**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

**Институт среднего профессионального образования**

**ОТЧЕТ**

**по производственной практике (по профилю специальности)**

по профессиональному модулю

ПМ.02 Осуществление интеграции программных модулей\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(код и наименование)

Специальность 09.02.07 Информационные системы и программирование .

(код и наименование специальности)

Студенту IVкурса 42919/4 группы

Буров Сергей Витальевич

(фамилия, имя, отчество)

Место прохождения практики СПБ ГУП «СПБ ИАЦ», ул. Черняховского 59, СПБ

(наименование и адрес организации)

Период прохождения практики

с «11» марта 2024 г. по «13» апреля 2024 г.

Руководители практики:

От колледжа \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Зернова Е.Н. .

(подпись) (Ф.И.О.)

От организации \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Стариков А.Ю.\_ \_\_\_\_\_

(подпись) (Ф.И.О.)

М.П.

Итоговая оценка по практике (от колледжа):\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург

2024

**Задание на производственную практику (по профилю специальности)**

по профессиональному модулю ПМ.02 Осуществление интеграции программных модулей .

(код и наименование)

Специальность09.02.07 Информационные системы и программирование .

(код и наименование специальности)

Студенту IVкурса 42919/4 группы

Буров Сергей Витальевич

(фамилия, имя, отчество)

Место прохождения практики СПБ ГУП «СПБ ИАЦ», ул. Черняховского 59, СПБ

(наименование и адрес организации)

Период прохождения практики

с «11» марта 2024 г. по «13» апреля 2024 г.

**Виды работ, обязательные для выполнения** *(переносится из программы соответствующего ПМ)*

1. Знакомство с предприятием
2. Участие в выработке требований к программному обеспечению
3. Участие в проектировании программного обеспечения с использованием специализированных программных пакетов
4. Участие в интеграции программных модулей
5. Разработка рабочего проекта и технологической документации
6. Приемо-сдаточные мероприятия

**Индивидуальное задание (***заполняется в случае необходимости дополнительных видов работ для решения практикоориентированных задач и т.д.)*

1. Анализ предприятия «»
2. Анализ задач для разработки проекта
3. Описание требований к проекту
4. Анализ рынка существующих решений
5. Анализ целесообразности выполнения проекта
6. Анализ и выбор методов решения
7. Анализ и выбор средств разработки

Задание выдал «12» февраля 2024 г. \_\_\_\_\_\_\_\_\_\_\_\_\_ Стариков А.Ю.

(подпись) (Ф.И.О.)

С заданием ознакомлен

«\_\_» \_\_\_\_\_\_\_\_\_\_ 2024 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Буров С.В.

(подпись) (Ф.И.О. студента)

**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

**Институт среднего профессионального образования**

**ДНЕВНИК**

**прохождения производственной практики**

**(по профилю специальности)**

по профессиональному модулю

ПМ.02 Осуществление интеграции программных модулей .

(код и наименование)

Специальность09.02.07 Информационные системы и программирование .

(код и наименование специальности)

Студенту IVкурса 42919/4 группы

Буров Сергей Витальевич

(фамилия, имя, отчество)

Место прохождения практики СПБ ГУП «СПБ ИАЦ», ул. Черняховского 59, СПБ

(наименование и адрес организации)

Период прохождения практики

с «11» марта 2024 г. по «13» апреля 2024 г.

Руководители практики:

От колледжа \_\_\_\_\_\_\_\_\_\_\_\_ Зернова Е.Н. .

(подпись) (Ф.И.О.)

От организации\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Стариков А.Ю\_\_\_\_

(подпись) ( Ф.И.О.)

М.П.

Санкт-Петербург

2024

**Содержание дневника**

|  |  |  |
| --- | --- | --- |
| Дата | Виды выполненных работ и заданий по программе практики | Подпись руководителя практики |
| 1 | 2 | 3 |
| 11.03 | Вводный инструктаж по технике безопасности. Организационно-производственная структура предприятия. Основной состав средств вычислительной техники. |  |
| 12.03 | Ознакомление с характеристиками средств вычислительной техники, их назначением и влиянием на производство основных видов продукции. |  |
| 13.03 | Ознакомление с основными этапами и методикой проектирования вычислительных задач на предприятии |  |
| 14.03 | Разработка технического задания |  |
| 15.03 | Этапы проектирования программного комплекса |  |
| 18.03 | Исследовательские работы. Обоснование принципиальной возможности решения задачи |  |
| 19.03 | Участие в выработке требований к программе, утверждение технического задания |  |
| 20.03 | Разработка схемы алгоритма программного продукта |  |
| 21.03 | Определение форм представления входных и выходных данных |  |
| 22.03 | Разработка структуры программы |  |
| 25.03 | Составление схем программы в соответствии с ЕСПД |  |
| 26.03 | Разработка пояснительной записки |  |
| 27.03 | Проектирование пользовательского интерфейса |  |
| 28.03 | Разработка схем алгоритмов модулей |  |
| 29.03 | Приведение схемы алгоритма модуля к структурному виду |  |
| 01.04 | Написание структурных модулей и их отладка |  |
| 02.04 | Написание структурных модулей и их отладка |  |
| 03.04 | Разработка текста программы |  |
| 04.04 | Разработка текста программы |  |
| 05.04 | Разработка интерфейса пользователя |  |
| 08.04 | Разработка методики испытаний |  |
| 09.04 | Отладка и тестирование программы |  |
| 10.04 | Оптимизация программы |  |
| 11.04 | Разработка технологической документации |  |
| 12.04 | Проведение приемо-сдаточных испытаний |  |
| 13.04 | Оформление отчета по практики |  |

**АТТЕСТАЦИОННЫЙ ЛИСТ**

**ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ (ПО ПРОФИЛЮ СПЕЦИАЛЬНОСТИ)**

по профессиональному модулю

ПМ.02 Осуществление интеграции программных модулей .

(код и наименование)

Специальность09.02.07 Информационные системы и программирование .

(код и наименование специальности)

Студенту IVкурса 42919/4 группы

Буров Сергей Витальевич

(фамилия, имя, отчество)

Место прохождения практики СПБ ГУП «СПБ ИАЦ», ул. Черняховского 59, СПБ

(наименование и адрес организации)

с «11» марта 2024 г. по «13» апреля 2024 г.

**Виды и качество выполнения работ**

|  |  |  |
| --- | --- | --- |
| Виды выполненных работ обучающимся  во время практики | Объем работ, час. | Качество выполнения работ (оценка по пятибальной системе) |
| Знакомство с предприятием | 18 |  |
| Участие в выработке требований к программному обеспечению | 24 |  |
| Участие в проектировании программного обеспечения с использованием специализированных программных пакетов | 36 |  |
| Участие в интеграции программных модулей | 30 |  |
| Разработка рабочего проекта и технологической документации | 48 |  |
| Приемо-сдаточные мероприятия | 24 |  |

**Характеристика учебной/профессиональной деятельности обучающегося во время производственной практики (по профилю специальности):**

Общие и профессиональные компетенции, предусмотренные программой практики, освоены/не освоены.

(нужное подчеркнуть)

Итоговая оценка по практике (от колледжа): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель практики Зернова Е.Н. \_\_\_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О.) (подпись)

Дата «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024г.

**ХАРАКТЕРИСТИКА**

от руководителя производственной (по профилю специальности) практики

В период с 11 марта по 13 апреля 2024 года Буров Сергей Витальевич успешно завершил практику на предприятии СПБ "ИАЦ". Разработал чат-бота, проявив при этом высокую ответственность и исполнительность. Чат-бот, созданный им, оценили за его удобство использования и отсутствие проблем при проверке.

Кроме того, в процессе индивидуальной работы Буров Сергей проявил исключительную самоорганизацию и умение эффективно планировать свою деятельность.

Он демонстрировал высокий уровень самодисциплины и способность к самостоятельной работе без посторонней помощи. Его преданность и настойчивость помогли ему преодолевать трудности и достигать поставленных целей.

Итоговая оценка по практике \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

Место работы и должность руководителя СПБ ГУП «СПБ ИАЦ», Начальник Сектора автоматизации деятельности исполнительных органов государственной власти Отдела автоматизации государственных услуг

и функций территориальных и отраслевых органов №237.

Ф.И.О. руководителя Стариков Александр Юрьевич.

13 апреля 2024 г. Подпись \_\_\_\_\_\_\_\_\_\_\_\_\_

М.П.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 10](#_Toc163863692)

[1. Индивидуальное задание по практике 11](#_Toc163863693)

[1.1. Анализ предприятия 11](#_Toc163863694)

[1.2. Анализ поставленных задач 12](#_Toc163863695)

[1.3. Создание чат-бота 12](#_Toc163863696)

[1.4. Настройка чат-бота 13](#_Toc163863697)

[1.5. Загрузка чат-бота в мессенджер 14](#_Toc163863698)

[1.6. Структура приложения 14](#_Toc163863699)

[1.7. Реализация чат-бота 18](#_Toc163863700)

[1.8. Разработка требований к поставленным задачам 21](#_Toc163863701)

[2. Описание работы приложения. 22](#_Toc163863702)

[3. Виды работ, обязательные для выполнения. 23](#_Toc163863703)

[3.1. Проведение эскизного проектирования 23](#_Toc163863704)

[3.2. Отладка и тестирование приложения 24](#_Toc163863705)

[ЗАКЛЮЧЕНИЕ 26](#_Toc163863706)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 27](#_Toc163863707)

[ПРИЛОЖЕНИЕ А 28](#_Toc163863708)

# ВВЕДЕНИЕ

Сегодня прогрессивные технологии революционизируют мир, превращая рутинные процессы в автоматизированные системы, сокращая время на выполнение задач и повышая стандарты предоставления услуг. В этой цифровой эпохе чат-боты стали неотъемлемой частью инновационного прогресса. Эти программы способны вести диалог с пользователями в реальном времени через мессенджеры и социальные сети, а их функционал становится все более широким и разнообразным.

Чат-боты широко применяются в различных сферах, от бизнеса и маркетинга до образования и медицины. Они могут выполнять разнообразные задачи, от предоставления стандартной информации до обучения пользователей. Создание и внедрение чат-бота в бизнес-процессы приводит к улучшению взаимодействия с клиентами, увеличивая скорость ответов и упрощая процесс обслуживания.

Конкурентоспособность бизнеса напрямую зависит от качества чат-бота. Информативность, понятность и эффективность в общении с пользователем определяют, вернется ли клиент снова или уйдет к конкурентам. Создание и постоянное совершенствование чат-бота становятся ключевыми факторами, влияющими на успех и прибыльность предприятия в цифровой эпохе.

В ходе учебной практики нужно решить следующие задачи:

1. Анализ предприятия СПб ИАЦ;
2. Анализ поставленных задач;
3. Разработка требований к поставленной задачи;
4. Разработка чат-бота.

Структура отчета состоит из введения, двух пунктов, разделенных на подпункты, заключения и списка литературы.

# Индивидуальное задание по практике

# Анализ предприятия

Предприятие Санкт-Петербургский информационно-аналитический центр (СПб ИАЦ) специализируется в сфере информатизации и информационного обеспечения органов государственной власти Санкт-Петербурга, а также занимается разработкой информационных систем. Основные цели деятельности центра включают:

* Создание единого информационного пространства для органов государственной власти Санкт-Петербурга;
* Внедрение информационных технологий в работу исполнительных органов государственной власти города;
* Предоставление услуг в области информатизации и информационного обеспечения.
* Осуществление мониторинга социально-экономического развития и информационно-аналитическое обеспечение администрации Санкт-Петербурга;

Деятельность компании основывается на сборе, аналитике и систематизации данных. В первую очередь, для гос. служащих Санкт-Петербурга, но также предоставляются услуги и отдельным клиентам.

Деятельность компании базируется на сборе, анализе и систематизации данных, прежде всего для государственных служащих Санкт-Петербурга, но также услуги предоставляются и отдельным клиентам. Кроме того, СПб ИАЦ занимается разработкой и поддержкой информационных систем, играя важную роль в сфере государственной информатизации. Компания также тесно связана с порталом "Госуслуг", что подчеркивает ее значимость на рынке. Необходимо отметить, что спрос на информационные и аналитические услуги высок, а ниша рынка, которую занимает компания, является важной и влиятельной.

# Анализ поставленных задач

На практике нам выдали задание на создание чат-бота по теме, были предоставлены способы создания и модерирования чат-бота. При изучении всех материалов было принято решение делать бота для мессенджера «Telegram», так как практика и все основные чаты сотрудников базируются там. В ходе нашей практики было поручено создать чат-бота. После изучения предоставленной информации было принято решение разработать бота для платформы мессенджера "Telegram", поскольку большинство сотрудников используют мессенджер "Telegram" и все основные обсуждения сотрудников ведутся именно там.

# Создание чат-бота

Для создания чат-бота было выбрано приложение Visual Studio Code, именно в нем ввелись все основные задачи по созданию кода. Так же было задействовано встроенное приложение на MacBook – «Терминал», с помощью которого я создавал необходимые корневые папки и добавлял библиотеки и так же был задействован бот мессенджера "Telegram" – BotFather, с помощью которого так же велась настройка моего чат бота.

Используя все необходимые инструменты, я начал создавать и программировать бота.

Каждый инструмент выполнял свои задачи, такие как:

* Создание чат-бота "Telegram", получение его API ключа, выбор никнейма, выбор ссылки на него и выбор аватара для бота. Получение токена бота для последующей настройки функций, токен использует APIпосредством http-протокола. Это так называемый ключ для настройки и опознавания бота в системе мессенджера, он уникален и должен быть известен только разработчикам бота. - BotFather;
* Добавление библиотек, создание корневых папок и всех необходимых документов - Терминал;
* Создание начинки, а именно написание кода и вставление в него нашего API ключа для корректной работы – VSCode;

Для создания бота нужно выполнить несколько действий:

* Перейти в диалог с инструментом для разработки чатов — https://telegram.me/BotFather;
* Нажать кнопку «Start» или введите в диалоге команду /start;
* Далее введите команду /newbot, чтобы сделать новый бот;
* Указать имя бота.
* Указать системное имя бота, которое нужно для подключения к чату.

При создании чат бота создаётся его внешний вид и имя.

# Настройка чат-бота

Для настройки функционала и записи бота я выбрал Visual Studio Code.

Я выбрал сервис Visual Studio Code для создания Telegram бота по нескольким причинам:

* Легко настраивается и обладает широким сообществом пользователей, что обеспечивает доступ к множеству расширений и дополнений для улучшения процесса разработки. Это позволяет мне адаптировать среду разработки под свои потребности и повышает производительность.
* Visual Studio Code предоставляет мощные инструменты для разработки, включая поддержку различных языков программирования, удобную систему управления расширениями и интеграцию с Git для контроля версий. Это позволяет мне комфортно работать с кодом и эффективно управлять проектом.
* Visual Studio Code является бесплатным и открытым исходным кодом, что делает его доступным для всех разработчиков без необходимости платить за лицензию. Это важно для меня, поскольку я могу использовать его без каких-либо финансовых затрат и иметь возможность внести свой вклад в его развитие, если это необходимо.

Перед публикацией бота необходимо провести тестирование для выявления ошибок и недостатков в его функционале. Тестирование должно охватывать широкий спектр сценариев использования, включая данные, выходящие за пределы стандартных входных параметров, а также проверку на возможность преодоления ограничений. Это позволит улучшить алгоритмы работы бота и найти новые способы решения задач.

После завершения тестирования и внесения необходимых корректировок можно приступить к публикации бота. Для этого необходимо создать пустого бота с помощью встроенного сервиса от Telegram и получить токен, который будет использоваться для взаимодействия с API мессенджера.

# Загрузка чат-бота в мессенджер

Для загрузки чат бота в мессенджер нужно проделать все предыдущие шаги по порядку.

Перед публикацией бота необходимо провести тестирование для выявления ошибок и недостатков в его функционале. Тестирование должно охватывать широкий спектр сценариев использования, включая данные, выходящие за пределы стандартных входных параметров, а также проверку на возможность преодоления ограничений. Это позволит улучшить алгоритмы работы бота и найти новые способы решения задач.

После завершения тестирования и внесения необходимых корректировок можно приступить к публикации бота. Для этого необходимо создать пустого бота с помощью встроенного сервиса от Telegram и получить токен, который будет использоваться для взаимодействия с API мессенджера.

# Структура приложения

Функционал бота:

1. Пользователь находит моего бота с помощью уникального тега

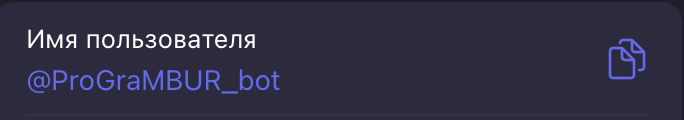


Рисунок 1 (Ник-нейм бота)

1. Пользователь заходит в чат и появляется команда “/**start”**, либо кнопка “**начать”**

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 2 (Кнопка «Начать»)

1. После нажатия на кнопку появляется текст с информацией:

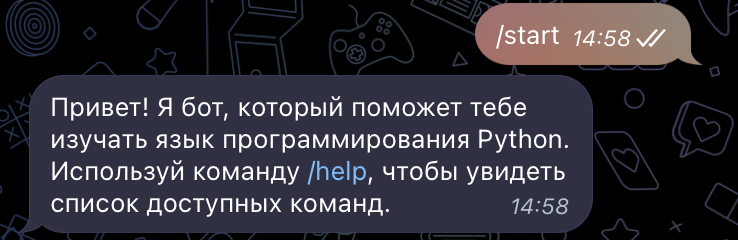


Рисунок 3 (Кнопка «/start»)

1. После нажатия на команду /help появляется список доступных команд, а так же пояснение к каждой команде:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 4 (Кнопка «/help»)

1. При нажатии на каждую кнопку появляется та или иная информация:

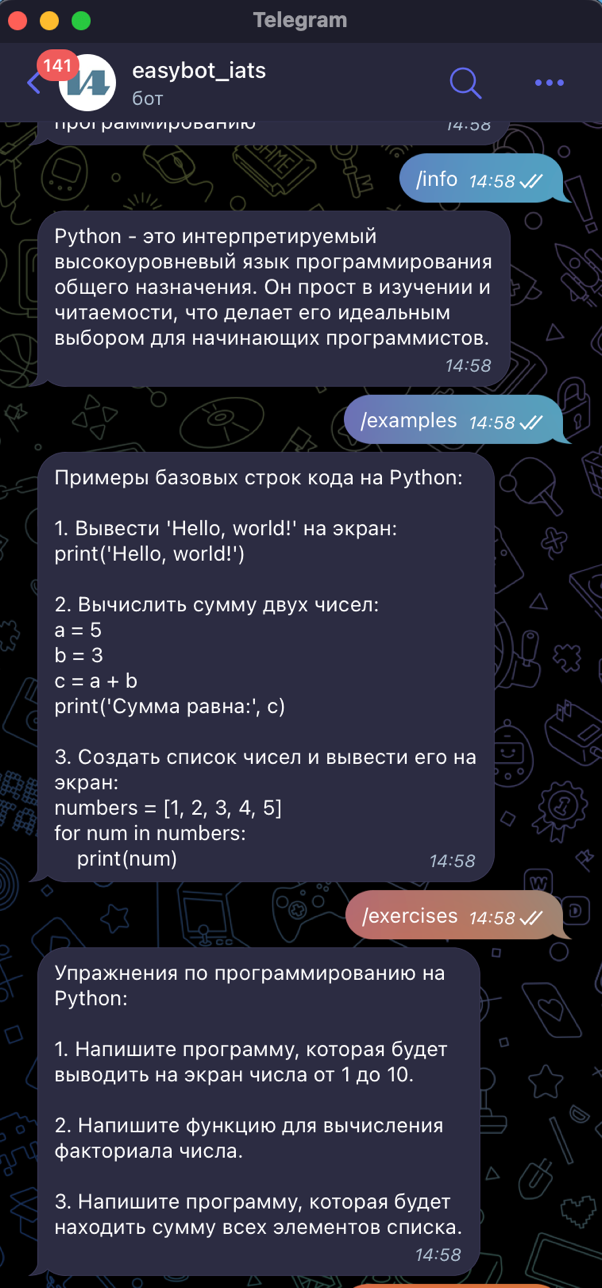


Рисунок 5 (другие команды)

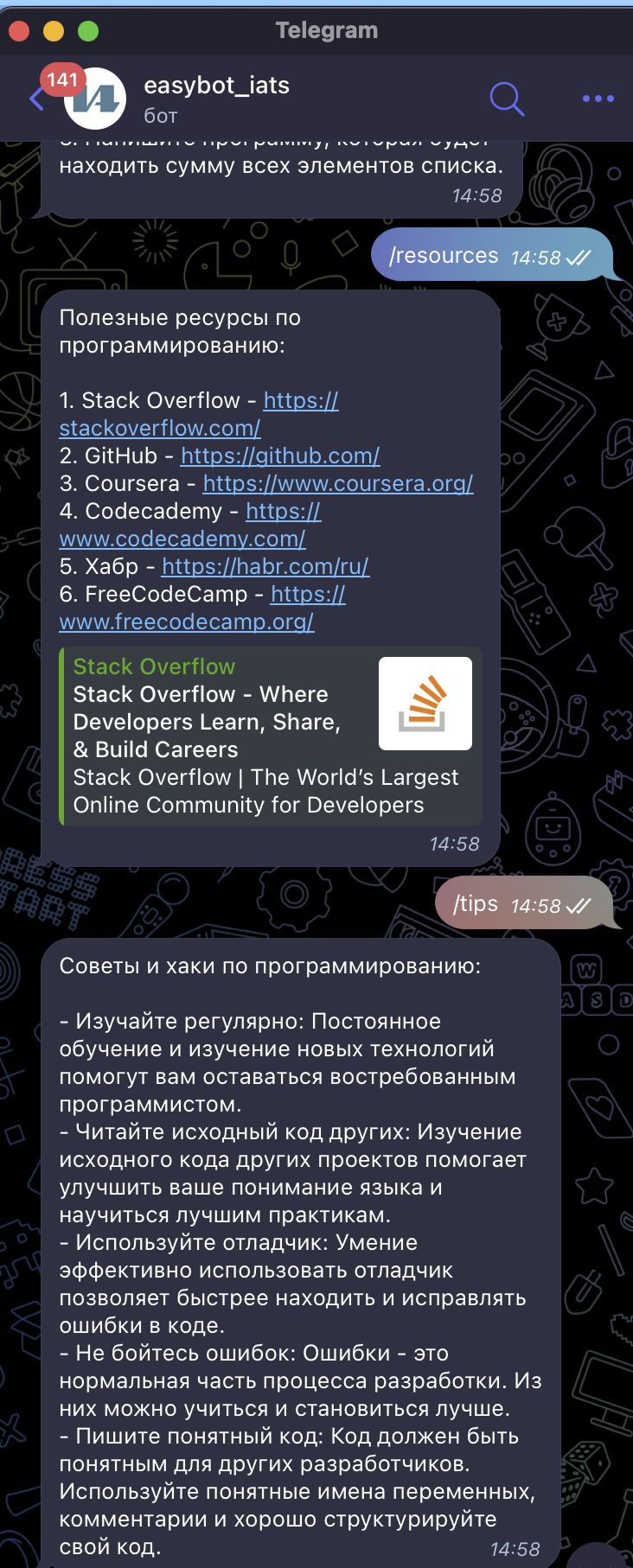


Рисунок 6 (другие команды)

1. При вводе слов с клавиатуры непонятных для бота появляется соответствующая ошибка:

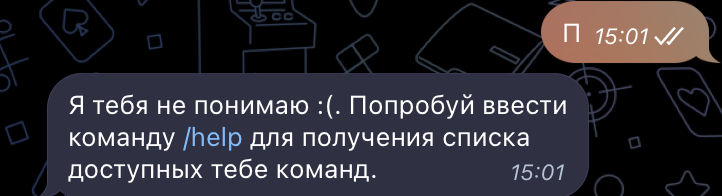


Рисунок 7 (ошибка ввода)

# Реализация чат-бота

Первым шагом надо было подключить все нужные нам библиотеки:

using System;

using System.Threading.Tasks;

using Telegram.Bot;

using Telegram.Bot.Args;

using Telegram.Bot.Types.Enums;

using Telegram.Bot.Types;

дальше нам нужно добавить токен нашего бота для корректной работы:

static async Task Main()

{

botClient = new TelegramBotClient("токен нашего бота");

Предварительно мы создали нашего бота с помощью бота для создания ботов, скажем так.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 8 (создание бота)

**@BotFather** – с помощью него мы и создали бота

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 9 (чат с ботом по созданию бота)

Когда мы создали бота, нам присваивается уникальный токен, который нужно будет вписываться в код:

**bot = telebot.TeleBot(“token”)** – сюда мы впишем токен нашего бота.

Теперь в **@BotFather** мы пропишем название для нашего бота.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 10 (имя)

Там же можно установить фото для нашего бота



Рисунок 11 (аватар)

Пропишем для чего команды, которые потом можно будет использовать в коде, чтобы все работало

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 12 ( команды)

В коде после токена делаем лист команд:  
await botClient.SetMyCommandsAsync(new[]

{

new BotCommand { Command = "start", Description = "Начать обучение языку программирования Python" },

new BotCommand { Command = "help", Description = "Показать список доступных команд" },

new BotCommand { Command = "info", Description = "Получить базовую информацию о языке программирования Python" },

new BotCommand { Command = "examples", Description = "Получить примеры базовых строк кода на Python" },

new BotCommand { Command = "exercises", Description = "Получить упражнения по программированию на Python" },

new BotCommand { Command = "resources", Description = "Получить полезные ресурсы по программированию" },

new BotCommand { Command = "tips", Description = "Получить советы и хаки по программированию" }

});

После расписываем каждую команду и вводим то что должна выводить каждая команда например:

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 13 (фрагмент)

# Разработка требований к поставленным задачам

Все требования к выполненным заданиям были предоставлены нашим куратором после выдачи самого задания.

Сами требования к поставленным задачам:

* Название соответственное теме и аватар компании ИАЦ;
* Логичность работы бота, то есть интуитивно понятный ход действий при использовании;
* Имитация общения, бот должен подражать человеку в общении;
* Краткость, бот должен общаться малыми однозначными фразами;
* Ясность – все этапы заполнения данных должны быть понятны пользователю.;
* Обработка ошибок – если данные заполнены некорректно, должно выводиться сообщение об неправильности ввода;
* Помощь, добавление функций по информированию пользователей о его возможностях.

# Описание работы приложения.

Мой чат-бот предназначен для предоставления обучающего контента по языку программирования Python. Его основная функция - отвечать на вопросы о синтаксисе, предоставлять примеры кода и давать задания для практики. Пользователь может обратиться к боту с запросом о конкретном аспекте Python, и бот предоставит соответствующую информацию или пример кода.

Чат-бот также может помочь пользователям в решении конкретных задач или проблем, с которыми они могут столкнуться при программировании на Python. Это может включать в себя помощь в исправлении ошибок в коде, объяснение сложных концепций или поддержку при разработке проектов.

Кроме того, бот может предложить пользователю пройти интерактивные курсы или обучающие материалы по Python, чтобы расширить свои знания и навыки в этой области. Он также может предоставить ссылки на полезные ресурсы, такие как документация Python, онлайн-курсы и сообщества разработчиков, где пользователь может найти дополнительную поддержку и информацию.

Таким образом, наш чат-бот предоставляет возможность эффективно изучать Python, отвечая на вопросы, предоставляя примеры и задания для практики, что делает его полезным инструментом как для начинающих, так и для более опытных программистов.

# Виды работ, обязательные для выполнения.

# 3.1. Проведение эскизного проектирования

На этом этапе были созданы UML диаграммы:

* Диаграмма прецедентов;
* Диаграмма активностей

Диаграмма прецедентов предоставлена на рисунке 14

Изображение выглядит как текст, снимок экрана, диаграмма, дизайн

Автоматически созданное описание

Рисунок 14 – (Диаграмма прецедентов)

В данной диаграмме описаны:

* Отношения между пользователем и хостом – основанная часть использования программы;

Изображение выглядит как снимок экрана, текст, диаграмма, План

Автоматически созданное описание

Рисунок 15 – (Диаграмма Активностей)

Диаграмма показывает:

* Динамические отношения между пользователем и сервером;
* Алгоритм действий системы;

# 3.2. Отладка и тестирование приложения

Несколько раз я тестировал своего бота, встречался с многочисленными трудностями, переписывал код, начинал заново, читал большое количество статей и смотрел видео по тому, что связанно с написанием такого бота, но в итоге я добился того, чтобы мой бот функционировал и выводил нужную информацию.

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 14 (Чат с ботом)

Как видим, все работает исправно.

# ЗАКЛЮЧЕНИЕ

В настоящее время популярность мессенджеров, как средств общения и передачи информации растёт, так же, как и растет популярность ботов. Чат-боты – это инновационные ресурсы, позволяющие с помощью языка программирования, автоматически обрабатывать и отправлять сообщения. Важным фактором является то, что пользователям не придётся скачивать множество других приложений, потому что достаточно иметь лишь мессенджер и чат-ботов, которые предоставляют необходимую информацию пользователям.

Таким образом, в ходе выполнения моей работы, были выполнены поставленные задачи. Чат-бот представляет собой ценный ресурс для всех, кто интересуется изучением языка программирования Python. Он обладает широким спектром функций, включая предоставление обучающего контента, ответы на вопросы, предоставление примеров кода и выполнение практических заданий. Наш бот стремится стать надежным партнером для всех уровней опыта - от новичков до опытных разработчиков.

Я уверен, что мой чат-бот поможет пользователям расширить свои знания и навыки в программировании на Python, обеспечивая доступ к обучающим ресурсам, полезной информации и поддержке на каждом этапе их учебного пути.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

* 1. "Python для профессионалов" Мартин Мишельсон, Сандро Тассо.
  2. "Building Chatbots with Python: Using Natural Language Processing and Machine Learning" от Sumit Raj.
  3. Чем отличаются чат-боты в Telegram, WhatsApp, Viber, Facebook, VK (Электронный ресурс).

1. Технологии создания и применения чат-ботов (Текст научной статьи)
2. https://habr.com/ru/articles/699550/
3. https://markjames.dev/blog/building-telegram-bot-csharp
4. https://telegram.me/BotFather

# ПРИЛОЖЕНИЕ А

using System;

using System.Threading.Tasks;

using Telegram.Bot;

using Telegram.Bot.Args;

using Telegram.Bot.Types.Enums;

using Telegram.Bot.Types;

class Program

{

private static TelegramBotClient botClient;

static async Task Main()

{

botClient = new TelegramBotClient("мой токен(секрет)");

var me = await botClient.GetMeAsync();

Console.WriteLine($"Bot ID: {me.Id} Bot Name: {me.FirstName}");

Console.WriteLine("Bot started. Press Ctrl+C to stop.");

await botClient.SetMyCommandsAsync(new[]

{

new BotCommand { Command = "start", Description = "Начать обучение языку программирования Python" },

new BotCommand { Command = "help", Description = "Показать список доступных команд" },

new BotCommand { Command = "info", Description = "Получить базовую информацию о языке программирования Python" },

new BotCommand { Command = "examples", Description = "Получить примеры базовых строк кода на Python" },

new BotCommand { Command = "exercises", Description = "Получить упражнения по программированию на Python" },

new BotCommand { Command = "resources", Description = "Получить полезные ресурсы по программированию" },

new BotCommand { Command = "tips", Description = "Получить советы и хаки по программированию" }

});

await ReceiveMessage();

// Бесконечный цикл для удержания программы в работе

while (true)

{

await Task.Delay(1000); // Пауза на 1 секунду, чтобы не перегружать процессор

}

}

private static async Task ReceiveMessage()

{

int offset = 0;

while (true)

{

var updates = await botClient.GetUpdatesAsync(offset);

foreach (var update in updates)

{

var message = update.Message;

if (message == null || message.Type != MessageType.Text)

continue;

await Bot\_OnMessage(message);

offset = update.Id + 1;

}

}

}

private static async Task Bot\_OnMessage(Message message)

{

if (message == null)

return;

// Приветствие пользователя

if (message.Text != null && message.Text.ToLower() == "/start")

{

await botClient.SendTextMessageAsync(message.Chat.Id, "Привет! Я бот, который поможет тебе изучать язык программирования Python. Используй команду /help, чтобы увидеть список доступных команд.");

return;

}

// Список доступных команд

if (message.Text != null && message.Text.ToLower() == "/help")

{

await botClient.SendTextMessageAsync(message.Chat.Id, "Доступные команды:\n/start - начать работу с ботом\n/help - показать список доступных команд\n/info - получить базовую информацию о языке программирования Python\n/examples - получить примеры базовых строк кода на Python\n/exercises - получить упражнения по программированию на Python\n/resources - получить полезные ресурсы по программированию\n/tips - получить советы и хаки по программированию");

return;

}

// Базовая информация о языке программирования

if (message.Text != null && message.Text.ToLower() == "/info")

{

await botClient.SendTextMessageAsync(message.Chat.Id, "Python - это интерпретируемый высокоуровневый язык программирования общего назначения. Он прост в изучении и читаемости, что делает его идеальным выбором для начинающих программистов.");

return;

}

// Примеры базовых строк кода

if (message.Text != null && message.Text.ToLower() == "/examples")

{await botClient.SendTextMessageAsync(message.Chat.Id, "Примеры базовых строк кода на Python:\n\n1. Вывести 'Hello, world!' на экран:\nprint('Hello, world!')\n\n2. Вычислить сумму двух чисел:\na = 5\nb = 3\nc = a + b\nprint('Сумма равна:', c)\n\n3. Создать список чисел и вывести его на экран:\nnumbers = [1, 2, 3, 4, 5]\nfor num in numbers:\n print(num)");

return;

}

// Упражнения по программированию на Python

if (message.Text != null && message.Text.ToLower() == "/exercises")

{

await botClient.SendTextMessageAsync(message.Chat.Id, "Упражнения по программированию на Python:\n\n1. Напишите программу, которая будет выводить на экран числа от 1 до 10.\n\n2. Напишите функцию для вычисления факториала числа.\n\n3. Напишите программу, которая будет находить сумму всех элементов списка.");

return;

}

// Полезные ресурсы по программированию

if (message.Text != null && message.Text.ToLower() == "/resources")

{

await botClient.SendTextMessageAsync(message.Chat.Id, "Полезные ресурсы по программированию:\n\n1. Stack Overflow - https://stackoverflow.com/\n2. GitHub - https://github.com/\n3. Coursera - https://www.coursera.org/\n4. Codecademy - https://www.codecademy.com/\n5. Хабр - https://habr.com/ru/\n6. FreeCodeCamp - https://www.freecodecamp.org/");

return;

}

// Советы и хаки по программированию

if (message.Text != null && message.Text.ToLower() == "/tips")

{

await botClient.SendTextMessageAsync(message.Chat.Id, "Советы и хаки по программированию:\n\n- Изучайте регулярно: Постоянное обучение и изучение новых технологий помогут вам оставаться востребованным программистом.\n- Читайте исходный код других: Изучение исходного кода других проектов помогает улучшить ваше понимание языка и научиться лучшим практикам.\n- Используйте отладчик: Умение эффективно использовать отладчик позволяет быстрее находить и исправлять ошибки в коде.\n- Не бойтесь ошибок: Ошибки - это нормальная часть процесса разработки. Из них можно учиться и становиться лучше.\n- Пишите понятный код: Код должен быть понятным для других разработчиков. Используйте понятные имена переменных, комментарии и хорошо структурируйте свой код.");

return;

}

// Если ни одно условие не выполнилось, отправляем стандартный ответ

await botClient.SendTextMessageAsync(message.Chat.Id, "Я тебя не понимаю :(. Попробуй ввести команду /help для получения списка доступных тебе команд.");

}

}