*Полный разбор задачи по созданию телеграм-бота находится в разделе электронного курса модуля "Программирование на языке Python":*

*Если у Вас возникли затруднения, то в телеграмм канале, есть видео с подробным объяснением этой задачи:* [*https://t.me/digit\_sfu\_python/117*](https://t.me/digit_sfu_python/117) *(в середине видео)*

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«Сибирский федеральный университет»

**ИТОГОВАЯ АТТЕСТАЦИОННАЯ РАБОТА**   
по программе профессиональной переподготовки   
«Разработка программного обеспечения на языке Python»

на тему: Разработка телеграмм бота

|  |  |
| --- | --- |
| Слушатель: Петров Петр Петрович |  |
| Группа: 2 |  |
| Руководитель итоговой аттестационной работы: Раскина А.В., к.т.н, доцент кафедры ИС ИКИТ СФУ |  |

Красноярск

2023

СОДЕРЖАНИЕ

[Введение 6](#_Toc129338962)

[1. Обзорная часть 7](#_Toc129338963)

[1.1. Обзор задачи 7](#_Toc129338964)

[1.2. Обзор инструментов для создания приложений 7](#_Toc129338965)

[2. Проектирование приложения 8](#_Toc129338966)

[2.1. Описание желаемых функций приложения 8](#_Toc129338967)

[2.2 Требования к программе или программному изделию 9](#_Toc129338968)

[2.3 Стадии и этапы разработки 11](#_Toc129338969)

[2.4. Описание архитектуры приложения 11](#_Toc129338970)

[3. Реализация приложения 12](#_Toc129338971)

[3.1 Написание программного кода 12](#_Toc129338972)

[3.2 Тестирование приложения 12](#_Toc129338973)

[Заключение 12](#_Toc129338974)

[Список использованных источников 13](#_Toc129338975)

# Введение

Создание Telegram-ботов на Python является актуальной задачей, так как Telegram является одной из самых популярных мессенджеров в мире, и многие компании и организации используют его для общения с клиентами и пользователей. Боты могут помочь автоматизировать процессы, упростить коммуникацию и предоставить дополнительные сервисы. Python является одним из наиболее популярных языков программирования для создания ботов, так как он имеет множество библиотек и фреймворков, которые упрощают разработку. Кроме того, Python обладает простым и понятным синтаксисом, что делает его доступным для новичков в программировании.

Для создания приложения в работе использовался TelegramAPI - это набор инструментов и методов, которые позволяют разработчикам создавать ботов для Telegram. Он предоставляет доступ к функциям Telegram, таким как отправка сообщений, получение обновлений, управление чатами и пользователями, работа с клавиатурами и многим другим.

Цель работы является создание Telegram бота для информирования студентов сибирского федерального университета о программах, реализуемых в рамках проекта «Цифровая кафедра».

Задачи, необходимые для выполнения цели:

1. Анализ предметной области
2. Выбор технологии создания приложения
3. Разработка приложения
4. Анализ и тестирование.

Основной функцией приложение является обработка запросов от пользователей и предоставление им информации о программах дополнительной профессиональной переподготовки. Важным аспектом при создании Telegram бота является его удобство и простота использования для пользователей. Бот должен быть интуитивно понятным, быстрым и эффективным в решении задач, чтобы пользователи могли получить максимальную пользу от его использования.

# 1. Обзор технологий создания telegram бота

## 1.1. Обзор задачи создания telegram бота

Задача создания Telegram бота заключается в автоматизации некоторых процессов и упрощении взаимодействия с пользователями. Конкретные задачи могут быть различными в зависимости от целей создания бота, но в целом, бот должен предоставлять пользователю информацию, обрабатывать запросы и уведомлять о новых событиях.

Telegram боты появились в 2015 году и быстро стали популярными благодаря своей простоте и удобству использования. Они позволяют пользователям получать информацию и выполнять различные задачи без необходимости устанавливать дополнительные приложения.

Создание Telegram бота может быть выполнено как с использованием готовых платформ, таких как BotFather или Manybot, так и написанием собственного кода на языке программирования Python, Node.js, PHP и т.д.

Одной из основных задач при создании Telegram бота является его удобство и простота использования для пользователей. Бот должен быть интуитивно понятным, быстрым и эффективным в решении задач, чтобы пользователи могли получить максимальную пользу от его использования.

## 1.2. Обзор инструментов для создания telegram бота

Существует обширное количество инструментов, позволяющих создавать telegram боты. Рассмотрим самые популярные из них.

BotFather - это официальный инструмент Telegram для создания и управления ботами. Он позволяет создавать новых ботов, настраивать их функциональность и получать API-ключи для работы с ними. Manybot - это платформа для создания и управления Telegram ботами. Она предоставляет широкий набор функций, включая возможность создания кнопок, автоматическую отправку сообщений и многое другое. Python-telegram-bot - это библиотека на языке Python для создания Telegram ботов. Она предоставляет широкий набор функций, включая возможность создания кнопок, обработку команд и многое другое. Telegraf - это библиотека на языке JavaScript для создания Telegram ботов. Она предоставляет широкий набор функций, включая возможность создания кнопок, обработку команд и многое другое. PHP Telegram Bot - это библиотека на языке PHP для создания Telegram ботов. Она предоставляет широкий набор функций, включая возможность создания кнопок, обработку команд и многое другое. Chatfuel - это платформа для создания чат-ботов в различных мессенджерах, включая Telegram. Она предоставляет широкий набор функций, включая возможность создания кнопок, автоматическую отправку сообщений и многое другое. Flow XO - это платформа для создания чат-ботов в различных мессенджерах, включая Telegram. Она предоставляет широкий набор функций, включая возможность создания кнопок, автоматическую отправку сообщений и многое другое.

Выбор инструмента для создания Telegram бота зависит от целей и требований проекта, а также от опыта и предпочтений разработчика.В настоящей работе для создания приложения использовалась библиотека TelegramBotAPI.

Официальная библиотека TelegramAPI для Python называется python-telegram-bot и предоставляет удобный интерфейс для работы с TelegramAPI. Она имеет документацию и обширный набор примеров, что делает ее очень популярной среди разработчиков.

Некоторые из основных функций TelegramAPI включают в себя:

- Отправка сообщений: разработчики могут отправлять текстовые сообщения, изображения, аудио- и видеофайлы, голосовые сообщения и многие другие типы контента.

- Получение обновлений: боты могут получать обновления от TelegramAPI, такие как новые сообщения, изменения в чатах и другие события.

- Управление чатами и пользователями: боты могут управлять чатами и пользователями, добавлять новых пользователей в группы, удалять пользователей, изменять права доступа и многое другое.

- Работа с клавиатурами: боты могут создавать и отправлять клавиатуры, которые позволяют пользователям взаимодействовать с ботом, не вводя текст вручную.

- Интеграция с другими сервисами: боты могут интегрироваться с другими сервисами, такими как базы данных, API сторонних сервисов и многие другие.

TelegramAPI является мощным инструментом для создания ботов, которые могут помочь автоматизировать различные задачи и улучшить взаимодействие с пользователями.

# 2. Проектирование приложения

**2.1. Описание желаемых функций приложения**

Опишем рассматриваемого продукта функциональные и нефункциональные требования. Функциональное требование – это описание поведения системы в определенных условиях. Другими словами, это описание функций системы.

Для telegram бота предъявляются следующие требования к функциональным характеристика:

- просмотр информации о проекте «Цифровая кафедра»;

- создание кнопок: бот должен иметь возможность создавать кнопки для удобной навигации пользователей по функционалу бота (выбор интересующие программы дополнительной профессиональной переподготовки по нажатию кнопки);

- отправка сообщений: бот должен иметь возможность отправлять текстовые сообщения о интересующие программе пользователю.

- обработка команд: бот должен иметь возможность обрабатывать команды, введенные пользователями, и выполнять соответствующие действия (нажатие кнопки, команда /start, /help).

## 2.2 Требования к программе или программному изделию

К нефункциональным требованиям можно отнести:

1. Надежность: бот должен работать без сбоев и ошибок, чтобы пользователи могли полноценно использовать его функционал. Вероятность безотказной работы системы должна составлять не менее 99.99% при условии исправности сети (связи приложений на языке Python и telegram).

2. Безопасность: бот должен обеспечивать защиту персональных данных пользователей и предотвращать несанкционированный доступ к ним.

3. Скорость: бот должен быстро обрабатывать запросы пользователей и отвечать на них.

4. Удобство использования: бот должен быть интуитивно понятным и удобным в использовании для пользователей.

Исходный код должен быть написан на языке python c библиотеки TelegramBotAPI.

## 2.3 Стадии и этапы разработки

Для создания бота с использованием TelegramAPI необходимо выполнить следующие шаги:

1. Зарегистрироваться в Telegram и создать новый бот, следуя инструкциям на официальном сайте Telegram.

2. Получить API ключ для бота, который будет использоваться для взаимодействия с TelegramAPI.

3. Выбрать язык программирования, который будет использоваться для написания бота. Для работы с TelegramAPI можно использовать различные языки программирования, такие как Python, Java, C++, JavaScript и др.

4. Установить необходимые библиотеки и фреймворки для выбранного языка программирования.

5. Написать код для бота, который будет обрабатывать запросы от пользователей и отправлять им ответы.

6. Запустить бота на сервере или локальном компьютере, используя специальный скрипт или приложение.

7. Протестировать работу бота и внести необходимые изменения в код, если это необходимо.

8. Опубликовать бота в Telegram и начать его использование.

**2.4. Описание архитектуры приложения**

Архитектура приложения telegram бота организована в соответствии с принципами модульности. Основные модули приложения могут включают:

- модуль отправки сообщений,

- модуль обработки команд,

- модуль создания кнопок.

**3. Реализация приложения**

## 3.1 Написание программного кода

Первым шагом импортируем модуль telebot:

  import telebot  
   
Импортируем типы из модуля telebot, чтобы создавать кнопки:  
   
 from telebot import types  
   
Создаем экземпляр бота:

 bot = telebot.TeleBot(&apos;5865902498:AAENpKJkVcW16OtAI0zU1Its1ctPu\_2x\_j4&apos;)

Для создания клавиатуры используем InlineKeyboardMarkup() — это кастомную клавиатуру. С её помощью можно выполнять сложные действия. Она привязывается к сообщению, с которым была отправлена. В кнопки можно заложить любой текст размером от 1 до 64 байт. Инлайн кнопки позволяют скрыть в себе внутреннюю телеграм ссылку, ссылку на внешний ресурс, а также шорткат для инлайн запроса.

 keyboard = types.InlineKeyboardMarkup()

Далее по очереди готовим текст, и обработчик для каждой программы и добавляем кнопку на экран:

 key\_linux = types.InlineKeyboardButton(text=&apos;Программа по Linux&apos;, callback\_data=&apos;linux&apos;)

 keyboard.add(key\_linux)

 key\_python = types.InlineKeyboardButton(text=&apos;Программа по Python&apos;, callback\_data=&apos;python&apos;)

 keyboard.add(key\_python)

 key\_media = types.InlineKeyboardButton(text=&apos;Программа по медиапроектированию&apos;, callback\_data=&apos;media&apos;)  
 keyboard.add(key\_media)

 key\_huawei = types.InlineKeyboardButton(text=&apos;Программа про huawei&apos;, callback\_data=&apos;huawei&apos;)  
 keyboard.add(key\_huawei)  
   
   
Добавим функцию, обрабатывающую команду /start:

 @bot.message\_handler(commands=["start"])  
 def start(m, res=False):  
     bot.send\_message(m.chat.id, &apos;Привет! Сейчас я расскажу тебе о наших программах&apos;)

# Показываем все кнопки сразу и пишем сообщение о выборе  
     bot.send\_message(m.from\_user.id, text=&apos;Выбери понравившуюся программу&apos;, reply\_markup=keyboard)  
 

Добавим функцию, обрабатывающую команду «ввод произвольного текста»:

 @bot.message\_handler(content\_types=[&apos;text&apos;])  
 def get\_text\_messages(message):  
     if message.text == "/help":  
         bot.send\_message(message.from\_user.id, "Я бот, который позволит тебе узнать больше о проекте! Напиши /start")  
     else:  
         bot.send\_message(message.from\_user.id, "Я тебя не понимаю, обратись за помощью! Напиши /help")

Добавим функцию, обрабатывающую команду «нажатие на кнопку»:  
  
   
 @bot.callback\_query\_handler(func=lambda call: True)  
 def callback\_worker(call):  
     if call.data == "python":  
         msg = "Программа для всех НЕ ИТ-шников! Научим веб-программированию и машинному обучению!"  
         # Отправляем текст в Телеграм  
         bot.send\_message(call.message.chat.id, msg)  
     if call.data == "linux":  
         msg = "Программа для всех ИТ-шников! Учишься в ИКИТе? Научим администрировать линукс!"  
         # Отправляем текст в Телеграм  
         bot.send\_message(call.message.chat.id, msg)  
     if call.data == "media":  
         msg = "Программа для всех НЕ ИТ-шников! Научим управлять ИТ проектами в области креативных медиа!"  
         # Отправляем текст в Телеграм  
         bot.send\_message(call.message.chat.id, msg)  
     if call.data == "huawei":  
         msg = "Программа для всех ИТ-шников! научим справляться с оборудованием HUAWEI!"  
         # Отправляем текст в Телеграм  
         bot.send\_message(call.message.chat.id, msg)  
   
   
  Запускаем бот на исполнение:

 bot.polling(none\_stop=True, interval=0)

## 3.2 Тестирование приложения

Приведем пример использования приложения для определения его работоспособности.

После ввода команды /start можно увидеть следующий вид (рис.1)

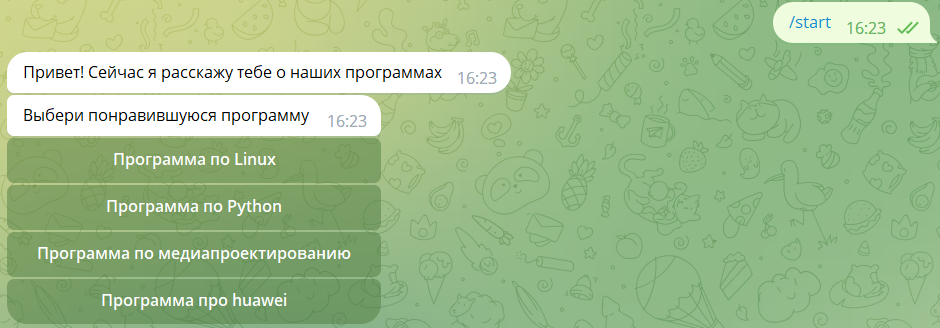


Рисунок 1 – Вид диалога с telegram bot после ввода команды /start

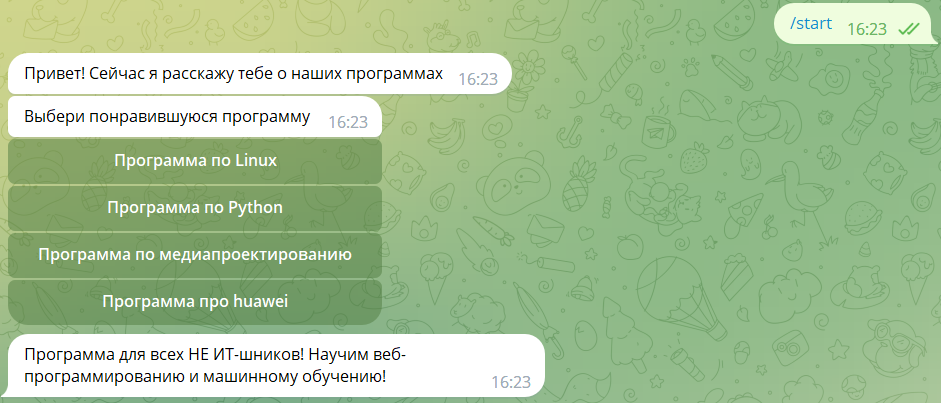


Рисунок 2 – Отправка сообщения пользователю после нажатия кнопки «Программа по Python»

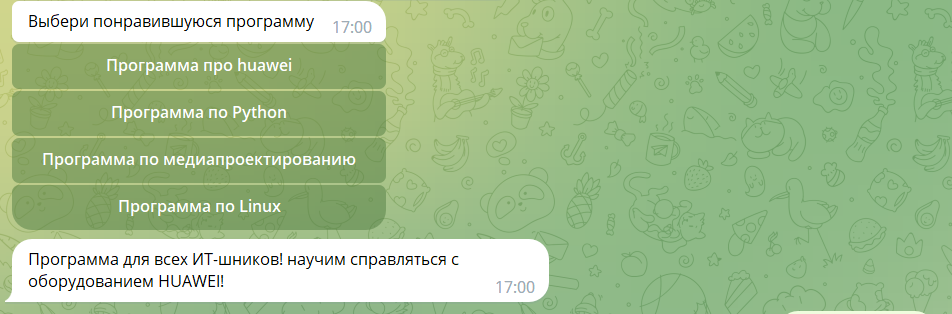


Рисунок 3 – Отправка сообщения пользователю после нажатия кнопки «Программа про huawei»

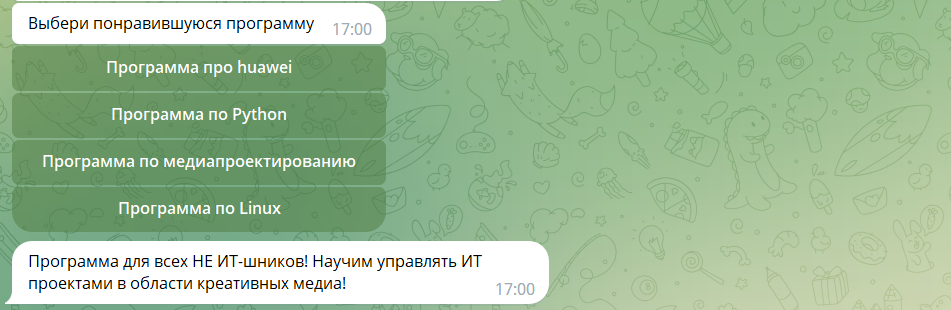


Рисунок 4 – Отправка сообщения пользователю после нажатия кнопки «Программа про медиапроектированию»

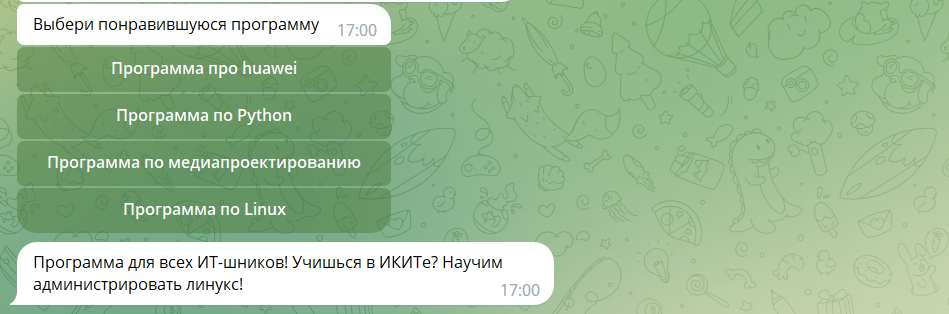


Рисунок 5 – Отправка сообщения пользователю после нажатия кнопки «Программа по linux»

Попробуем ввести в диалог с ботом произвольный текст:



Рисунок 6 – Отправка сообщения пользователю после ввода неопознанного текста

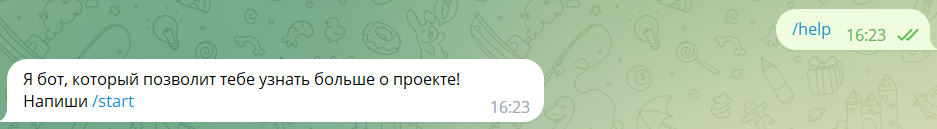


Рисунок 7 – Отправка сообщения пользователю после ввода команды /help

# Заключение

В итоговой аттестационной работе решена задача создания телеграм бота для предоставления информации пользователю о программах профессиональной переподготовки, реализуемых в рамках проекта «Цифровая кафедра».

Основные функции приложения:

- просмотр информации о проекте «Цифровая кафедра»;

- создание кнопок: бот должен иметь возможность создавать кнопки для удобной навигации пользователей по функционалу бота (выбор интересующие программы дополнительной профессиональной переподготовки по нажатию кнопки);

- отправка сообщений: бот должен иметь возможность отправлять текстовые сообщения о интересующие программе пользователю.

- обработка команд: бот должен иметь возможность обрабатывать команды, введенные пользователями, и выполнять соответствующие действия (нажатие кнопки, команда /start, /help).

Основными модулями приложения являются модуль отправки сообщений, модуль обработки команд, модуль создания кнопок.

Для создания приложения Telegram bot на Python использовался инструмент TelegramAPI, который предоставляет необходимые функции для работы с ботами в Telegram.

# Список использованных источников

1. Лутц М. Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.
2. Златопольский Д.М. Основы программирования на языке Python. – М.: ДМК Пресс, 2017. – 284 с.
3. Лутц М. Программирование на Python, том I, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
4. Лутц М. Программирование на Python, том II, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
5. Гэддис Т. Начинаем программировать на Python.  – 4-е изд.: Пер. с англ. – СПб.: БХВ-Петербург, 2019. – 768 с.
6. Лучано Рамальо Python. К вершинам мастерства. – М.: ДМК Пресс, 2016. – 768 с.
7. Свейгарт, Эл. Автоматизация рутиных задач с помощью Python: практическое руководство для начинающих. Пер. с англ. — М.: Вильямc, 2016. – 592 с.
8. Рейтц К., Шлюссер Т. Автостопом по Python. – СПб.: Питер, 2017. – 336 с.: ил. – (Серия «Бестселлеры O’Reilly»).
9. Любанович Билл Простой Python. Современный стиль программирования. – СПб.: Питер, 2016. – 480 с.: – (Серия «Бестсепперы O’Reilly»).
10. Федоров, Д. Ю.  Программирование на языке высокого уровня Python : учебное пособие для прикладного бакалавриата / Д. Ю. Федоров. – 2-е изд., перераб. и доп. – Москва : Издательство Юрайт, 2019. – 161 с. – (Бакалавр. Прикладной курс). – ISBN 978-5-534-10971-9. – Текст: электронный // ЭБС Юрайт [сайт]. – URL: <https://urait.ru/bcode/437489> (дата обращения: 13.02.2020).
11. Шелудько, В. М. Основы программирования на языке высокого уровня Python: учебное пособие / В. М. Шелудько. – Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2017. – 146 c. – ISBN 978-5-9275-2649-9. – Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. – URL: http://www.iprbookshop.ru/87461.html (дата обращения: 13.02.2020). – Режим доступа: для авторизир. пользователей
12. Шелудько, В. М. Язык программирования высокого уровня Python. Функции, структуры данных, дополнительные модули: учебное пособие / В. М. Шелудько. – Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2017. – 107 c. – ISBN 978-5-9275-2648-2. – Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. – URL: http://www.iprbookshop.ru/87530.html (дата обращения: 13.02.2020). – Режим доступа: для авторизир. пользователей
13. Доусон М. Программируем на Python. – СПб.: Питер, 2014. – 416 с.