Муниципальное бюджетное образовательное учреждение

«Средняя общеобразовательная школа № 31»

поселка Краснобродского

**Создание бота, как искусственного интеллекта,**

**посредством языка программирования Python**

*Проектная работа*

Выполнил:

Мамочкин Матвей

учащийся10 класса «Б»

Руководитель:

Мамочкина М.В.,

учитель информатики

Краснобродский

2019

Оглавление

[Введение 3](#_Toc5549182)

[История создания искусственного интеллекта 4](#_Toc5549183)

[Бот - искуственный интеллект 6](#_Toc5549184)

[Чат-бот с нуля 9](#_Toc5549185)

[Язык программирования Python 9](#_Toc5549186)

[База готовых команд VKAPI 12](#_Toc5549187)

[Создание бота на языке Python 16](#_Toc5549188)

[Создаем группу ВК 16](#_Toc5549189)

[Начинаем писать код 17](#_Toc5549190)

[Заключение 31](#_Toc5549191)

[Список литературы 32](#_Toc5549192)

# Введение

Современный мир немыслим без интернет-технологий, которые постоянно развиваются. То, что вчера казалось фантастикой, сегодня становится реальностью. Особенно бурно развиваются новые системы искусственного интеллекта. Виртуальный собеседник перестает быть живым человеком, на его смену приходят мощные программы. Их функция не только развлекать, но и быть надежным помощником в повседневной работе. Таковы чат-боты – самая прогрессивная и перспективная технология в мире интернета.

**Цель работы:** создание чат-бота в социальной сети Вконтакте, с помощью которого можно смотреть программу телепередач на разных каналах, играть с компьютером в игры "Города" и "Угадай число"

З**адачи:**

1) изучить историю создания искуственного интеллекта

2) Изучить и выбрать инструменты для создания чат-ботов;

3)Написать код для чат-бота Вконтакте

**Предметы изучения**: социальная сеть Вконтакте, язык программирования Python, APIVk (набор готовых классов, процедур, функций, структур и констант, предоставляемых социальной сетью)

**Актуальность работы**: созданный чат-бот можно использовать для развлечений и отслеживания телепередач в социальной сети с различных электронных устройств.

# История создания искусственного интеллекта

С конца 40-х годов ученые все большего числа университетских и промышленных исследовательских лабораторий устремились к дерзкой цели: построение компьютеров, действующих таким образом, что по результатам работы их невозможно было бы отличить от человеческого разума.

Терпеливо продвигаясь вперед в своем нелегком труде, исследователи, работающие в области искусственного интеллекта (ИИ), обнаружили, что вступили в схватку с весьма запутанными проблемами, далеко выходящими за пределы традиционной информатики. Оказалось, что прежде всего необходимо понять механизмы процесса обучения, природу языка и чувственного восприятия. Выяснилось, что для создания машин, имитирующих работу человеческого мозга, требуется разобраться в том, как действуют миллиарды его взаимосвязанных нейронов. И тогда многие исследователи пришли к выводу, что пожалуй самая трудная проблема, стоящая перед современной наукой - познание процессов функционирования человеческого разума, а не просто имитация его работы. Что непосредственно затрагивало фундаментальные теоретические проблемы психологической науки. В самом деле, ученым трудно даже прийти к единой точке зрения относительно самого предмета их исследований - интеллекта.

Искусственным интеллектом (ИИ) — способность интеллектуальных машин выполнять творческие функции, которые традиционно считаются прерогативой человека. Также этим термином обозначают науку и технологию создания интеллектуальных машин. Впервые это определение было дано американским информатиком Джоном Маккарти в 1956 году.

В английском языке словосочетание artificialintelligence не имеет той слегка фантастической антропоморфной окраски, которую оно приобрело в довольно неудачном русском переводе. Слово intelligence означает «умение рассуждать разумно», а вовсе не «интеллект», для которого есть английский аналог intellect.

Из-за неоднозначности самого понятия «интеллект» и даже оригинального термина «intelligence», существуют и другие дефиниции. Например, информационные технологии ИИ определяются как способность системы создавать в ходе самообучения программы (в первую очередь эвристические) для решения задач определенного класса сложности и решать эти задачи.

Эмпирический тест на определение искусственного интеллекта был предложен Аланом Тьюрингом в 1950 году, когда такого термина еще не существовало. Согласно этому тесту, мыслящей машиной считается та, которую человек в беседе с ней принял за человека.

Существует множество подходов к созданию систем искусственного интеллекта. Наиболее популярен из них эвристический подход, при котором машина при помощи нейронной сети познает окружающий мир методом проб и ошибок. Впервые идею о вероятностной машине, «обучающейся без учителя», высказал РэйСоломонофф в 1956 году на той же Дартмундской конференции, на которой Джон Маккарти стал родителем термина «искусственный интеллект».

В настоящий момент в области искусственного интеллекта наблюдается вовлечение многих предметных областей, имеющих скорее практическое отношение к ИИ, а не фундаментальное.

# Бот - искуственный интеллект

Бот — это искусственный интеллект, который имитирует некоторые действия человека. С его помощью можно автоматизировать рутинные задачи.

Обычно боты предназначаются для выполнения однообразной и повторяемой работы, с максимально возможной скоростью.

Первые эксперименты по созданию подобных программ начались еще полвека назад. Один из успешных примеров придания взаимодействию с машиной иллюзии человеческого общения – Элиза, написанный в 1966 году чат-бот, имитирующий (или, скорее, пародирующий) диалог с психотерапевтом.

Некогда в популярнейшем мессенджере ICQ существовали десятки ботов, выполнявших функции сборника анекдотов, переводчика, калькулятора, присылавших по запросу или расписанию прогноз погоды, курсы валют, новости, киноафишу и т.д. Для решения элементарных задач пользователю не нужно было покидать мессенджер и открывать браузер – нередко в офисах их блокировали или ограничивали внешний веб-трафик.

Заказать билеты, вызвать такси, сделать платеж или перевод, построить маршрут, скачать фильм или музыку тоже помогут боты.

**В чем же причина такого интереса к ним в последнее время?**

Рынок мобильных приложений перенасыщен: их счет пошел на миллионы, но пользователи уже не горят желанием устанавливать что-то новое. Согласно исследованиям ComScore, 80% времени пользователи проводят всего в трех приложениях. На этом фоне сегмент мессенджеров продолжает активно расти. В прошлом году совокупная аудитория самых популярных из них обогнала самые популярные соцсети. Мессенджеры приватны, требуют меньше ресурсов, работают на более дешевых устройствах и пока еще, в отличие от соцсетей, не забиты лишней информацией, навязчивой рекламой и чужими новостями.

В итоге разработчикам сервисов стало очевидно, что проще добраться до пользователя в той программе, которую он и так установил и ежедневно открывает, чем убеждать его в необходимости работы с отдельным приложением. Чат-бот не требует трафика для скачивания, времени на установку, не занимает места в памяти и на экране смартфона. Так начался бум чат-ботов.

Работать с чат-ботом просто: нужно добавить его в список контактов и начать переписку. Чаще всего в ответ бот пришлет информацию о себе, список доступных команд или выведет на экран кнопки, превращающие окно диалога в интуитивно-понятное мини-приложение.

Обзоры новых ботов регулярно появляются в медиа, как лет пять назад это происходило с новыми приложениями в AppStore. Удобнее использовать специальные каталоги ботов: например, Telegrambotstore, где боты ранжированы по популярности, свежести, языкам и категориям. Развитие экосистемы ботов – вопрос ближайшего будущего. В мессенджерах должны появиться магазины ботов, и в выигрыше будет тот, кто окажется в рядах первопроходцев.

Боты вряд ли убьют рынок приложений, как обещают громкие заголовки, однако наверняка отберут большую долю у индустрии колл-центров. SonyPictures уже заменила 70 сотрудников на один чат-бот. Перераспределение трафика в пользу ботов ожидаемо в областях электронной коммерции и информационных сервисов.

Кроме того, интригу оставляет за собой активное развитие технологий искусственного интеллекта и обработки языка: сеть может существенно измениться, если чат-боты научатся делать что-то, чего пока не умеют сайты и приложения.

**Чат-бот** — это программа, работающая внутри социальной сети или другого приложения. Такая программа способна отвечать на вопросы а также самостоятельно задавать их. Чат-боты используются в разных сферах для решения типовых задач.

По прогнозам BusinessInsider, уже к 2020 году 80 % компаний будет пользоваться чат-ботами.

Чат-боты могут использоваться в службах поддержки, помогая решить простые вопросы, например, такие как смена пароля.

Чат-боты можно использовать для поиска информации. Например, прогноз погоды, афиша мероприятий. В мессенджере Telegram можно найти десятки тысяч ботов, способных рассказать о погоде или помочь выбрать подарок.

Чат-боты помогают работодателям и соискателям в процессе поиска работы и подбора кадров. Летом 2016 года агентство по поиску работы FirstJob выпустило чат-бота Мия, который не только предлагает подходящие вакансии, но и проводит с кандидатом на вакансию собеседование. В России собеседования проводил чат-бот от компании Superjob для компании «Связной».

**Для чего нужен чат-бот?**

Они нужны там, где есть много похожих действий, которые выполняются руками. Чат-бота можно использовать для поиска информации. Например, прогноз погоды, афиша мероприятий.

Сфера путешествий стали одними из самых первых использовать ботов. Бот может предложить направления/рейсы/рестораны — на основании поисковых запросов и предпочтений пользователя. После покупки программа обеспечивает клиентскую поддержку, предоставляя ответы на часто задаваемые вопросы

Бот будет полезен, если Вы хотите:

* обрабатывать много однотипных обращений;
* организовывать рассылки для подписчиков;
* фильтровать заявки в системах поддержки;
* мгновенно реагировать на сообщения.

Бот может стать отличным средством, чтобы привлечь внимания к своему сообществу, создавая интерактивные квесты и игровые сценарии. Пользователи не пройдут мимо интересных предложений!

Это самые распространенные сценарии. Боты для любой тематики. Можно придумать собственную реализацию, которая подошла бы для развития именно Вашей организации.

Создать чат-бот можно с помощью готовых конструкторов, а можно с помощью языков программирования, то есть создать "с нуля"

Одним из таких языков является Python

# Чат-бот с нуля

## Язык программирования Python

Язык программирования Python 3 — это современный и мощный инструмент для создания программ самого разнообразного назначения. С его помощью можно решать задачи различных типов.

Python обладает огромный функционалом для выполнения различных целей в том числе и создания чат-ботов.

Он пригоден для решения разнообразных задач и предлагает те же возможности, что и другие языки программирования: динамичность, поддержку ООП и кросс-платформенность. Разработку Python начал Гвидо Ван Россум (GuidoVanRossum) еще в середине 1990-х годов, поэтому к настоящему времени удалось избавиться от стандартных «детских» болезней, существенно развить лучшие стороны языка и привлечь множество программистов, использующих Python для реализации своих проектов.

Многие программисты считают, что необходимо изучать только «классические» языки программирования, такие как Java или C++, так как другие языки все равно не смогут обеспечить таких же возможностей. Однако в последнее время возникло убеждение, что программисту желательно знать более одного языка, так как это расширяет его кругозор, позволяя более творчески решать поставленные задачи и повышая его конкурентоспособность на рынке труда.

Изучить в совершенстве два таких языка как Java и C++ достаточно сложно и заняло бы много времени; кроме того, многие аспекты этих языков противоречат друг другу. В то же время Python идеально подходит на роль второго языка, так как он сразу же усваивается благодаря уже имеющимся знаниям в ООП, и тому, что его возможности не конфликтуют, а дополняют опыт, накопленный при работе с другим языком программирования.

Если же программист только начинает свой путь в области разработки ПО, то Python станет идеальным «вводным» языком программирования. Благодаря своей лаконичности он позволит быстрее овладеть синтаксисом языка, а отсутствие «наследства» в виде формировавшихся на протяжении многих лет аксиом поможет быстро освоить ООП. В силу этих факторов «кривая обучения» Python будет довольно короткой, и программист сможет перейти от учебных примеров к коммерческим проектам.

Из огромного множества языков программирования я выбрал именно Python. Разберемся почему.

**Архитектура Python**

Любой язык, неважно – для программирования или общения, состоит как минимум из двух частей – словаря и синтаксиса. Язык Python организован точно так же, предоставляя синтаксис для формирования выражений, образующих исполняемые программы, и словарь – набор функциональности в виде стандартной библиотеки и подключаемых модулей.

Как уже упоминалось, синтаксис Python достаточно лаконичный, особенно если сравнивать с Java или C++. С одной стороны – это хорошо, так как чем проще синтаксис, тем проще его изучить и тем меньше ошибок можно совершить в процессе его использования. Однако у подобных языков есть недостаток – с их помощью можно передавать самую простую информацию и нельзя выражать сложные конструкции.

К Python это не относится, так как это язык простой, но упрощенный. Дело в том, что Python является языком с более высоким уровнем абстракции, выше, например, чем у Java и C++, и позволяет передать такое же количество информации в меньшем объеме исходного кода.

Также Python является языком общего назначения, поэтому может применяться практически в любой области разработки ПО (standalone, клиент-сервер, Web-приложения) и в любой предметной области. Кроме того, Python легко интегрируется с уже существующими компонентами, что позволяет внедрять Python в уже написанные приложения.

Другая составляющая успеха Python – это его модули расширения, как стандартные, так и специфические. Стандартные модули расширения Python – это отлично спроектированная и неоднократно проверенная функциональность для решения задач, возникающих в каждом проекте по разработке ПО, обработка строк и текстов, взаимодействие с операционной системой, поддержка Web-приложений. Эти модули также написаны на языке Python, поэтому обладают его важнейшим свойством – кросс-платформенностью, позволяющей безболезненно и быстро переносить проекты с одной операционной системы на другую.

Если необходимой функциональности не оказалось в стандартной библиотеке Python, то можно создать собственный модуль расширения для его последующего неоднократного использования. Здесь стоит отметить, что модули расширения для Python можно создавать не только на самом языке Python, но и с помощью других языков программирования. В этом случае появляется возможность более эффективной реализации ресурсоемких задач, например сложных научных вычислений, однако теряется преимущество кросс-платформенности, если язык модуля расширения не является сам по себе кросс-платформенным, как Python.

**Основные преимущества Python**

1. Динамическая типизация
2. Поддержка модульности
3. Одна из самых понятных реализаций объектно-ориентированного программирования
4. Сборка мусора, отсутствие утечек памяти
5. Интеграция с другими языками программирования
6. Удобный и лаконичный синтаксис
7. Огромная база модулей и возможность добавлять в нее свои модули для расширения функционала
8. Кроссплатформенность. Pythonбудет отлично себя чувствовать на других операционных системах
9. Низкий порог вхождения в изучение языка

## База готовых команд VKAPI

Кроме языка программирования нам понадобиться инструменты, предоставляемые самим сайтом или приложением. Таким инструментом является – VKAPI(набор готовых классов, процедур, функций, структур и констант, предоставляемых социальной сетью)

**API**

Что такое API? Аббревиатура расшифровывается как ApplicationProgrammingInterface, или интерфейс для программирования приложений. Название, вроде бы, говорит само за себя, но лучше рассмотреть более детальное объяснение.

Как уже было сказано, API – это, в первую очередь, интерфейс. Интерфейс, который позволяет разработчикам использовать готовые блоки для построения приложения. В случае с разработкой мобильных приложений в роли API может выступать библиотека для работы с "умным домом" – все нюансы реализованы в библиотеке и вы лишь обращаетесь к этому API в своём коде.

В случае веб-приложений, API может отдавать данные в отличном от стандартного HTML формате, благодаря чему им удобно пользоваться при написании собственных приложений. Сторонние общедоступные API чаще всего отдают данные в одном из двух форматов: XML или JSON. На случай, если вы решили сделать API для своего приложения, запомните, что JSON намного более лаконичен и прост в чтении, чем XML, а сервисы, предоставляющие доступ к данным в XML-формате, постепенно отказываются от последнего.

**API как средство интеграции приложений**

API определяет функциональность, которую предоставляет программа (модуль, библиотека), при этом API позволяет абстрагироваться от того, как именно эта функциональность реализована.

Если программу (модуль, библиотеку) рассматривать как чёрный ящик, то API — это множество «ручек», которые доступны пользователю данного ящика и которые он может вертеть и дёргать.

Программные компоненты взаимодействуют друг с другом посредством API. При этом обычно компоненты образуют иерархию — высокоуровневые компоненты используют API низкоуровневых компонентов, а те, в свою очередь, используют API ещё более низкоуровневых компонентов.

По такому принципу построены протоколы передачи данных по Интернет. Стандартный стек протоколов (сетевая модель OSI) содержит 7 уровней (от физического уровня передачи бит до уровня протоколов приложений, подобных протоколам HTTP и IMAP). Каждый уровень пользуется функциональностью предыдущего («нижележащего») уровня передачи данных и, в свою очередь, предоставляет нужную функциональность следующему («вышележащему») уровню.

Важно заметить, что понятие протокола близко по смыслу к понятию API. И то, и другое является абстракцией функциональности, только в первом случае речь идёт о передаче данных, а во втором — о взаимодействии приложений.

API библиотеки функций и классов включает в себя описание сигнатур и семантики функций.

**Сигнатура функции** — часть общего объявления функции, позволяющая средствам трансляции идентифицировать функцию среди других. В различных языках программирования существуют разные представления о сигнатуре функции, что также тесно связано с возможностями перегрузки функций в этих языках.

Иногда различают сигнатуру вызова и сигнатуру реализации функции. Сигнатура вызова обычно составляется по синтаксической конструкции вызова функции с учётом сигнатуры области видимости данной функции, имени функции, последовательности фактических типов аргументов в вызове и типе результата. В сигнатуре реализации обычно участвуют некоторые элементы из синтаксической конструкции объявления функции: спецификатор области видимости функции, её имя и последовательность формальных типов аргументов.

Например, в языке программирования C++ простая функция однозначно опознаётся компилятором по её имени и последовательности типов её аргументов, что составляет сигнатуру функции в этом языке. Если функция является методом некоторого класса, то в сигнатуре будет участвовать и имя класса.

В языке программирования Java сигнатуру метода составляет его имя и последовательность типов параметров; тип возвращаемого значения в сигнатуре не участвует.

**Семантика функции** — это описание того, что данная функция делает. Семантика функции включает в себя описание того, что является результатом вычисления функции, как и от чего этот результат зависит. Обычно результат выполнения зависит только от значений аргументов функции, но в некоторых модулях есть понятие состояния. Тогда результат функции может зависеть от этого состояния, и, кроме того, результатом может стать изменение состояния. Логика этих зависимостей и изменений относится к семантике функции. Полным описанием семантики функций является исполняемый код функции или математическое определение функции.

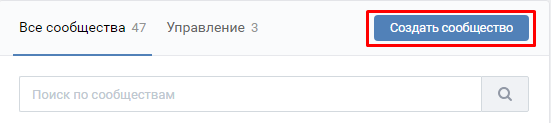
**VKApi**

API ВКонтакте — это интерфейс предоставляемый ВКонтакте, который позволяет получать информацию из базы данных vk.com с помощью http-запросов к специальному серверу.

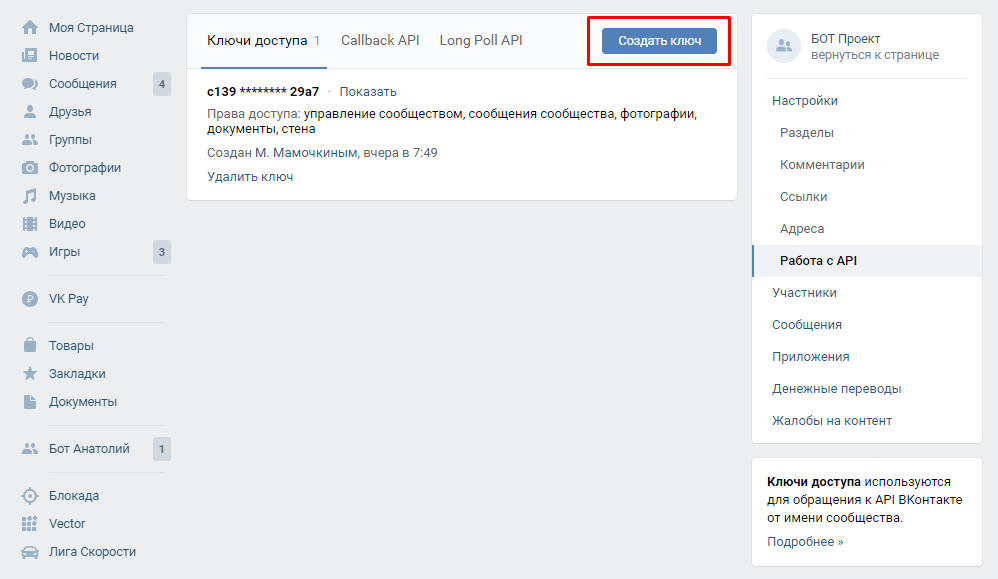
Благодаря ApiVkнам не нужно знать в подробностях, как устроена база, из каких таблиц и полей каких типов она состоит — достаточно того, что API-запрос об этом «знает». Синтаксис запросов и тип возвращаемых ими данных строго определены на стороне самого сервиса.

# Создание бота на языке Python

## Создаем группу ВК

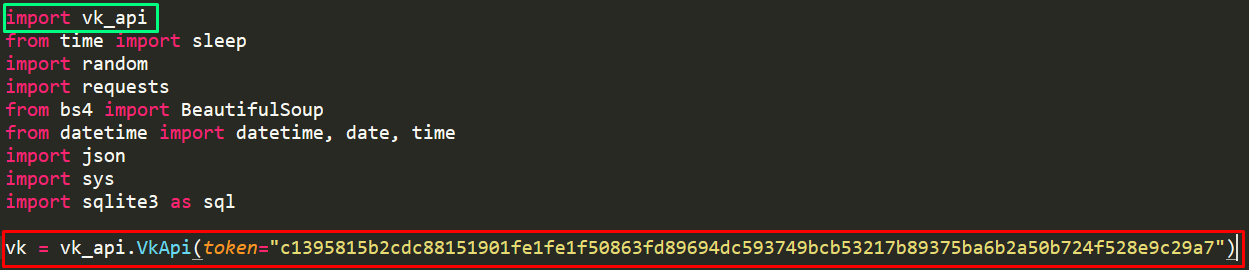
Для начала необходимо создать сообщество, в котором будет “жить” наш чат-бот. В социальной сети Вконтакте кликаем по кнопке “создать сообщество”.

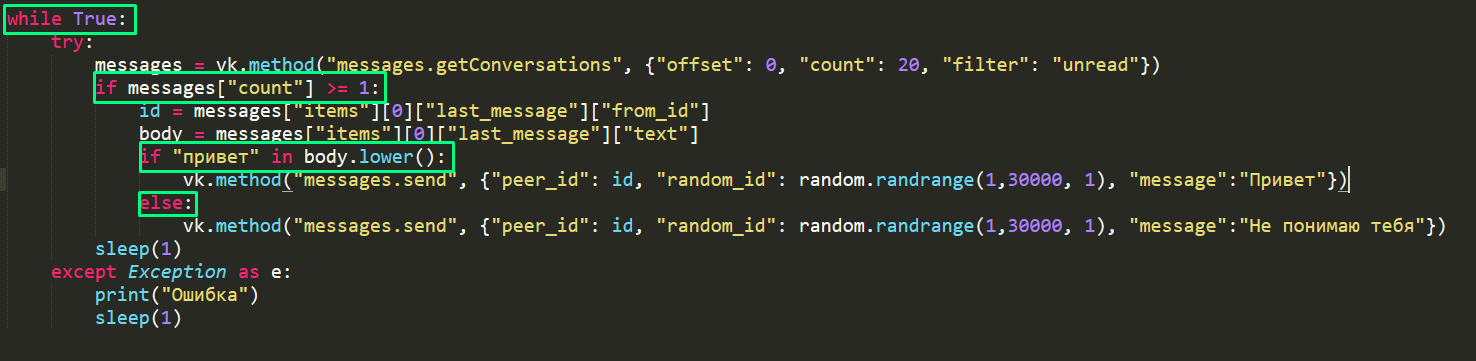
Следующим действием нужно будет настроить бота для работы с сообщения, которые мы будем ему посылать. Переходим во вкладку “Управление” –> “Работа с API” -> “Создать ключ”. Получившийся ключ сохраняем



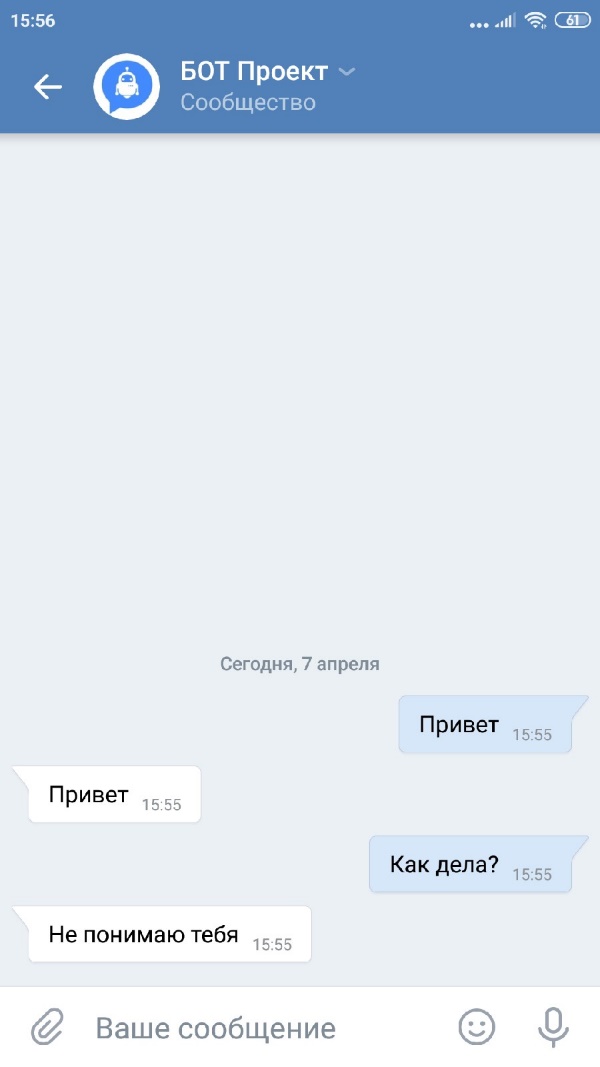
## Начинаем писать код

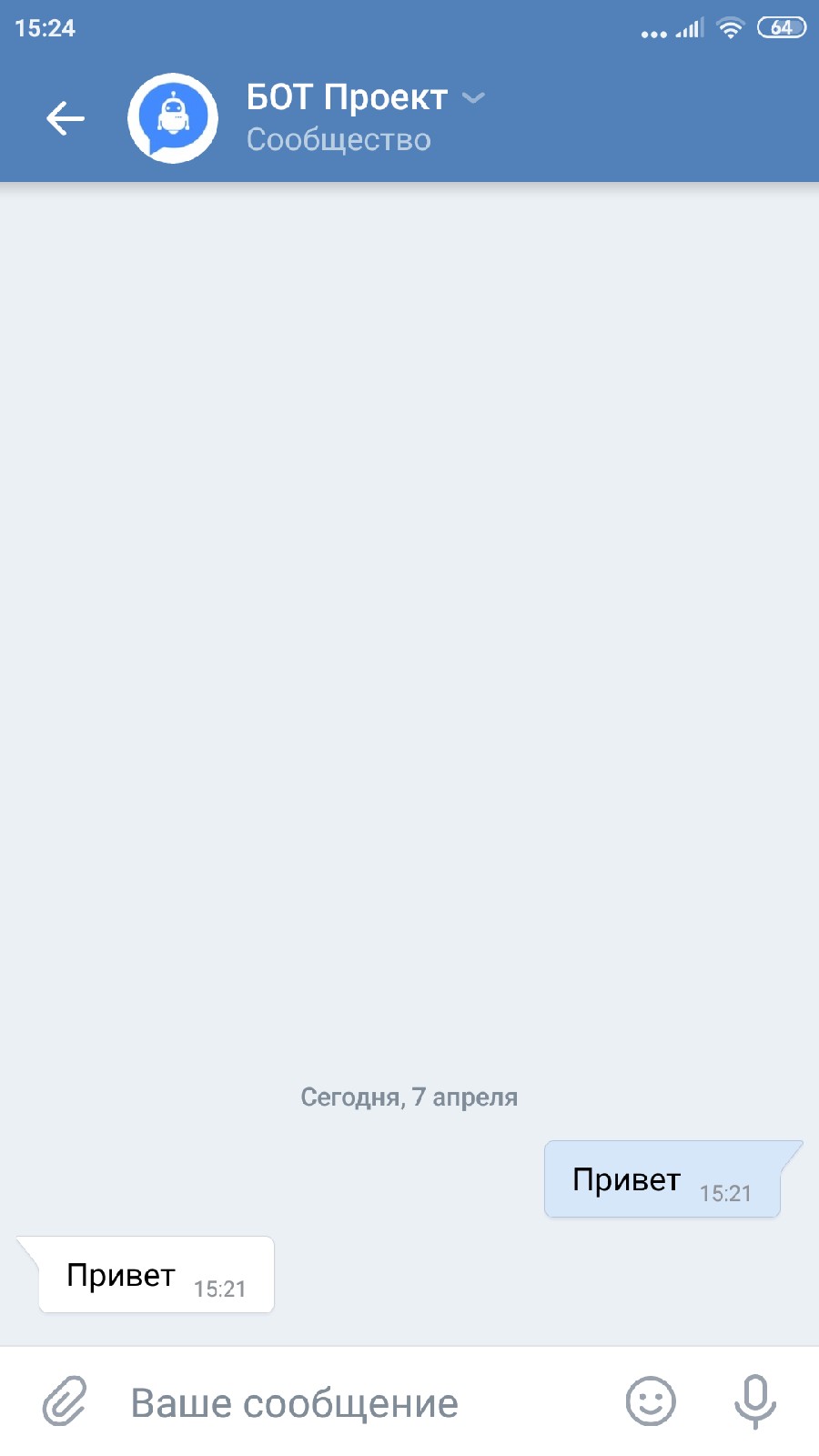
***Базовые функции***

Импортируем библиотеку vk\_api, а также остальные модули для работы с нашим ботом. Записываем наш ключ через функцию в переменную vk

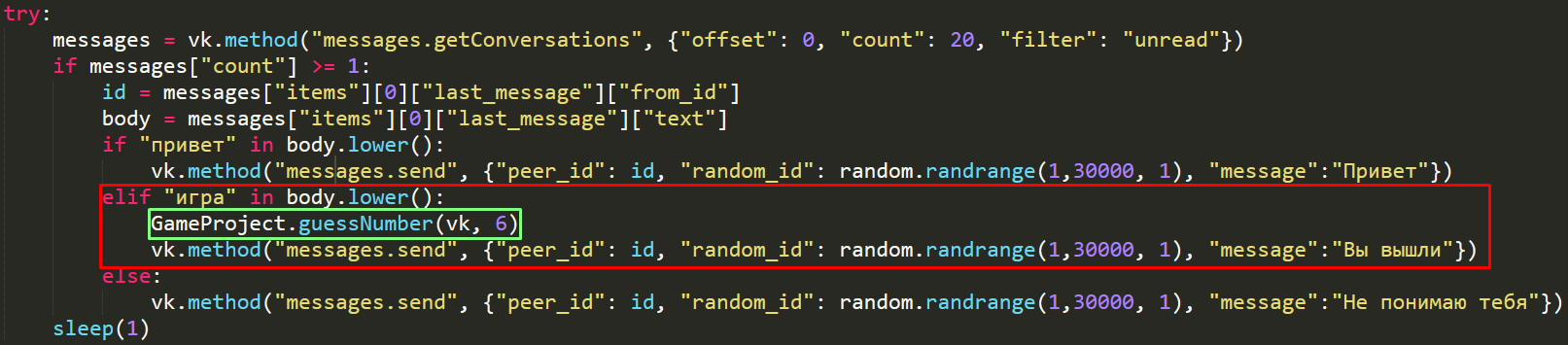
Создаем главный цикл, в котором будет находиться наш обработчик сообщений. Проверяем на наличие новых сообщений. Исходя из команды, выполняем нужное действие.

В данном примере показано, что при отправке боту сообщения “привет” он ответит: “Привет”. Если же команда иная, выведет: “Не понимаю тебя”.

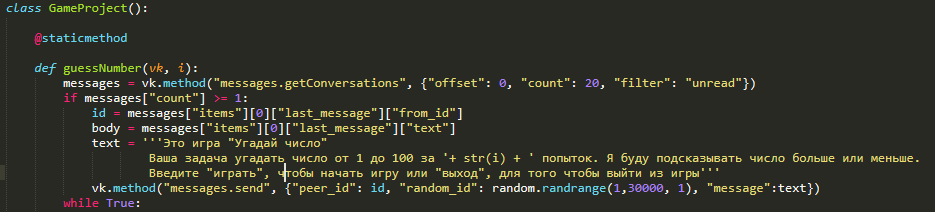


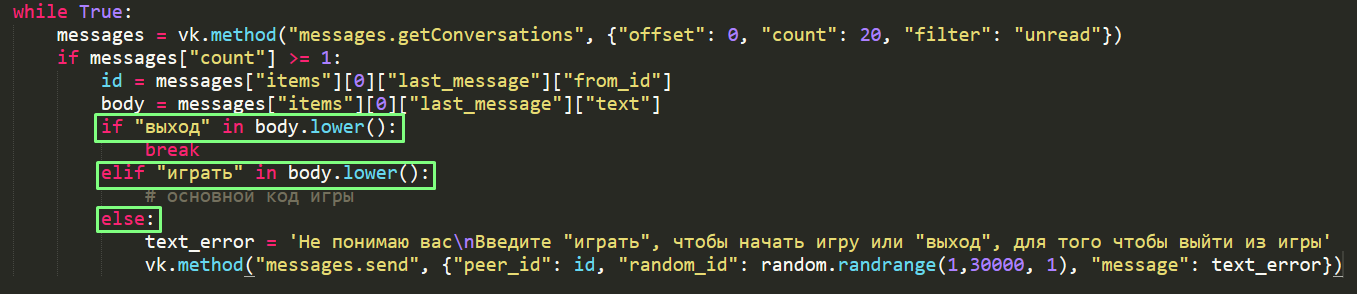


***Пишем код для игры “Угадай число”***

Для создания игры “Угадай число” нам потребуется в обработчике сообщений прописать проверку на команду “игра”. Код я буду писать в отдельном статическом методе guessNumber класса GameProject.Функция guessNumber() принимает два значения – ответ от сервера и число попыток, за которое нужно угадать число (n).

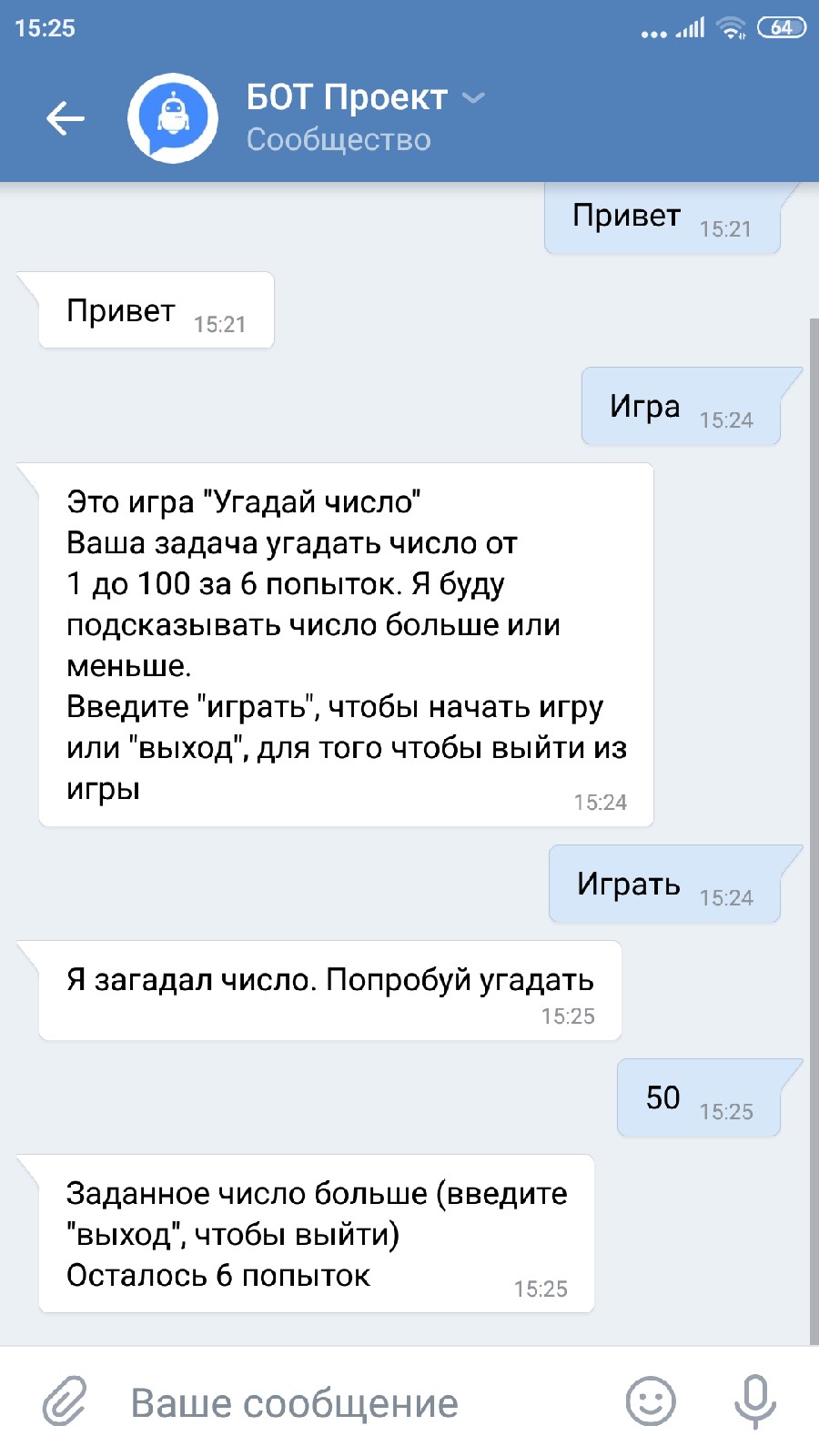
В методе выводим небольшую инструкцию пользователю.Задача угадать число от 1 до 100 за n попыток. Бот будет подсказывать число больше или меньше. Написать "играть", чтобы начать игру или "выход", для того чтобы выйти из игры.

Отправляем бота в цикл.

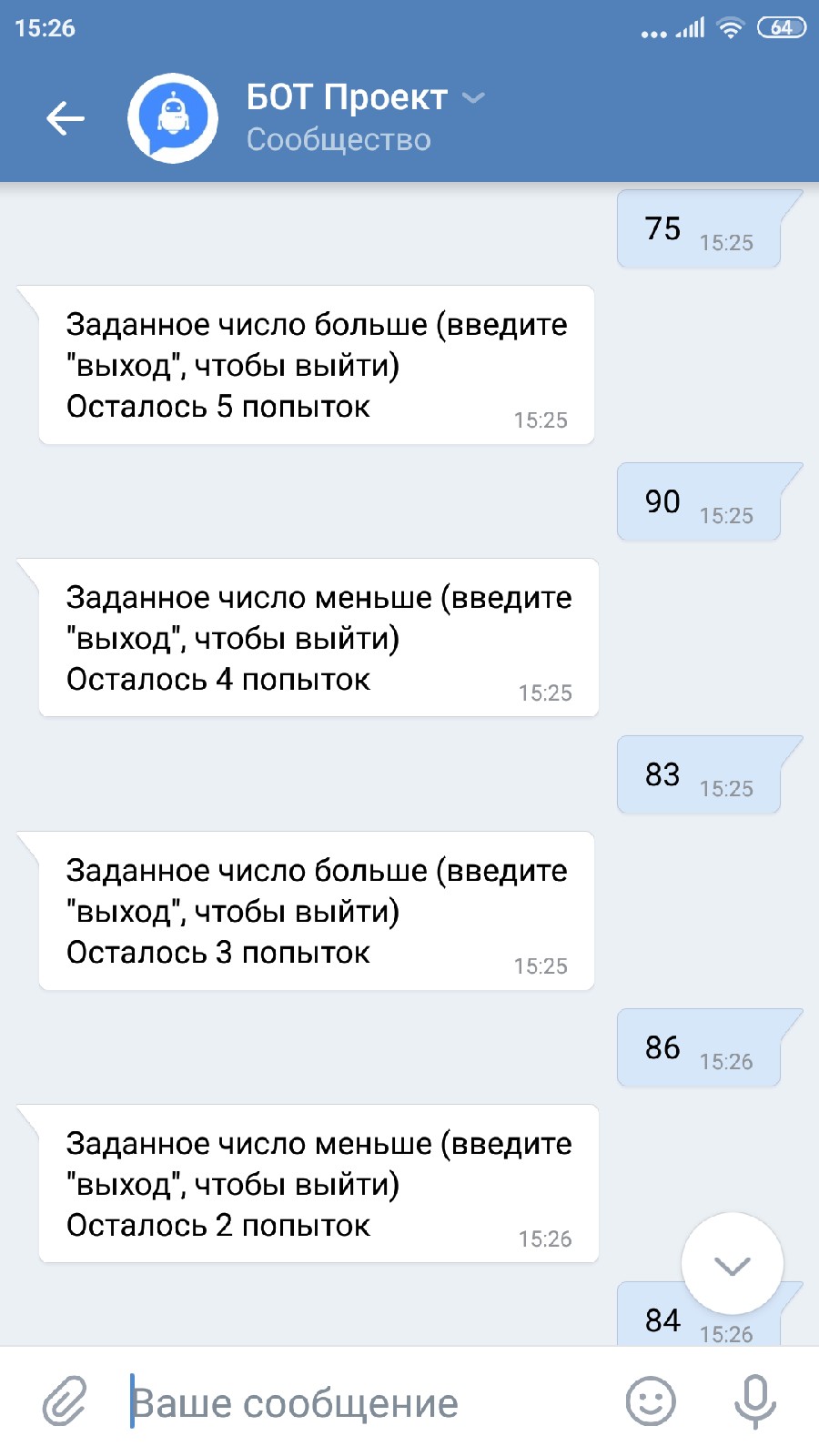
Прописываем условия. Если была введена команда “выход”, выходим из цикла и приводим бота в прежнее состояние. Если команда не известная – выводим ошибку.

Снова проверяем на наличие сообщения. Создаем случайное число от 1 до 100 и записываем в переменную num\_r. Отправляем в еще один цикл. Проверяем значение введенное пользователем на число. Если это не число – выводим ошибку, иначе делаем проверку. Если пользователь угадал значение num\_r – выводим поздравление и выходим из цикла, если число больше или меньше и находится в диапазоне от 1 до 100 – выводим соответствующее сообщение.

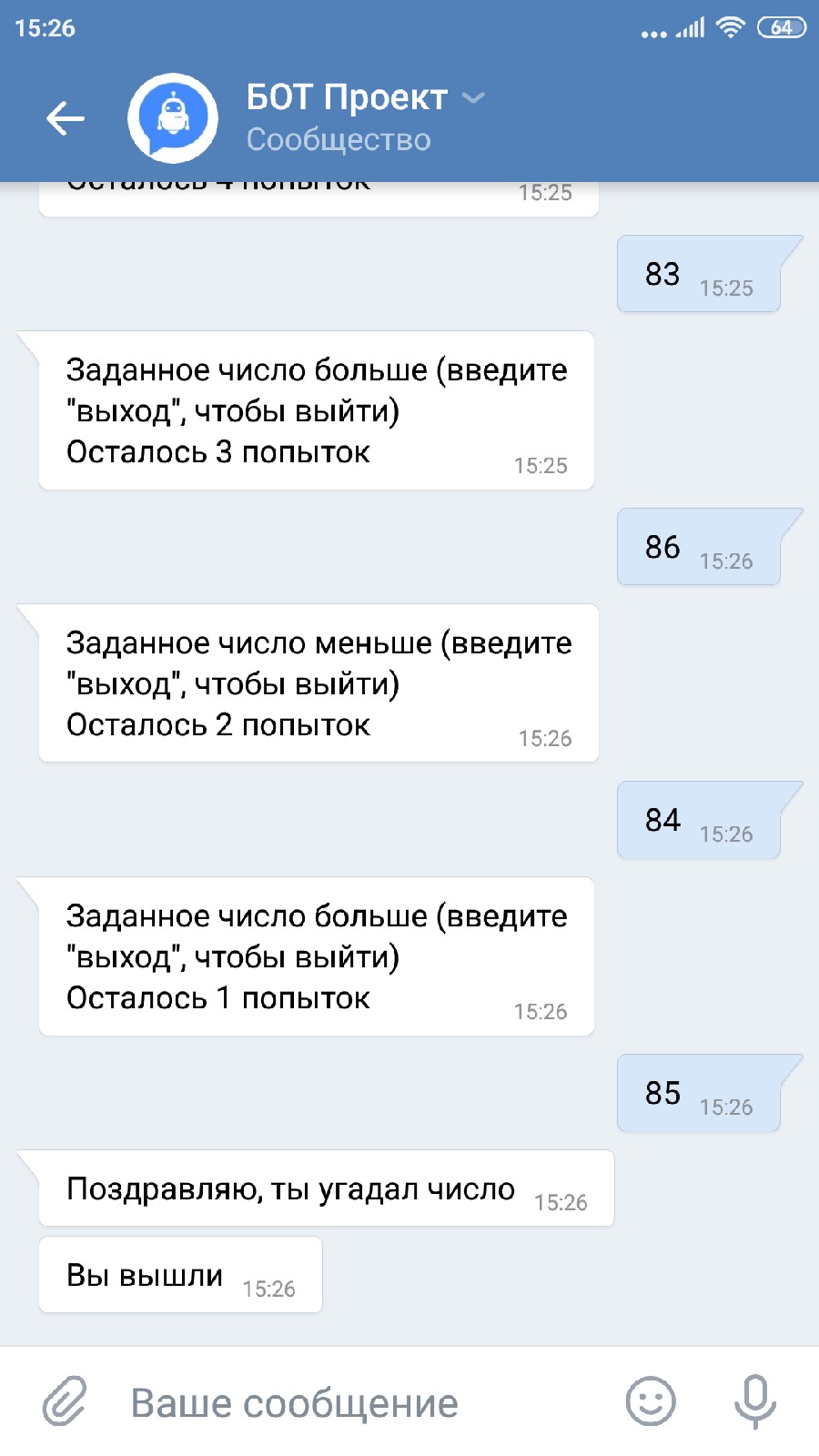
Принцып работы игры “Угадай число”



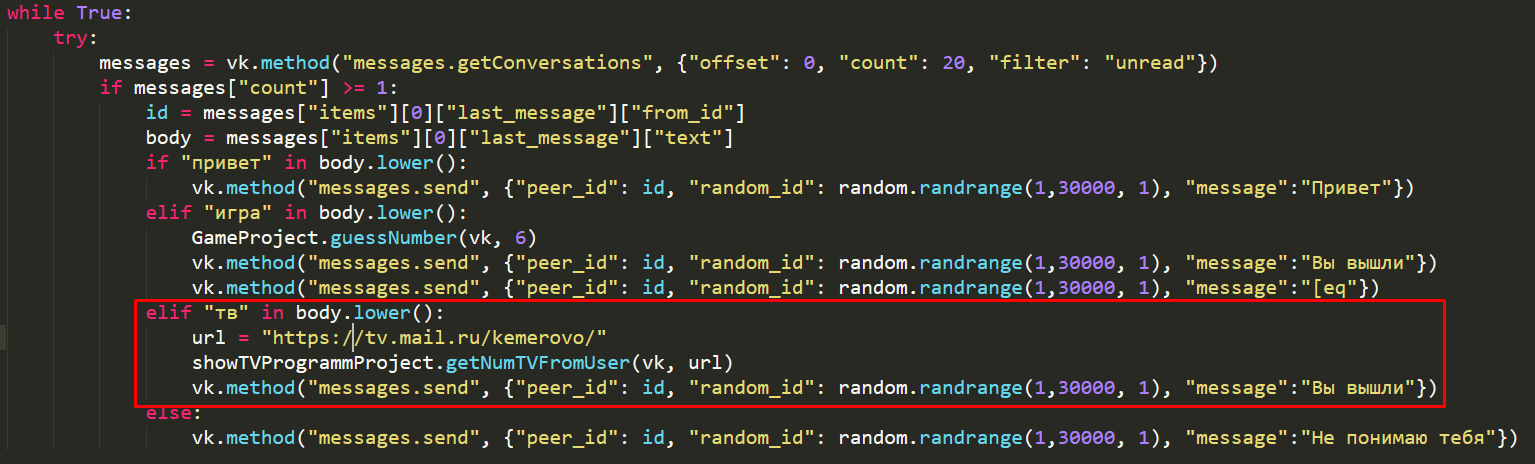
Как мы видим, количество попыток с каждым ходом уменьшается.



Если пользователь угадал число выводим: “Поздраляю, ты угадал число” и выходим из игры.



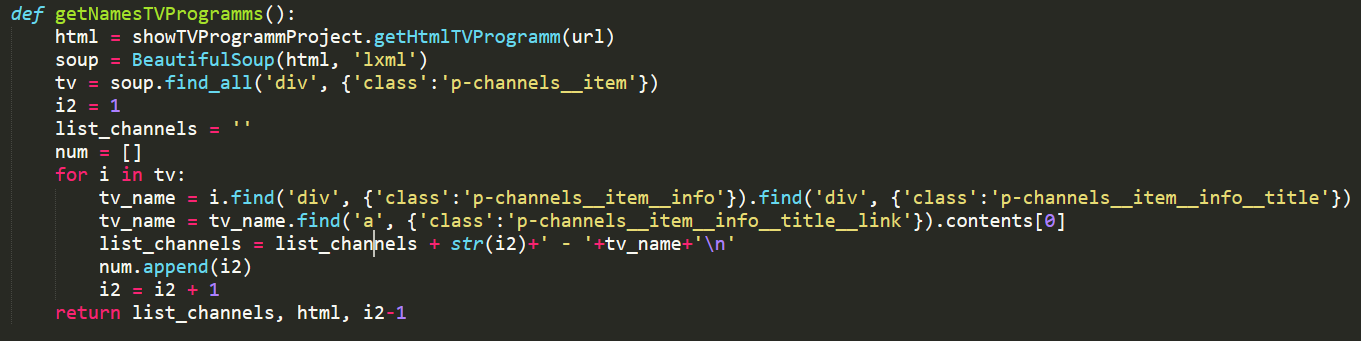
***Пишем код для выдачи телепрограммы на сегодняшний день***

 Добавляем в наш главный обработчик команду “тв”. В переменную urlзаписываем сайт с которого будем брать телепрограмму. В данном примере я беру информацию с сайта TVMail.Обращаемся кметоду getNumTVFromUser класса showTVProgrammProject, который принимает два значения – ответ от сервера и urlстраницы.

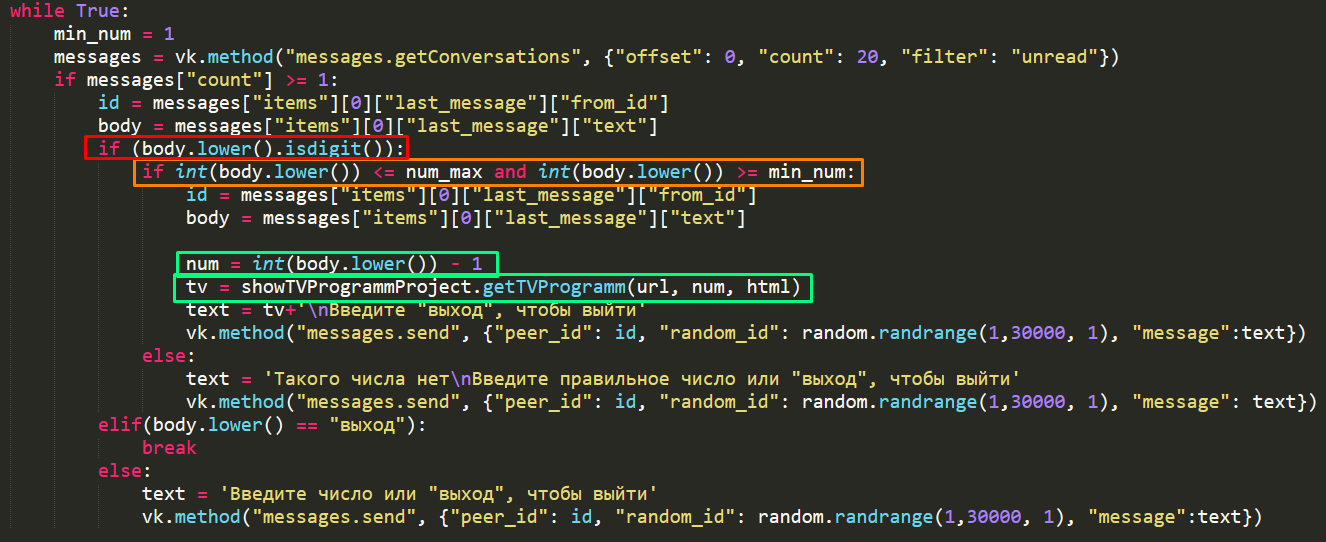
В методе обращаемся к функции getNamesTVProgramms того же класса. Функция возвращает название всех телепрограмм, htmlкод, а также количество каналов. И входим в цикл.



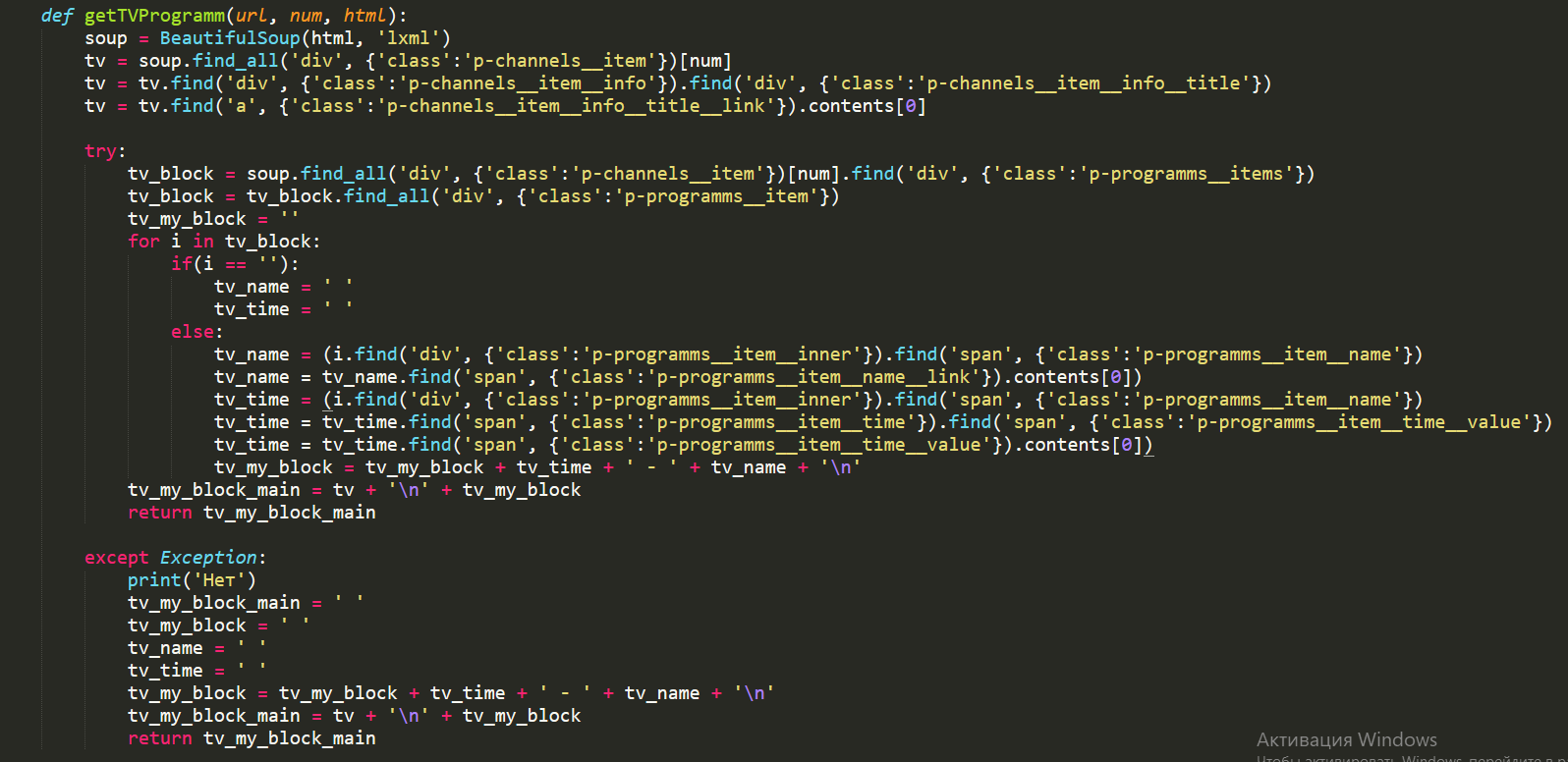
Исходный код функции getNamesTVProgramms (здесь применяем инструмент BeautifulSoupдля изъятия элементов из страницы):



Проверяем сообщение на число и вход его в диапазон всех телеканалов. Если все в порядке – запоминаем число в переменной numи обращаемся к новой функции getTVProgramm, которая принимает три параметра. Функция возвращаети отправляет расписание выбранного телеканала.

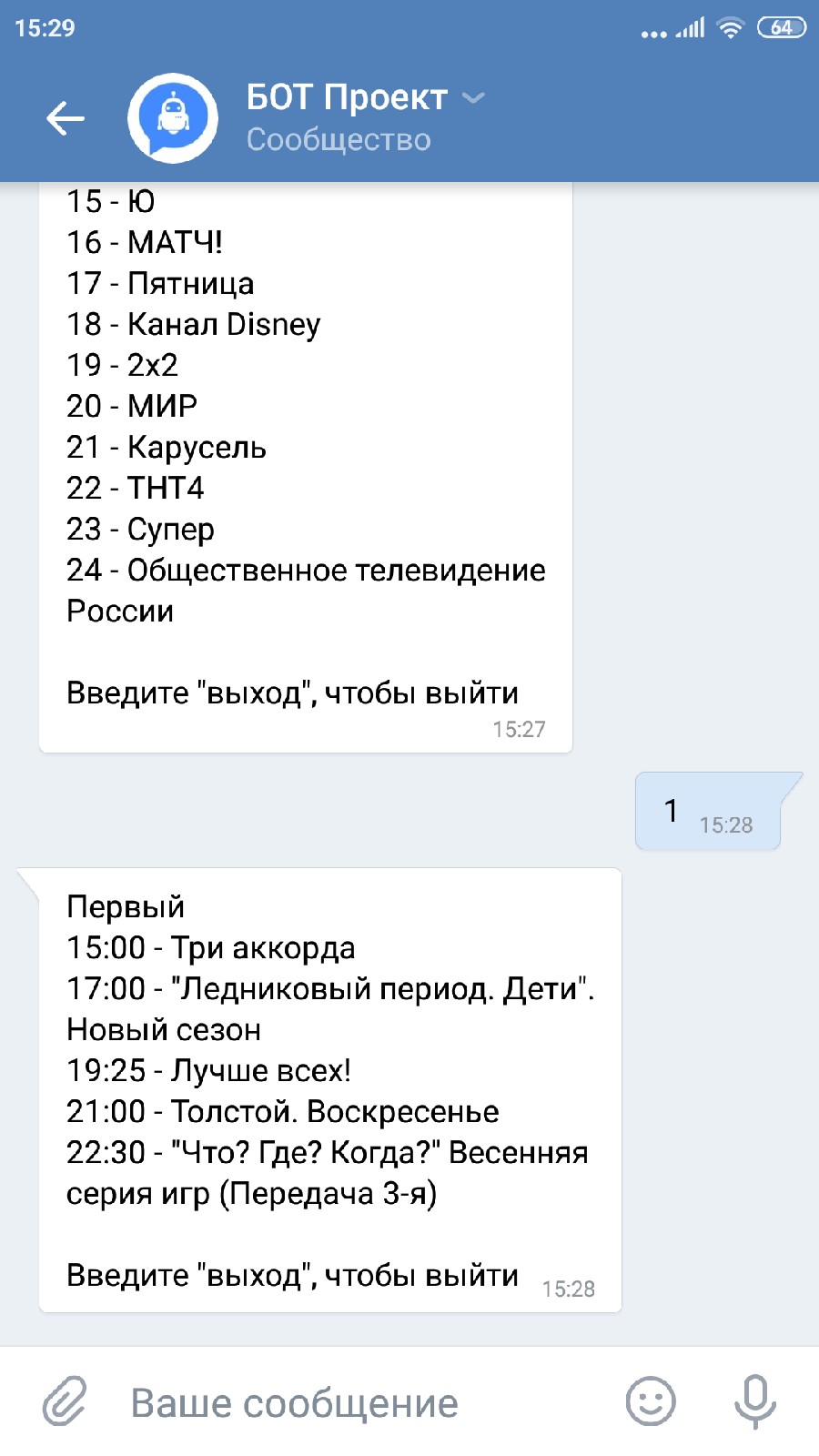


Исходный код метода getTVProgramm(применяем инструмент BeautifulSoupдля изъятия элементов из страницы):

 Проверим работу выдачи телепрограммы.



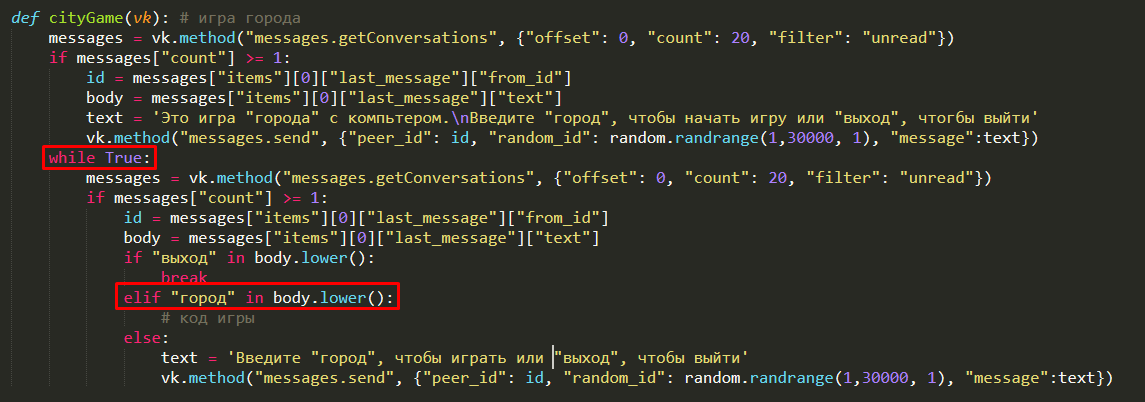
Выведим например телепрограмму первого канала. Для этого отправляем сообщение с номером “1”.

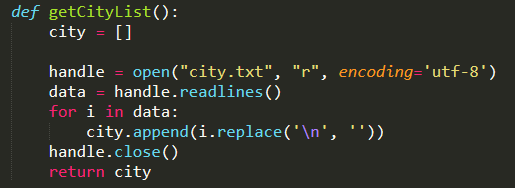


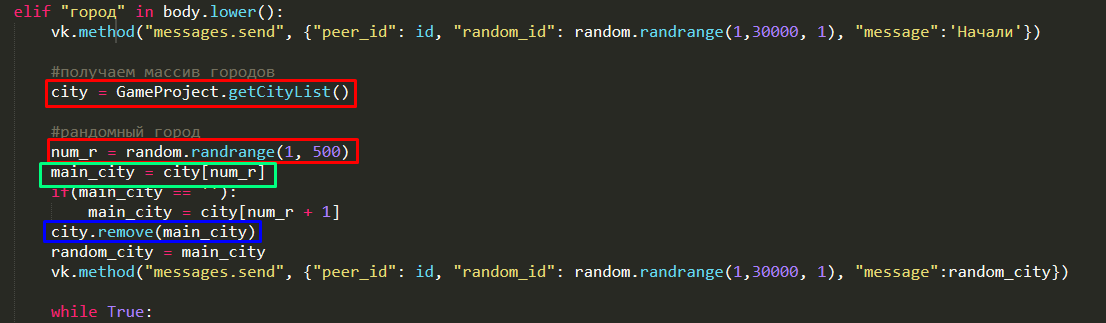
***Создание игры “Города”***

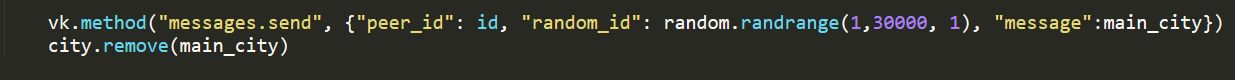
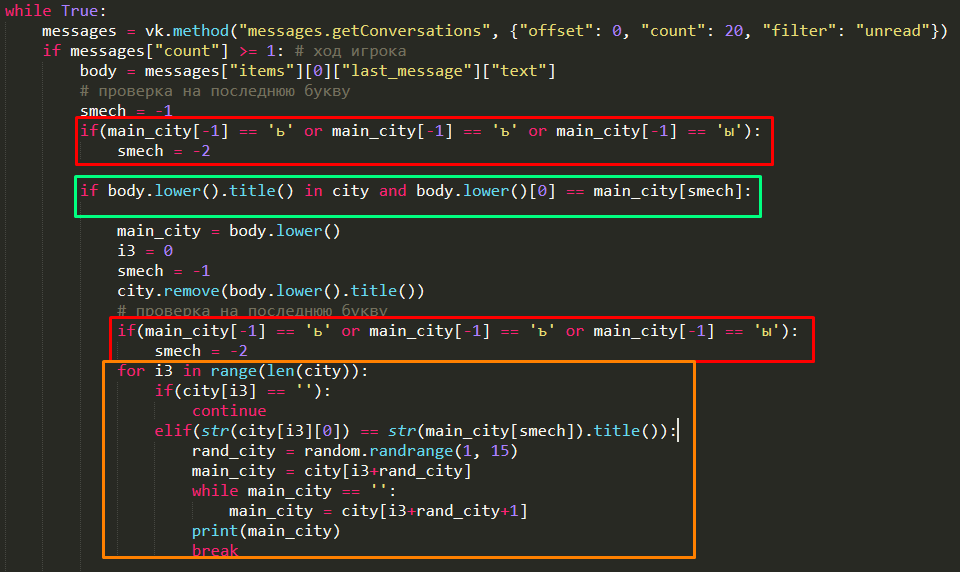
Добавляем в обработчик сообщений команду “города”. Обращаемся к методу cityGameкласса GameProject, который принимает одно значение – ответ от сервера.

Выводим пользователю небольшую инструкцию и отправляемся в цикл. Далее проверяем на наличие команд “города” или “выход”.

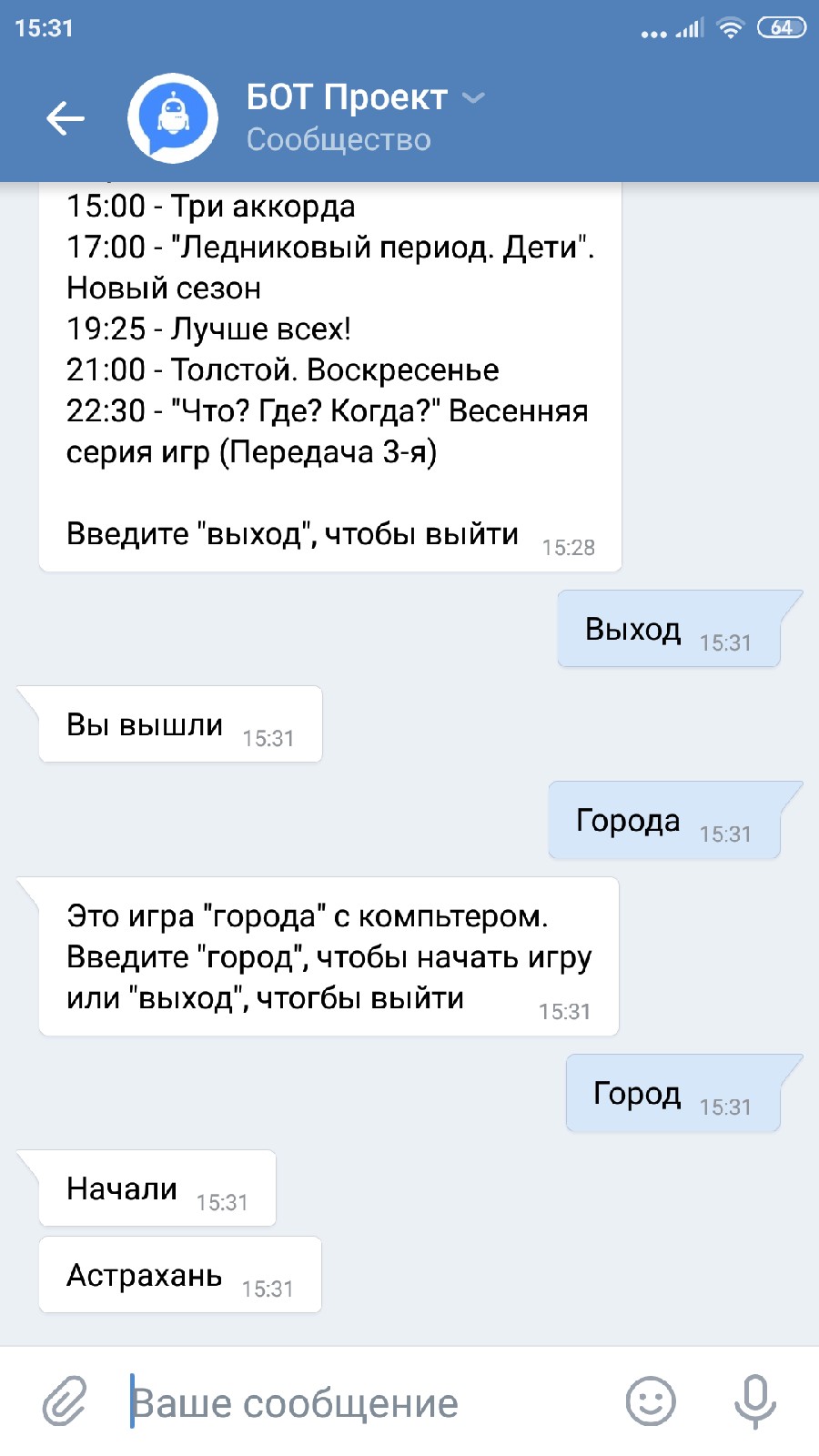
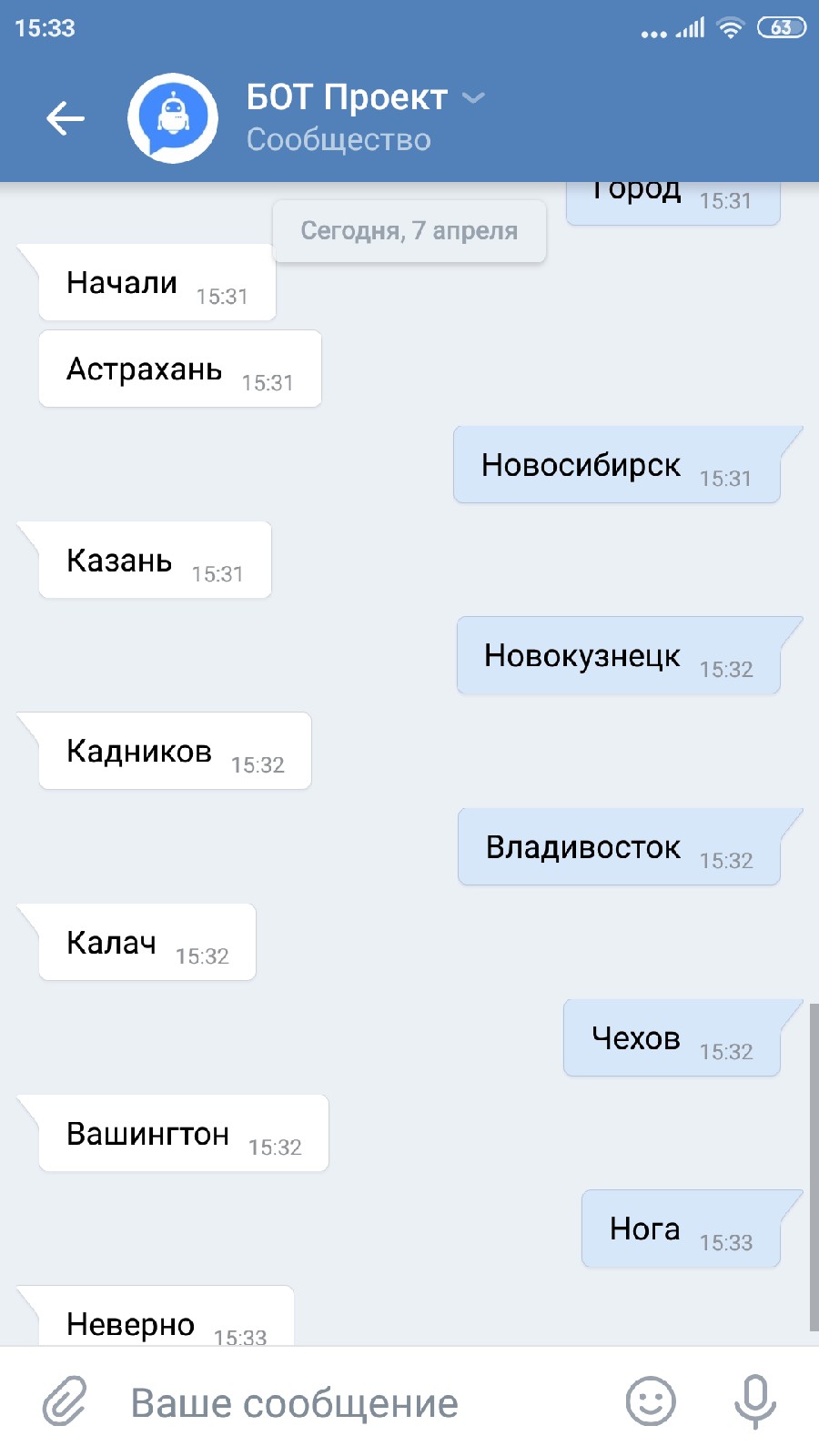


Первым делом нужно достать базу данных названия городов.В Интернете я нашел список городов (~1500) и записал в текстовый блокнот под именем city. Создадим функцию getCityList, которая будет возвращать массив с городами из текстового блокнота.

Обращаемся из метода cityGameк функции getCityListи получаем массив с городами. Начинает компьютер, поэтому из списка выбираем случайный город. Удаляем этот город из списка, чтобы компьютер не использовал его повторно. И переходим в цикл.

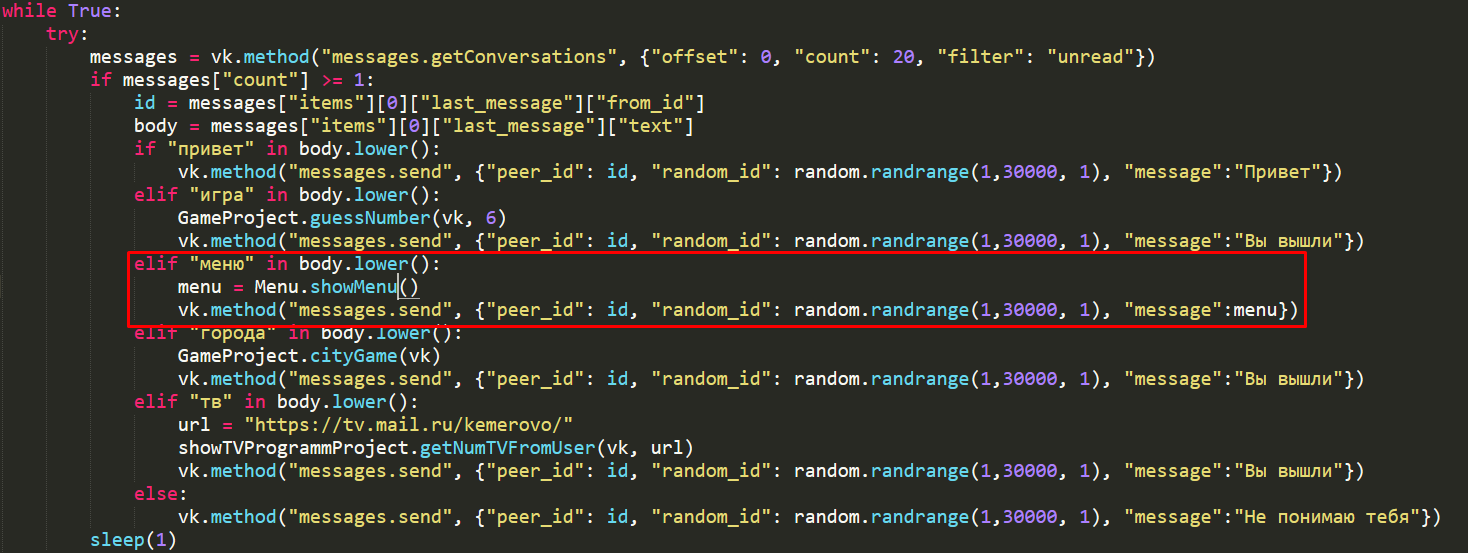
Делаем проверку на последнюю букву города, который записал пользователь. Если это буквы ”ъ”,”ь” или “ы”, то берем букву, стоящую перед последней. Проверяем наличие такого города с списке, если существует – удаляем и отправляем новый город.

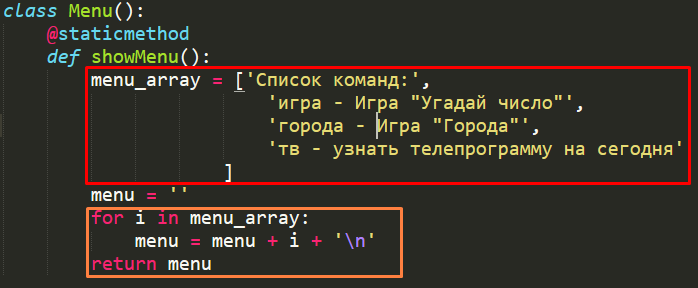
Прописываем слово “город” и начинаем игру. Как можно заметить, город выдал город, у которого последняя буква “ь”. Берем букву перед последней и пишем на нее город.



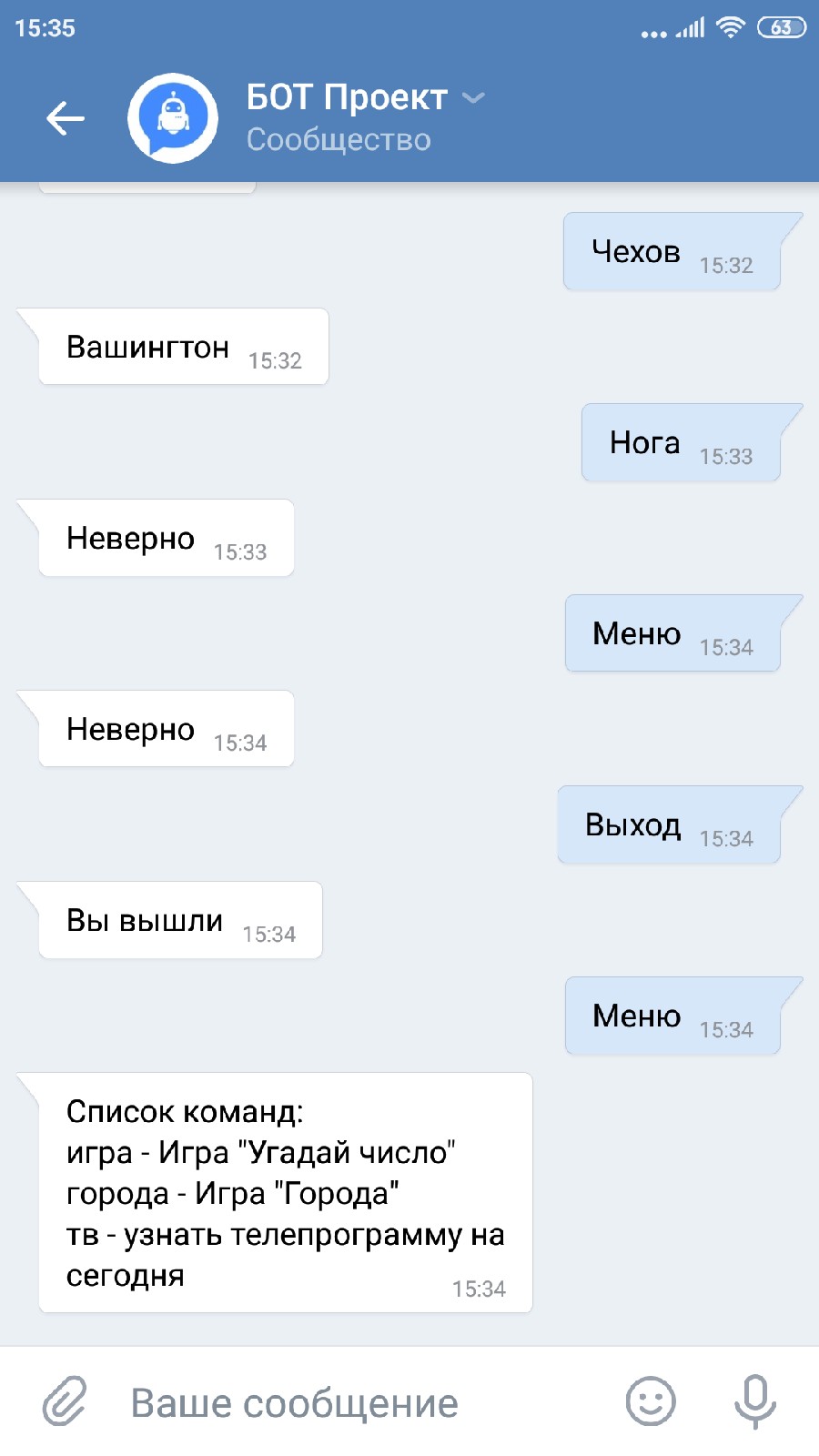
Если такого города нет, бот выдаст ошибку.

***Делаем помощь по командам***

Чтобы помочь пользователю узнать, какие функции существуют мы создадим помощь по командам. По слову “меню” бот будет выдавать список действий, которые может выполнить бот. Прописываем в условии команду и вызываем метод showMenuкласса Menu, который будет возвращать список возможных взаимодействий с ботом.

 Функция содержит массив, в котором хранятся названия команд и их описание. Далее преобразовываем массив в строку и возвращаем результат.

Принцип работы команды “меню”:



## Заключение

В ходе создания чат-бота с нуля, я изучил историю создания искусственного интеллекта, выбрал инструменты для создания чат-ботов, углубил знания в языке программирования Python и написал код для чат-бота Вконтакте. Я выяснил, что таким способом можно создать полноценного многофункционального чат-бота, который можно использовать в самых разных целях, не только для развлечений, но и как полноценного помощника при покупках, доставках и обслуживании клиентов. В дальнейшем планирую углублять свои знания в этой области и создавать других чат-ботов, которые будут полезны людям.

# Список литературы

1. Бхаргава А, Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих/[Текст], Адитья Бхаргава: С-Пбг: Питер, 2018
2. Читлур, С “Укус Питона”, Сваруп/[Текст], Читлур , М.: Канто, 2013
3. Бот [Электронный ресурс] , Википедия URL: https://ru.wikipedia.org/wiki/Бот\_(программа) (дата обращения 03.04.2019)
4. Создание чат-бота в вк [Электронный ресурс]/ URL: https://habr.com/ru/post/427691/ (дата обращения 03.04.2019)
5. Создание чат бота в вк [Электронный ресурс] /https://www.youtube.com/watch?v=wqWb9\_QdE\_g (дата обращения 06.03.2019)