

4. Jekonomicheskie obzory OJeSR [OECD Economic Reviews].  
[https://oecdru.org/zip/Overview\\_RUSSIARus\\_2013.pdf](https://oecdru.org/zip/Overview_RUSSIARus_2013.pdf)

5. Beljakova E.V., Ryzhaja A.A. Rol' transportnoj infrastruktury v razvitii goroda [The role of transport infrastructure in the development of the city]. Reshetnevskie chtenija. 2017. №21-2.

УДК 004.912:613.2:796.015

**О.В. Новик, А.О. Борисевич, В.О. Войтехович**  
Национальный детский технопарк  
Минск, Беларусь

## **ПРОПИТАШКА: РАЗРАБОТКА ТЕЛЕГРАМ-БОТА ДЛЯ СОЗДАНИЯ ПЕРСОНАЛЬНОГО ПЛАНА ПИТАНИЯ И ТРЕНИРОВОК**

*Аннотация.* Современные технологии активно интегрируются в области здравоохранения и фитнеса, предоставляя пользователям доступ к персонализированным инструментам для поддержания здорового образа жизни. Данный проект представляет Telegram-бота под названием PROпиташка, который предоставляет пользователям простое и удобное средство для создания и поддержки персонального плана питания и тренировок.

**O.V. Novik, A.O. Borisevich, V.O. Voitekhovich**  
National Children's Technopark  
Minsk, Belarus

## **PROFITASHKA: DEVELOPMENT OF A TELEGRAM BOT FOR CREATING A PERSONAL NUTRITION AND TRAINING PLAN**

*Abstract.* Modern technologies are actively integrating into the fields of health and fitness, providing users with access to personalized tools for maintaining a healthy lifestyle. This project presents a Telegram bot called PROpitashka, which provides users with a simple and convenient tool for creating and maintaining a personalized nutrition and training plan.

Целью данной статьи является описание процесса проектирования и разработки Telegram-бота, поддерживающего

составление и ведение индивидуального плана питания и тренировок.

Для реализации поставленной цели необходимо решить ряд задач, которые являются основными этапами разработки подобных сервисов:

- определение базовых объектов системы. Проектирование и внедрение системы ввода данных, позволяющей пользователям задавать информацию о своих параметрах (возраст, пол, вес, цели по физической активности);

- реализация рекомендаций по питанию и тренировкам. Алгоритмы расчета и предоставления пользователям предложений по калорийности и составу питания на основе текущих данных о физической активности;

- создание интерфейса взаимодействия с ботом, обеспечивающего удобство использования и высокую степень интерактивности для различных категорий пользователей.

Чат-бот в Телеграм – это роботизированный аккаунт в мессенджере, который запрограммирован на автоматическое совершение действий: продавать товары, развлекать пользователей или искать полезную информацию. Управляется по принципу «вопрос – ответ»: пользователь вводит текстовую команду, бот обрабатывает запрос, находит в своей базе данных подходящий сценарий и отвечает пользователю по скрипту.

В сообщении от бота может быть:

- текст;
- ссылки;
- изображения;
- видео;
- файлы;
- кнопки [1].

С учетом этого, а также, цели и поставленных задач, основными объектами бота, подлежащие разработке, являются:

- пользовательские данные;
- план питания;
- план тренировок;
- графический интерфейс;
- анимации тренировок.

Пользовательские данные вносятся через интерфейс Telegram и включают параметры, влияющие на расчет питания и тренировок. Бот использует данные для формирования рекомендаций, позволяя

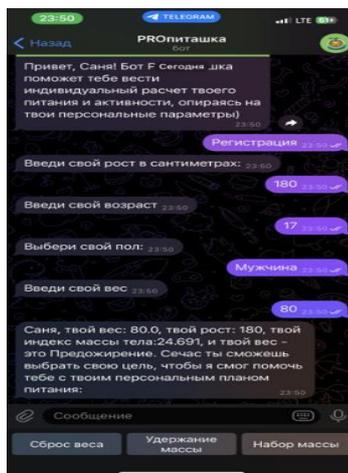
пользователям контролировать свои результаты и корректировать план.

Для работы бота используются следующие функции и шаблоны методов:

— шаблоны персональных рекомендаций и шаблонов для расчета калорийности и состава макронутриентов на основе индивидуальных данных. Например, расчеты суточного потребления белков, жиров и углеводов на основании физической активности и индивидуальных параметров пользователя;

— взаимодействие с пользователем. Telegram-бот предоставляет удобный интерфейс для ввода данных о физической активности и питании, который поддерживает гибкую настройку в зависимости от целей пользователя (снижение веса, набор массы или поддержание текущего веса) (рис. 1);

— хранение данных и отчетность. Все данные пользователя сохраняются в централизованной базе данных для предоставления сводных отчетов, которые показывают динамику изменений за день, неделю, месяц.



**Рис.1 – Пример интерфейса бота для взаимодействия с пользователем**

Для обеспечения гибкости и масштабируемости были разработаны шаблоны методов и структуры данных, позволяющие легко модифицировать функциональность бота.

Например, методы управления потреблением воды позволяют добавлять и изменять данные о выпитой жидкости, а система отчетности предоставляет пользователю сводные данные по расходу калорий и времени, затраченному на тренировки. Бот может быть

расширен для поддержки новых функций и модулей, например, добавление анимации тренировок (рисунок 2).



**Рис.2 – Модели для анимации тренировок**

Добавление анимации тренировок позволит людям, которые далеки от спорта, понять, как делать то или иное упражнение. Это выгодно отличает бот от конкурентов.

Для разработки Telegram-бота были использованы следующие методы и технологии:

- SQLAlchemy – для работы с базой данных, содержащей данные пользователей и их активности;

- Python Telegram API – для интеграции с Telegram и обеспечения легкости взаимодействия с пользователем;

- Asyncio и AioGram – для повышения стабильности работы бота и обеспечения обработки запросов в асинхронном режиме;

- компьютерное зрение – используется для распознавания продуктов по фотографиям, что упрощает ввод данных о питании.

Разработанный бот ПРОПИТАШКА предоставляет пользователям удобный способ контроля питания и тренировок, обеспечивая персонализированные рекомендации и возможность отслеживания результатов. Система поддерживает гибкую настройку шаблонов рекомендаций, что позволяет адаптировать ее под потребности каждой отдельной группы пользователей. В будущем планируется добавление новых функций и усовершенствование существующих методов с целью повышения точности рекомендаций и удобства использования.

### **Список использованных источников**

[1] 1С-рарус [Электронный ресурс]. – Режим доступа: <https://rarus.ru/bonus/telegram-bot/>.