

ИССЛЕДОВАНИЕ И РАЗРАБОТКА БОТОВ В TELEGRAM С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА ПРОГРАММИРОВАНИЯ C#

Тимошевич К. С., Котова К. А.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Гламаздин И. И. – старший преподаватель каф. информатики

В данной работе исследуется процесс разработки ботов для мессенджера Telegram с использованием языка программирования C#. Рассмотрены основные принципы работы ботов, включая их взаимодействие с пользователем, и возможности API Telegram. Приведена пошаговая схема разработки.

Боты – это компактные приложения, которые функционируют внутри приложения Telegram. Пользователи взаимодействуют с ботами через гибкие интерфейсы, способные обеспечить выполнение самых разнообразных задач и услуг: создание инструментов для определенных целей, интеграция с другими сервисами, прием платежей, поддержка игр внутри мессенджера и даже замена полноценных веб-сайтов.

Существует несколько разновидностей ботов:

- Чат-боты: генерируют простейшие диалоги на определенную тему, которую задает пользователь.
- Боты-информаторы: основная задача таких ботов заключается в донесении до пользователя информации об определенных событиях.
- Игровые боты: эти роботы позволяют играть во всевозможные игры.
- Боты-ассистенты: являются дополнениями к веб-версиям онлайн-сервисов.
- Боты для оплаты: принимают платежи пользователей и автоматически исключают их из платных чатов или каналов, если оплаты не было.
- Боты для напоминаний: уведомляют о приближающихся событиях.
- Боты-каталоги: помогают подобрать книгу, музыку или сериал.

Использование языка программирования C# для разработки ботов в Telegram предоставляет несколько преимуществ. Во-первых, C# является высокоуровневым объектно-ориентированным языком программирования, что упрощает разработку и поддержку кода. Во-вторых, C# позволяет использовать различные фреймворки и библиотеки, в числе которых есть библиотеки для написания ботов в Telegram, такие как Telegram.Bot и TLSharp, а также C# имеет отличную поддержку асинхронного программирования, что может быть полезно для обработки входящих сообщений.

Telegram использует собственный протокол шифрования MTProto. MTProto API (он же Telegram API) – это API, через который ваше приложение Telegram связывается с сервером. Telegram API полностью открыт, так что любой разработчик может написать свой клиент мессенджера.

Telegram API обеспечивает доступ к множеству функций, которые позволяют взаимодействовать с пользователями, обрабатывать сообщения и медиафайлы, а также управлять настройками и возможностями ботов. С помощью API разработчики могут создавать и настраивать ботов, добавлять им функционал и интегрировать с другими сервисами.

Например, для создания пользовательской клавиатуры используются объекты ReplyKeyboardMarkup с возможностью конструирования однострочной и многострочной быстрой клавиатуры, замещающей стандартную, а также объекты InlineKeyboardMarkup для создания клавиатуры отправки обратного вызова, обрабатываемого ботом.

Рассмотрим пошаговую разработку простейшего бота на C#, представленную на рисунке 1.

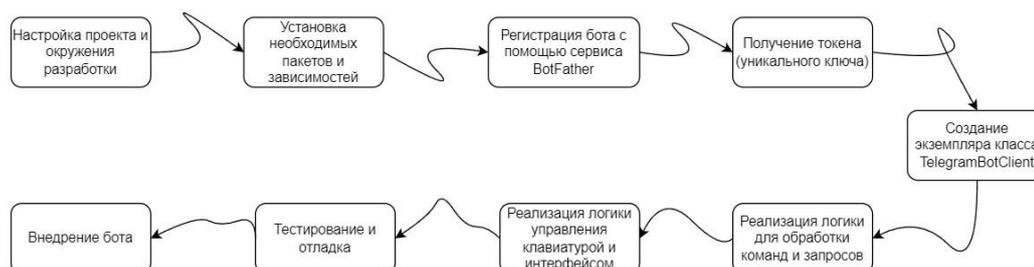


Рисунок 1 – Схема создания Telegram-бота

Следуя этой схеме создания, был разработан Telegram-бот для теста «Твоя роль в IT-сфере», представленный на рисунке 2.

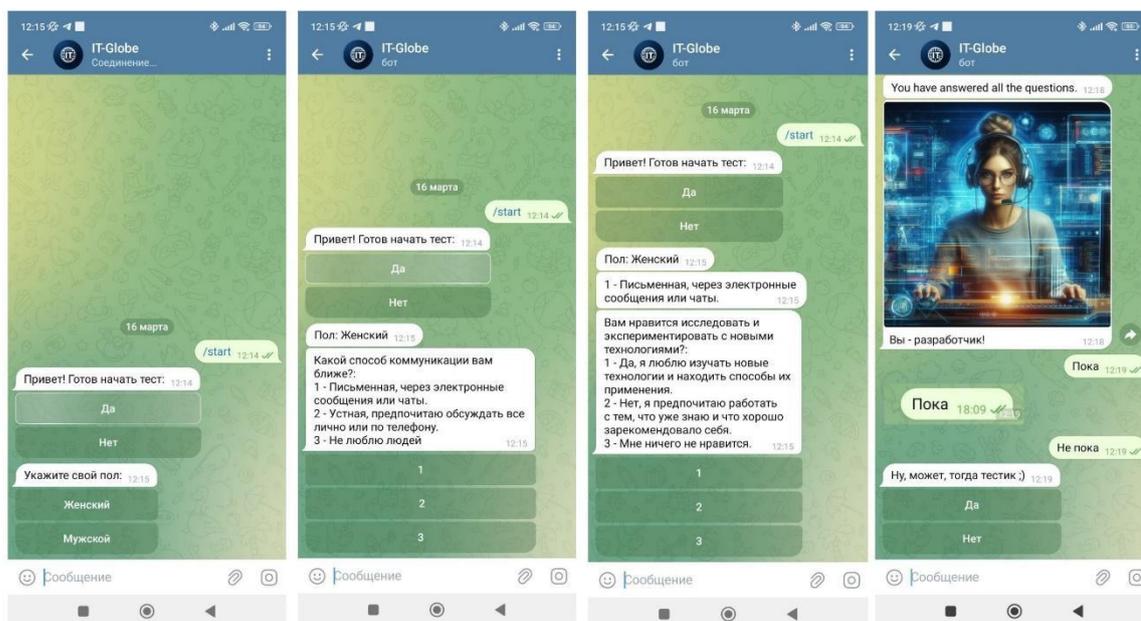


Рисунок 2 – Тест-бот на языке программирования C#

Создание любого бота в Telegram начинается с получения секретного токена у официального бота Telegram BotFather (рисунок 3)



Рисунок 3 – Официальный бот Telegram для получения токенов

В процессе разработки были использованы наиболее популярные функции Telegram API и библиотеки Telegram.Bot. Так, для создания удобной пользовательской клавиатуры для ответов был использован объект класса `InlineKeyboardMarkup`. Взаимодействие с пользователем осуществлялось посредством разнообразных методов. Для отправки стандартных текстовых сообщений был использован `SendMessageAsync()`, чтобы придать боту интерактивности и динамичности при общении с пользователем были применены методы `EditMessageTextAsync()` для изменения сообщений в результате какого-либо причастного к нему взаимодействия, а также `SendStickerAsync()` для возможности ответа стикером.

Когда настал момент подведения итогов прохождения теста, был использован метод `SendPhotoAsync()`, который предназначен для отправления фотографий в чат с пользователем. С помощью этого метода вывод результата стал визуально привлекательнее.

Именно благодаря своей простоте в написании боты хорошо зарекомендовали себя во всех отраслях, где есть большой поток людей с одинаковыми запросами, которые нужно обрабатывать быстро.

В контексте сферы IT были выявлены преимущества использования языка C# для создания ботов. Этот язык программирования предоставляет разработчикам широкий спектр инструментов и возможностей, что делает процесс создания ботов более эффективным и удобным.

Благодаря своей простоте и гибкости, язык C# позволяет быстро и легко разрабатывать Telegram-ботов, обладающих разнообразным функционалом. Это включает в себя возможность обработки сообщений, отправки уведомлений, взаимодействия с внешними сервисами и многие другие задачи, что делает его идеальным выбором для создания ботов различного назначения.

Список использованных источников:

1. *Bots: An introduction for developers* [Электронный ресурс] – Режим доступа: <https://core.telegram.org/bots>. Дата доступа: 03.03.2024
2. *Блог ЯПрактикума* [Электронный ресурс] – Режим доступа: <https://practicum.yandex.ru/blog/telegram-boty-kak-rabotayut-i-kak-nastroit/>. Дата доступа: 03.03.2024
3. *Редакция сайта GeekBrains* [Электронный ресурс] – Режим доступа: <https://gb.ru/blog/chto-takoe-telegram/>. Дата доступа: 03.03.2024
4. *Книга ботов Telegram* [Электронный ресурс] – Режим доступа: <https://telegrambots.github.io/book/2/reply-markup.html>. Дата доступа: 03.03.2024
5. *Документация Telegram: Боты* [Электронный ресурс] – Режим доступа: <https://tigrm.ru/docs/bots>. Дата доступа: 05.03.2024.