

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

Создание Telegram бота для обучения языку программирования Python
АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 441 группы

направления 02.03.03 Математическое обеспечение и администрирование
информационных систем

факультета компьютерных наук и информационных технологий

Кузьмина Дмитрия Сергеевича

Научный руководитель:

Старший преподаватель кафедры ИиП _____ Е. Е. Лапшева

подпись, дата

Зав. кафедрой:

Доцент _____ М. В. Огнева

подпись, дата

Саратов 2022

ВВЕДЕНИЕ

Актуальность темы. Чат-бот – это программа, которая работает в приложениях-мессенджерах и социальных сетях. Бот – это программа, имитирующая деятельность человека. Боты используются для автоматизации процессов взаимодействия с пользователем в чатах (средствах обмена сообщениями). Боты сами задают вопросы пользователям и отвечают, ищут необходимую информацию, предоставляют полезный функционал и так далее. Сейчас чат-боты набрали огромную популярность и широко применяются для:

- технической поддержки;
- консультации по типовым вопросам;
- поиска необходимой информации;
- помощи в выборе продукта;
- рассылки сообщений;
- развлечения пользователей и других нестандартных функций.

Чаще всего чат-боты – это виртуальные собеседники, которые общаются с пользователями в режиме реального времени, имитируя реального человека. Также существует другой вид – сервисные чат-боты, которые выполняют некоторые поручения: осуществляют заказы, совершают денежные транзакции, рекламируют продукт и так далее. Чат-боты обладают рядом преимуществ:

- возможность круглосуточной работы;
- оперативность: пользователь получает ответ мгновенно;
- автоматизация бизнес-процессов;
- упрощенная коммуникация;
- сегментация клиентов: с помощью ботов можно проводить опросы, выявлять потребность и делать уникальные предложения для разных аудиторий;
- уменьшение стоимости привлечения клиента.

Но также есть и недостатки у ботов, главный из которых – ошибки в нестандартных ситуациях: бот запрограммирован на определенные сценарии и не может помочь пользователю абсолютно во всех задачах, отсюда возникает необходимость вмешательства реальных людей.

Чат-боты используются практически во всех мессенджерах и социальных сетях, таких, например, как: ВКонтакте, Facebook, Viber и Telegram. Примеры таких ботов: бот такси «Максим», бот Aviasales (ищет авиабилеты по минимальным ценам), T-Mobile (бот для продажи мобильных телефонов), Bot Burger (данный бот специализируется на доставке еды) и так далее.

Цель бакалаврской работы – создание чат-бота в мессенджере Telegram для изучения языка программирования Python.

Поставленная цель определила **следующие задачи**:

1. Изучить основы языка программирования C# и Python.
2. Изучить основы работы с платформой .NET Core.
3. Изучить библиотеки Telegram.Bot, Newtonsoft.Json, IronPython, AngleSharp языка программирования C#, используемые для данной работы.
4. Ознакомиться с работой Telegram API (Application Program Interface).
5. Изучить виды взаимодействий чат-бота и пользователя.
6. Найти необходимый материал по языку Python, который бот будет использовать для интерактивного обучения пользователя.
7. Создать бота и запустить его.
8. Протестировать работу чат-бота.

Методологические основы создания чат-ботов представлены в документации Telegram API.

Практическая значимость бакалаврской работы заключается в обучении пользователей языку программирования Python через интерактивное взаимодействие с чат-ботом.

Структура и объём работы. Бакалаврская работа состоит из введения, трех разделов, заключения, списка использованных источников и трех

приложений. Общий объем работы – 77 страниц, из них 53 страницы – основное содержание, включая 22 рисунка, цифровой носитель в качестве приложения, список использованных источников информации – 20 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «TELEGRAM API И ПЛАТФОРМА .NET»

посвящен описанию основных инструментов, библиотек и способов разработки чат-ботов для Telegram мессенджера. Перечень понятий и определений раздела представлены ниже.

Telegram – это популярный мессенджер, который набрал огромную популярность за последние годы благодаря некоммерческому характеру, поддержке кроссплатформенности, гарантиям безопасности и открытым API.

API (Application Programming Interface) представляет собой совокупность различных инструментов, функций, реализованных в виде интерфейса для создания новых приложений, благодаря которому одна программа будет взаимодействовать с другой.

Наиболее известным Telegram API является Bot API, основанное на HTTP-запросах и позволяющее разработчикам взаимодействовать с бот-платформой. Bot API позволяет программистам контролировать Telegram ботов путем получения сообщений, их обработки и отправки пользователям.

Кроме Bot API существует сам Telegram API. Это API используется приложением Telegram для осуществления всех действий, которые доступны пользователям данной платформы: отображение чатов, отправка и получение сообщений, создание нового канала и так далее. С помощью Telegram API можно запрограммировать абсолютно любые действия в приложении Telegram.

Боты – это один из типов приложений, которые запускаются непосредственно в Telegram. Пользователи могут взаимодействовать с ботами, посылая им сообщения, команды или встроенные запросы. В свою очередь, программист контролирует ботов, используя HTTPS-запросы.

Telegram предоставляет разработчикам библиотеку Telegram.Bot для работы в .NET. Данная библиотека содержит огромное количество разнообразного функционала, который помогает легко и быстро создавать логику для Telegram бота. Чтобы начать пользоваться данной библиотекой,

необходимо ее скачать, используя NuGet Package Manager (менеджер пакетов NuGet).

Существует два способа взаимодействия бота с Telegram: webhook и long polling. При работе с webhook Telegram сам оповещает бота о различных изменениях и событиях в тех чатах, каналах, группах, в которых находится бот. Long polling метод подразумевает, что боту раз в определенное время необходимо самостоятельно обращаться к Telegram и получать от него различные обновления.

.NET экосистема состоит из различных компонентов разработки и включает в себя различные среды выполнения, в том числе и следующие:

- .NET Framework (WPF, Windows Forms, ASP.NET) – Ориентирован для Windows;
- .NET Core (ASP.NET Core, Universal Windows Platform, UWP) – Кроссплатформенное решение;
- Mono for Xamarin (IOS, OS X, Android) – Кроссплатформенная система для разработки мобильных приложений.

AngleSharp – это .NET библиотека, которая предоставляет возможность парсить гипертекст HTML, SVG и MathML. Также поддерживается XML разметка. Еще одной особенностью является то, что с помощью данной библиотеки можно парсить CSS. AngleSharp умеет работать с HTML5 DOM (объектная модель документа) представлением и позволяет осуществлять запросы выбора нужного элемента в DOM дереве.

IronPython – это библиотека с открытым исходным кодом, которая представляет собой реализацию языка программирования Python, тесно интегрированную с платформой .NET. IronPython построен на основе DLR (Dynamic Language Runtime – Среда выполнения динамических языков) и позволяет использовать преимущества .NET платформы и библиотек Python в одной среде, что предоставляет разработчикам огромные возможности при разработке приложений.

Асинхронное программирование – это средство параллельного программирования, в котором единица работы выполняется отдельно от основного потока приложения и уведомляет вызывающий поток о своем завершении, сбое или прогрессе.

Внедрение зависимостей – это одна из наиболее известных техник, которая помогает создавать более удобный в сопровождении код. .NET Core предоставляет готовый инструмент для работы с внедрением зависимостей.

SQLite – это библиотека, которая предоставляет систему управления реляционными базами данных (СУБД). Слово lite в названии означает легкость с точки зрения настройки, администрирования базы данных и необходимых ресурсов. SQLite имеет следующие особенности: автономность, отсутствие сервера и транзакционность.

Второй раздел «ЯЗЫК PYTHON И DATA SCIENCE БИБЛИОТЕКИ» посвящен описанию языка программирования Python, а также основных библиотек по анализу и обработке данных: Numpy, Matplotlib и Pandas.

Третий раздел «СОЗДАНИЕ TELEGRAM БОТА НА ЯЗЫКЕ C# ДЛЯ ОБУЧЕНИЯ ЯЗЫКУ ПРОГРАММИРОВАНИЯ PYTHON» посвящен реализации приложения чат-бота, написанного на языке программирования C#. В первом пункте «Подготовка библиотек, инструментов и среды разработки» перечислены основные инструменты и библиотеки, с помощью которых было разработано приложение, среди них: Visual Studio 2019, C# 8.0, .NET Core 3.1, библиотека Telegram.Bot 15.7.1.

В пункте «Описание работы» представлены этапы создания чат-бота. Первый этап состоит в том, чтобы зарегистрировать чат-бота в Telegram при помощи другого бота «BotFather», также перечислены команды, на которые будет реагировать чат-бот:

- /start – команда, которая приветствует пользователя и показывает ему весь доступный функционал бота;

- `/code` – команда, которая позволяет пользователю ввести Python скрипт, запустить его и посмотреть результат выполнения;
- `/tutorial` – команда, которая выводит список тем для прохождения курса по языку Python;
- `/quiz` – команда, которая позволяет пройти курс на знание языка Python;
- `/contest` – команда, которая дает возможность решить задачи по программированию для языка Python.

Следующим этапом является создание архитектуры приложения. Проект состоит из нескольких составных частей: парсер Python курса, база данных, уровень работы с базой данных, уровень бизнес-логики и уровень представления, описывающий интерфейс бота и различные сценарии взаимодействия с пользователем. Чат-боту для взаимодействия с пользователем необходимо постоянно манипулировать данными: сохранять состояние пользователя, предоставлять пользователю данные, извлеченные с помощью парсера. Для осуществления подобных операций необходимы сущности. Основная логика обработки и предоставления контента расположена в сервисном уровне. Сервисы представлены в виде отдельных классов. Для поддержки нескольких пользователей была добавлена SQLite база данных. SQLite была выбрана по той причине, что эта база данных не занимает много места, проста в настройке и ее можно хранить прямо в проекте.

По итогу данной работы был разработан чат-бот для Telegram мессенджера. В ходе работы возникло множество проблем, связанных с отсутствием подробной документации по Telegram API для .NET. Также возникали сложности с парсингом Python кода из HTML страницы, здесь пришлось вручную записать Python скрипты в файлы json формата и каждый раз парсить код из эти фалов. Помимо этого, возникала проблема поддержки нескольких пользователей, которая решилась добавлением базы данных, где хранятся текущие состояния всех пользователей.

ЗАКЛЮЧЕНИЕ

Целью данной работы является создание Telegram бота для обучения языку программирования Python. В ходе данной работы были изучены языки программирования Python и C#, и все необходимые библиотеки такие, как Telegram.Bot, Newtonsoft.Json, IronPython, AngleSharp. Также были изучены принципы работы чат-ботов в мессенджере Telegram. Был изучен и выбран необходимый материал для обучения языку Python. Кроме того, был создан Telegram-бот, а затем был протестирован несколькими реальными пользователями. Данная работа имеет дальнейшее развитие: можно сделать перевод всего курса на русский язык и добавить поддержку как русского, так и английского языков, можно добавить больше разнообразного образовательного контента, также можно продолжить тему data science и сделать курс по машинному обучению и нейронным сетям, помимо этого можно добавить поддержку других мессенджеров и социальных сетей таких, как Facebook, VK, Viber и других.

Основные источники информации:

1. Telegram API: «Официальная документация» [Электронный ресурс]. URL: <https://core.telegram.org/api> (дата обращения: 06.01.2021).
2. AngleSharp: «API Documentation» [Электронный ресурс]. URL: <https://anglesharp.github.io/docs/API.html> (дата обращения: 09.01.2021).
3. IronPython: «Overview of IronPython» [Электронный ресурс]. URL: <https://ironpython.net/> (дата обращения: 09.01.2021).
4. Telegram Bots: «Официальная документация» [Электронный ресурс]. URL: <https://core.telegram.org/bots> (дата обращения: 06.01.2021).
5. Michael J. Foord, Christian Muirhead IronPython in Action / © 2009 by Manning Publications Co. All rights reserved.