Кенчошвили Вадим Викторович
студент
ФГБОУ ВО «Государственный университет морского
и речного флота имени адмирала С.О. Макарова»
г. Санкт-Петербург

АВТОМАТИЗАЦИЯ РАБОЧЕГО ПРОЦЕССА ВЕДУЩЕГО ПРЯМЫХ ТРАНСЛЯЦИЙ НА САЙТЕ TWITCH.TV ПОСРЕДСТВОМ СОЗДАНИЯ БОТА ДЛЯ ЧАТА

***Аннотация**: статья посвящена созданию чат-бота для стриминговой площадки twitch.tv. Рассматриваются все элементы, из которых состоит бот и их взаимодействие друг с другом.*

***Ключевые слова**: бот, потоки, модуль, ведущий трансляции, чат.*

Введение. Каждый ведущий прямых трансляций стремится к максимизации полезности использования своего времени в процессе их проведения. В наше время для улучшений качества работы ведущих используют специальных ботов, которые снижают их рабочую нагрузку, тем самым упрощая работу и улучшая качество ее выполнения. Для автоматизации рабочего процесса необходимо разработать и установить бота на устройство, с которого ведущий проводит свои трансляции. Актуальность данной статьи объясняется тем, что, в настоящее время, ведущим прямых трансляций очень сложно общаться с большой аудиторией самим. Для решения этих проблем необходимо внедрение чат-бота, который будет выполнять часть его работы.

Чат-бот – это программа, работающая внутри мессенджера. Такая программа способна отвечать на вопросы, а также самостоятельно задавать их. Чат-боты используются в разных сферах для решения типовых задач.

Принцип работы Чат-бота (рис. 1):

1. Подключение к серверу, на котором ведется прямая трансляция;

2. Постоянная проверка чата на наличие в нем сообщений;

3. При наличии сообщений бот проверяет в своем справочнике совпадения по ключевым словам;

4. Если совпадения найдены, отправляет ответ на сообщение зрителя.

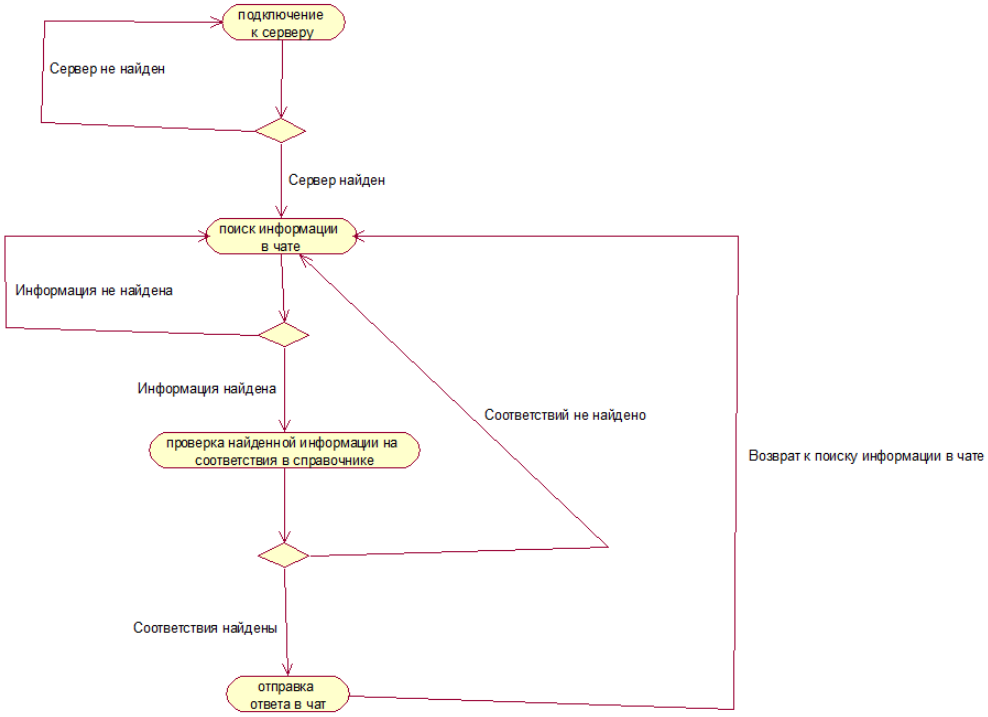

Рис. 1. Диаграмма деятельности

Составляющие части разрабатываемого бота

Данный бот состоит из трех основных файлов:

Первый файл является обычным конфигурационным файлом (обычно его называют config), в котором содержатся метаданные такие как: данные о главном узле, порт данного узла, наименование бота, пароль бота (пароль аутентификации для работы с ним, выдается главным узлом для связи с ним), в чате какого пользователя будет использоваться бот и интервал отправки сообщения данным ботом. Так же в нем можно создавать словари для хранения данных, например: о редакторах чата.

Второй файл (к примеру пусть будет называться utils) отвечает за API бота, т.е. за его функционал. В него импортированы библиотеки и модули, такие как:

– config – сам файл, который был создан до этого; он необходим для работы остальных библиотек и для связи с сервером;

– urllib2 – данная библиотека предназначена для работы с URL. В нашем боте эта библиотека необходима для того, чтобы подключиться к URL необходимого нам чата. В данном боте используется именно urllib2, а не urllib, потому что urllib2 может в отличие от старой версии добавлять заголовки к запросу, что и используется для связи с сервером. Данная библиотека с помощью функции urllib2.Request задает исходные данные, которые отправляются на указанный нами сервер (искомый чат). Для отправления используются метаданные, указанные в файле config;

– json – формат обмена данными, который имеет вид понятного человеческому глазу текста (рис. 2). Данная библиотека используется для загрузки специальных данных, которые в себе содержат информацию о трансляции, такую как: количество пользователей в чате и их роли (модераторы, админы, простые зрители), кто проводит прямую трансляцию и количество зрителей, которые просто находятся на трансляции и при этом не находятся в чате.


```
{
  "_links": {},
  "chatter_count": 0,
  "chatters": {
    "broadcaster": [],
    "vips": [],
    "moderators": [],
    "staff": [],
    "admins": [],
    "global_mods": [],
    "viewers": []
  }
}
```


-

Рис. 2. Пример данных в формате json

– time – модуль для работы со временем, из данного модуля используется только функция sleep, которая производит задержку выполнения программы, т.е. устанавливается время ожидания до очередного использования нужного нам цикла.

Третий файл является самой имплементацией бота, т.е. его фактическая реализация. В данном файле реализуется отправка ботом различных сообщений в чат в заданном промежутке, а также ответы на сообщения пользователей чата. Так же здесь реализуется проверка связи между сервером и ботом посредством принятия сообщений от сервера, декодирование их и отправка обратно. В него импортированы библиотеки и модули, такие как:

– файлы, которые были созданы до этого: config и utils, необходимы для отправки сообщений в чат и проверки связи с сервером;

– socket – данный модуль необходим для всех клиент-серверных приложений, для непосредственной связи с сервером;

– re – модуль необходимый для работы с регулярными выражениями. В данном боте, этот модуль избавляет от переписывания ссылки на необходимый нам сервер. Также необходим для создания отдельного объекта, который будет использован для поиска имен пользователей чата;

– time – в данном файле модуль time необходим для указания точного времени для некоторых команд в чате, а также для установки задержки перед очередным использованием бесконечного цикла;

– Thread – модуль позволяющий создать отдельный поток, который используется для добавления пользователей в файл utils, а также необходим для отправления сообщения в чат параллельно с основным потоком.

На рис. 3 представлена диаграмма классов, которая показывает логическую взаимосвязь элементов бота. В них включаются 3 основные файла, между ними

постоянная связь, и вспомогательные модули, которые использую ботом при определенных обстоятельствах.

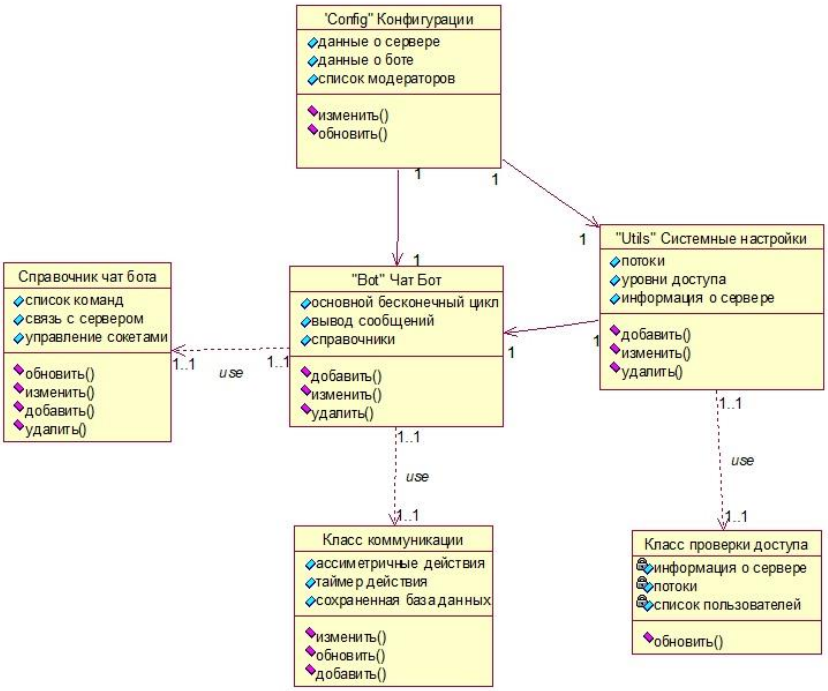

Рис. 3. Диаграмма классов

Перспективы развития

Созданный бот может использоваться на многих серверах кроме twitch.tv, а также имеет большой потенциал развития: можно подключиться к API необходимого сервера, что предоставит огромное количество новых возможностей для усовершенствования кода. Данный бот имеет только базовые возможности необходимые для автоматизации работы людей ведущих прямые трансляции в интернете.

Вывод. Цель создания данного бота, т.е. автоматизация рабочего места ведущего трансляции на площадке twitch.tv была выполнена. Визуальный интерфейс отсутствует и отслеживать действия бота можно только через консоль, но это не сильно влияет на удобство пользования.

Данный бот был протестирован на нескольких пользователях twitch.tv, ошибок при этом выявлено не было.

Список литературы

1. Библиотеки и модули Python [Электронный ресурс]. – Режим доступа: https://pythonworld.ru

2. Кватрани Т. Rational Rose 2000 и UML. Визуальное моделирование / пер. с англ. – М.: ДМК Пресс, 2001. – 176 с.