50

/show 9:51 ✓✓

Принимаемые вами препараты:
Формат: Уникальный номер) Название препарата,
Количество препарата в один приём с единицами
измерения, Время, Количество дней приёма

- 1) Глицин, 1 таблетка, 09:46, 4
- 2) Парацетамол, 3 таблетки, 09:47, 1
- 3) Витаминный комплекс, 1 таблетка, 09:50, 99
- 4) Аквамарис, спрей 1 раз в каждую пазуху, 09:52, 30
- 5) Аквамарис, спрей 1 раз в каждую пазуху, 18:01, 30

9:51

! Shanebt, пришло время принять препарат "Аквамарис", количество: спрей 1 раз в каждую пазуху !

9:52

/show 9:53 ✓✓

Принимаемые вами препараты:
Формат: Уникальный номер) Название препарата,
Количество препарата в один приём с единицами
измерения, Время, Количество дней приёма

- 1) Глицин, 1 таблетка, 09:46, 4
- 2) Парацетамол, 3 таблетки, 09:47, 1
- 3) Витаминный комплекс, 1 таблетка, 09:50, 99
- 4) Аквамарис, спрей 1 раз в каждую пазуху, 09:52, 29
- 5) Аквамарис, спрей 1 раз в каждую пазуху, 18:01, 30

9:53

/show 9:53 ✓✓

/del 2 9:53 ✓✓

Препарат успешно удален 9:53

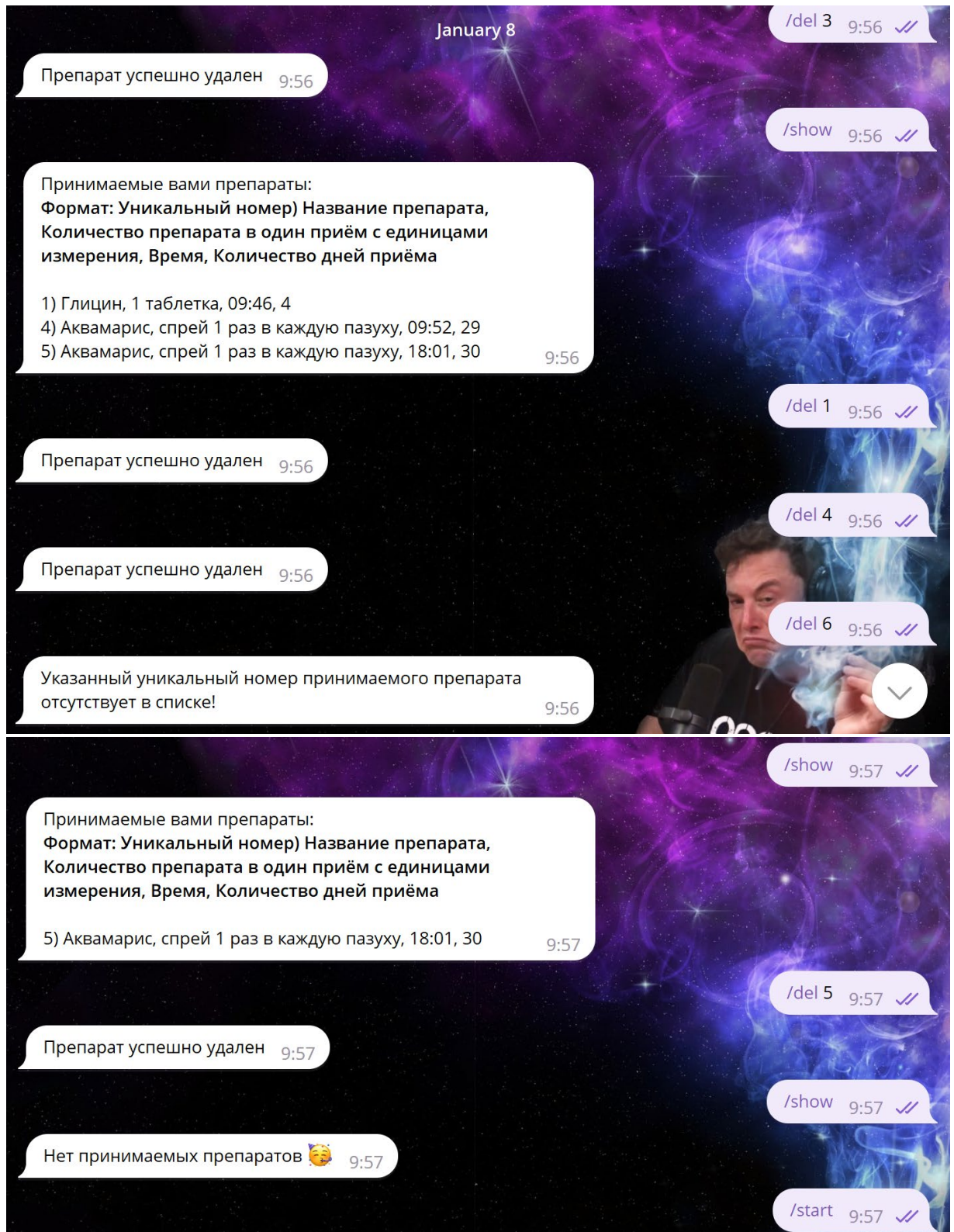
/show 9:53 ✓✓

Принимаемые вами препараты:
Формат: Уникальный номер) Название препарата,
Количество препарата в один приём с единицами
измерения, Время, Количество дней приёма

- 1) Глицин, 1 таблетка, 09:46, 4
- 3) Витаминный комплекс, 1 таблетка, 09:50, 99
- 4) Аквамарис, спрей 1 раз в каждую пазуху, 09:52, 29
- 5) Аквамарис, спрей 1 раз в каждую пазуху, 18:01, 30

9:53





5. Выводы

Мне понравилась данная работа, т. к. она дала возможность узнать мне как работать с Telegram-ботами. Было и интересно, и сложно (в моментах с непонятной документацией или при возникновении ошибок), и полезно, кроме того в очередной раз я столкнулся с проблемами в самоорганизации и понял, что лучше начинать работу заранее и не тянуть до конца.

6. Список литературы

- 1) <https://github.com/hedgefo9/Pill-Reminder> - репозиторий с программой на GitHub
- 2) <https://docs.aiogram.dev/en/dev-3.x/> - документация к библиотеке для взаимодействия с Telegram Bot API
- 3) <https://ru.wikipedia.org/wiki/Python> - справочная информация о языке Python
- 4) <https://practicum.yandex.ru/blog/vsyo-o-yazyke-programmirovaniya-python/> - справочная информация о языке Python