

# Разработка программного обеспечения телеграмм бота для студентов и абитуриентов ГБОУ ВО НГИЭУ

С.В. Зуева, А.С. Краснов, А.С. Краснова

**Аннотация** – В данной статье проанализирован рейтинг языков программирования и график изменения популярности языков программирования. На основании проведённого анализа создан и реализован проект, главной целью которого является помощь студентам в период обучения, путем отделения от основного источника информации требуемых данных в единый портал с возможностью использования Telegram чат-бота. Актуальность разработки и возможности внедрения данного проекта в повседневное использование студентами НГИЭУ объясняется тем, что мессенджер Telegram не является официальным источником для получения информации. Данная платформа является наиболее актуальной для создания ботов с набором цепочек-ответов, так как ее суть заключается во взаимодействии пользователя с приложением с помощью текстовых сообщений. На данный момент НГИЭУ насчитывает порядка 5 официальных источников информации, одним из которых является web-портал «ngie.u.ru», он же является основным центром сбора, оповещения и получения данных. В связи с этим было принято решение создать и реализовать проект, главной целью которого будет помощь студентам в период обучения, путем отделения от основного источника информации требуемых данных в единый портал с возможностью использования Telegram – бота.

**Ключевые слова** – Telegram, чат-бот, справочно-информационная система, Python, функция.

Статья получена 10 февраля 2023  
Зуева Светлана Владимировна, Нижегородский государственный инженерно-экономический университет. (e-mail: limonovasvetlana@mail.ru).  
Краснов Александр Сергеевич, Нижегородский государственный инженерно-экономический университет. (e-mail: sahk2016@mail.ru).  
Краснова Анна Сергеевна, Нижегородский государственный инженерно-экономический университет. (e-mail: shochina96@mail.ru).

## I. ВВЕДЕНИЕ

Актуальность разработки и возможности внедрения данного проекта в повседневное использование студентами НГИЭУ объясняется тем, что мессенджер Telegram не является официальным источником для получения информации, таких как расписание, ссылки на образовательные ресурсы, новости и иных полезных сведений. Данная платформа является наиболее актуальной для создания ботов с набором цепочек-ответов. Суть таких ботов заключается во взаимодействии пользователя с приложением с помощью текстовых сообщений. Бот в Telegram не похож на обычный диалог, а также имеет отличия от ботов в любой из социальных сетей. Несмотря на то, что Telegram является одним из самых молодых мессенджеров, владелец сервиса позиционирует свою разработку лидером рынка. Это подтверждается наличием удобного интерфейса, быстрым откликом, обратной связью, как для простого пользователя, так и для разработчика. Универсальность данного приложения, а именно реализация под все современные платформы, такие как Android, iOS, Windows Phone, Windows, macOS, Linux, а также веб-версия проекта, помогло сделать выбор в сторону Telegram.

Вышеизложенное свидетельствует об актуальности рассматриваемой проблемы, которая имеет инженерно-техническое решение. В данной работе подробно рассмотрены вопросы разработки специального Telegram чат-бота, направленного на решение данной проблемы.

## II. ОПИСАНИЕ МЕТОДА РЕШЕНИЯ ЗАДАЧИ ИССЛЕДОВАНИЯ

На данный момент НГИЭУ насчитывает порядка 5 официальных источников информации, одним из которых является web-портал «ngie.u.ru». Он же является основным центром сбора, оповещения и получения данных, необходимых для абитуриентов, студентов. Изобилие информации, представленной на портале и в социальных сетях не дает

возможности получить мгновенно исчерпывающий ответ для студентов, которые обучаются на данный момент и который планируют поступать в вуз. В связи с этим было принято решение создать и реализовать проект, главной целью которого будет помощь студентам в период обучения, путем отделения от основного источника информации требуемых данных в единый портал с возможностью использования Telegram – бота.

Чат-бот – это виртуальный собеседник, программа, которая может решать типовые задачи: задавать вопросы и отвечать на них, искать информацию по запросу и выполнять простейшие поручения. [1]

Чат-боты используют в самых разных сферах. Например, банк Тинькофф разработал голосового помощника Олега. Олег обладает искусственным интеллектом [2]:

- распознаёт и интерпретирует запросы;
- совершает денежные переводы на счета в банке Тинькофф и Сбербанке;
- бронирует столики в ресторанах;
- записывает в салоны красоты;
- покупает билеты в кино;
- консультирует по банковским вопросам и акциям;
- а также может просто поболтать с вами.

В зависимости от принципа работы, выделяют обучаемых и необучаемых чат-ботов. Необучаемые чат-боты работают по заранее заданному сценарию и предлагают готовые варианты ответа. Обучаемые чат-боты

Java и сейчас является самым популярным языком программирования. В пятерку лучших языков вошли также: C#, Python, PHP. На рисунке 2 можно посмотреть, как менялись данные с 2012-2021 годы.

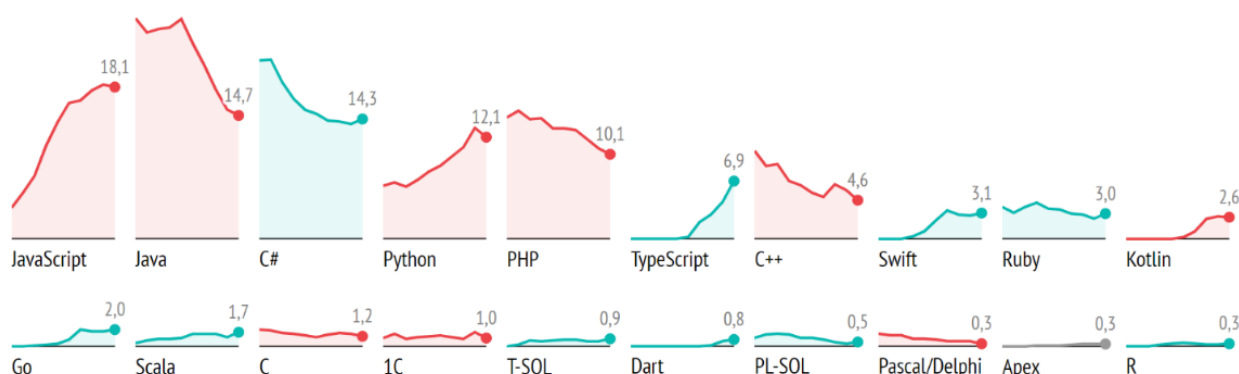


Рис. 2 – График изменения популярности языков программирования, % [3]

Архитектура информационной системы – абстрактное понятие, определяющее из каких составных частей (элементов, компонент) состоит приложение и как эти части между собой взаимодействуют. Архитектура информационной системы позволяет выделить:

- возможности системы по работе с интерфейсом пользователя;

разработаны с помощью технологии искусственного интеллекта.

На сегодняшний день существует большое количество различных языков программирования и у каждого из них своя сфера применения, но всё же для проведения анализа на выбор лучшего языка для написания чат-бота, необходимо выбрать несколько самых популярных языков, чтобы между ними проводить анализ, итак, в этой главе обратимся к статистике по популярности языков [3].

На рисунке 1 показан рейтинг языков 2021 в коммерческих рабочих проектах.

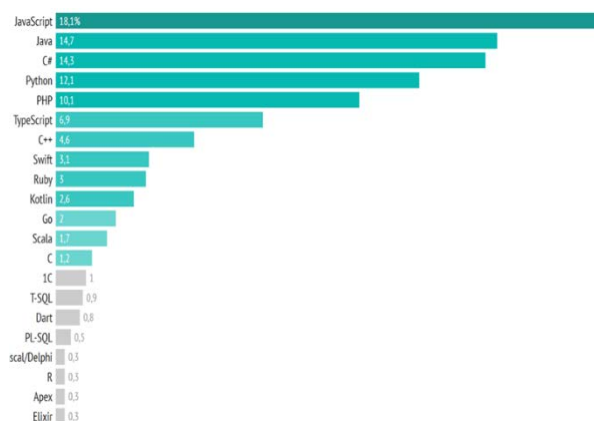


Рис. 1 - Рейтинг языков программирования в коммерческих проектах [3]

- принцип работы алгоритмов системы (обработчики событий);

- логику базы данных (добавление, удаление и хранение данных, а также выполнение запросов).

Для разработки информационной системы наиболее приемлемым будет использование «клиент-серверной» архитектуры. Пример архитектуры показан на рисунке 3.

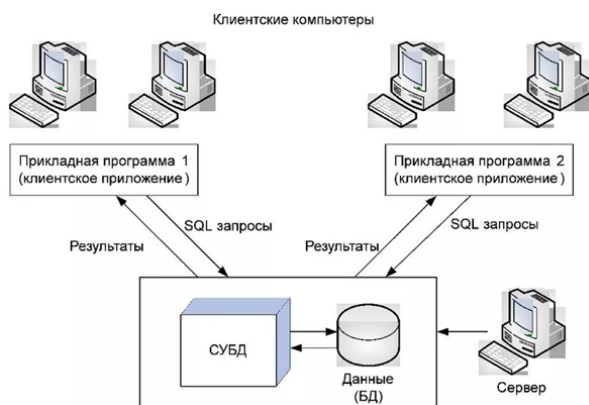


Рис. 3 – Архитектура «клиент-сервер»

Клиент-серверная архитектура чат-бота для платформы Telegram построена с применением концепции MVC («Модель-Представление-Контроллер»), то есть, модель базы данных отделена от контроллеров (подсистем) и системы обработки ввода/вывода. Применение концепции MVC позволяет выстроить безопасную и отказоустойчивую архитектуру, легкой в поддержке и модификации в будущем. Более того, крупные части приложения отделены друг от друга, т.е можно относительно без проблем заменить базу данных на другую, переписав ее модель.

База данных – это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе [4]. База данных в проекте необходима для выполнения операций с данными (добавление, сохранение, удаление).

```
class Button(db.Model):
    __tablename__ = 'buttons'
    query: sql.Select
    id = Column(Integer, Sequence('user_id_seq'), primary_key=True)
    category_code = Column(String(50))
    category_name = Column(String(50))
    subcategory_code = Column(String(50))
    subcategory_name = Column(String(50))
    info = Column(String(100000))
    button = Column(String(50))

    def __repr__(self):
        return f"""
        {self.id} - "{self.info}"""
```

Рис. 4 – Модель таблицы «buttons»

В ходе разработки базы данных были создана модель Button. В данной таблице хранятся сведения о кнопках, выдаваемых пользователю в меню чат-бота: id, код категории (для отображения в CallbackData), название категории (для отображение на кнопке), код подкатегории (для отображения в CallbackData), название подкатегории (для отображения в кнопке), название кнопки (для отображения на кнопке), информация под кнопкой (для отображения информации по нажатию по кнопке).

Модель таблицы представлена на рисунке 4

Подсистема обрабатывает события на вызов пользователем динамического меню через кнопку «Главное меню», обрабатывает событие при команде start, кнопки «Справка», отвечает на ввод неизвестной команды, обрабатывает ошибки, является входом в блок промежуточного программного обеспечения (middleware) такие как подсистема антифлуда и логгирование событий, возвращает пользователю результат обработки его запроса. Любое действие пользователя рассматривается как событие. Событийная модель реагирования на действия пользователя позволяет выстроить простую и понятную архитектуру входящих команд. Перечень событий и вспомогательных функций, используемых в подсистеме приведен таблице 1. Подсистема работы меню и взаимодействия с базой данных описывает логику работы меню т.е обработка событий по нажатию на кнопки и обработка запросов к базе данных для получения информации для кнопок в динамическом меню чат-бота. Перечень событий и вспомогательных функций, используемых в подсистеме приведен таблице 1.

Данная подсистема служит для заполнения данными динамического меню, для корректного заполнения информации, подсистема запускается администратором в ручном режиме после предварительной проверки подсистемы на заполнение шаблона заполнения достоверными данными.

У данной подсистемы всего одна функция, представленная на рисунке 5.

Таблица 1 – Подсистемы

Function Declaration обработки событий	Function Declaration работы меню и взаимодействия с базой данных
<pre>@dp.message_handler(text="Главное меню") @rate_limit(15, 'Главное меню') async def show_menu(message: types.Message):</pre>	<pre>menu_cd = CallbackData("show_menu", "level", "category", "subcategory", "button_id")</pre>
<pre>@dp.message_handler(CommandStart()) @rate_limit(15, 'start') async def bot_start(m: types.Message):</pre>	<pre>def make_callback_data(level, category="0", subcategory="0", button_id="0"): return menu_cd.new(level=level, category=category, subcategory=subcategory, button_id=button_id)</pre>

<pre> async def list_categories(message: Union[CallbackQuery, Message], **kwargs): markup = await categories_keyboard() </pre>	<pre> async def categories_keyboard(): </pre>
<pre> async def list_subcategories(callback: CallbackQuery, category, **kwargs): markup = await subcategories_keyboard(category) </pre>	<pre> categories = await get_categories() for category in categories: </pre>
<pre> async def list_buttons(callback: CallbackQuery, category, subcategory, **kwargs): markup = await buttons_keyboard(category, subcategory) </pre>	<pre> callback_data = make_callback_data(level=CURRENT_LEVEL + 1, category=category.category_code) </pre>
<pre> async def show_button(callback: CallbackQuery, category, subcategory, button_id): markup = button_keyboard(category, subcategory, button_id) </pre>	<pre> async def subcategories_keyboard(category): </pre>
<pre> button = await get_button(button_id) text = f"{button.info}" await callback.message.edit_text(text=text, reply_markup=markup) </pre>	<pre> subcategories = await get_subcategories(category) for subcategory in subcategories: </pre>
<pre> @dp.callback_query_handler(menu_cd.filter()) async def navigate(call: CallbackQuery, callback_data: dict): </pre>	<pre> callback_data = make_callback_data(level=CURRENT_LEVEL + 1,category=category, subcategory=subcategory.subcategory_code) </pre>
<pre> @dp.message_handler(text="Справка") @rate_limit(15, 'Справка') async def info(message: types.Message): </pre>	<pre> async def buttons_keyboard(category, subcategory): </pre>
<pre> @dp.message_handler() @rate_limit(15, None) async def unknown_command(m: types.Message): </pre>	<pre> buttons = await get_buttons(category, subcategory) for button in buttons: </pre>
	<pre> callback_data = make_callback_data(level=CURRENT_LEVEL + 1,category=category, subcategory=subcategory, button_id=button.id) </pre>
	<pre> def button_keyboard(category, subcategory, button_id): </pre>

```

async def add_buttons():
await add_button(info=f"***"
f"***",
category_name="***", category_code="***",
subcategory_name="***", subcategory_code="***",
button="***")

```

Рис. 5 – Подсистема заполнения базы данных

Промежуточное программное обеспечение, так же, называют программным обеспечением среднего слоя. Данный блок кода применяется к каждому запросу, приходящему в приложение. В нашем проекте, было реализована 3 промежуточные ПО это логгирование и подсистема антифлуда и оповещение администраторов о запуске бота.

Подсистема логгирования выводит в консоль администратора на сервере, логи со статусом

INFO, так же можно переключить режим регистрации событий в статус DEBUG, для отладки приложения, т.е любое событие в приложение от пользователей регистрируется в подсистеме.

Так выглядит функция для вызова подсистемы в консоль администратора, представленная на рисунке

6.

```
logging.basicConfig(format='u'%(filename)s [LINE:%(lineno)d] #%(levelname)-8s [%(asctime)s] %(message)s',
                    level=logging.INFO,
                    #level=logging.DEBUG,
```

Рис. 6 – Функция для вызова подсистемы в консоль администратора

Подсистема антифлуда обрабатывает *timeout* на вызов команды, введенной пользователем. Простыми словами, пользователь может ввести одну и ту же команду не чаще одного раза за 15

секунд. Функция *def rate\_limit(limit: int, key=None)*: вызывается декоратором в выбранном хэндлере в представленном рисунке 7 .

```
@dp.message_handler(text="Главное меню")
@rate_limit(15, 'Главное меню')
async def show_menu(message: types.Message):
    # Выполним функцию, которая отправит пользователю кнопки с доступными категориями
    await list_categories(message)
```

Рис. 7 – Подсистема антифлуда

### III. ПРИКЛАДНАЯ ИНТЕРПРЕТАЦИЯ И ИЛЛЮСТРАЦИЯ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ ИССЛЕДОВАНИЯ

Подсистема оповещения о запуске бота, вызывается функцией *async def on\_startup\_notify(dp: Dispatcher)*: в основном файле *app.py* служащим для запуска бота. Тем самым с запуском чат-бота на сервере, администраторами придёт оповещение о том, что «Бот запущен».

Для запуска информационной системы необходимо отправить команду */start* боту *@ngieu\_kbot*.

Команды боту можно отдавать, используя клавиатуру.

При вызове команды */start* пользователю будет выведено приветственное сообщение, представленное на рисунке 8.

Для вызова справки о боте, пользователю будет предложено нажать кнопку «Справка» и по нажатию чат-бот выведет информацию пользователю на экран.

Для вызова Главного меню, пользователю предложено нажать кнопку «Главное меню». С вызовом данного события, появится динамическое меню при нажатии, на кнопки которого из базы данных будут появляться запросы на получение данных. Кнопки будут динамически меняться, предлагая выбрать категорию, под категорию и кнопку с выводом информации под ней. Главное меню представлено на рисунках 9,10.



Рис. 8 – Старт чат-бота

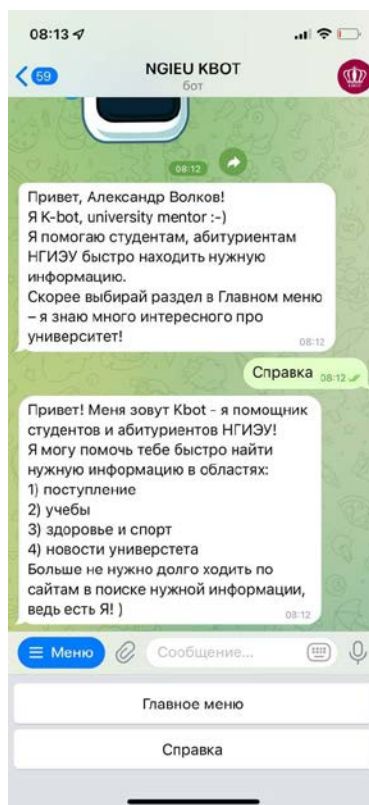


Рис.9 – Справка о боте

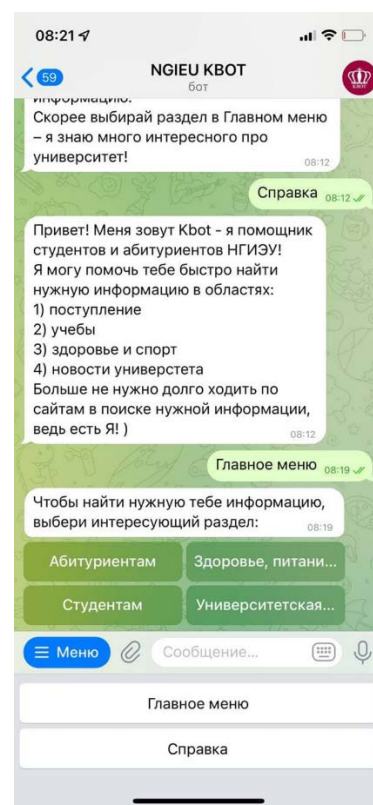


Рис.10 – Главное меню

## VII. ЗАКЛЮЧЕНИЕ

При реализации анализа предметной области был проведено сравнение средств реализации проекта, в результате которого определено что для разработки системы будет использоваться язык программирования Python и СУБД PostgreSQL. В соответствии со всеми предъявляемыми требованиями разработан простой и удобный интерфейс информационной системы, а также выполнен полный комплекс работ по написанию программного кода для реализации функционала информационной системы. Также в проектной части описан принцип работы пользователей с информационной системой.

## БИБЛИОГРАФИЯ

1. Что такое чат-боты и какие они бывают [Электронный ресурс]. –Режим доступа: <https://www.carrotquest.io/chatbot/chatbot-types/dostavki.html> (дата обращения: 15.12.2022)
2. Матвеева Н.Ю., Золотарюк А.В. Технологии создания и применения чат-ботов // Научные записки молодых исследователей. 2018. №1. URL: <https://cyberleninka.ru/article/n/tehnologii-sozdaniya-i-primeneniya-chat-botov> (дата обращения: 14.12.2022).
3. Рейтинг языков программирования 2021 [Электронный ресурс]. –Режим доступа: <https://habr.com/ru/post/543346/> (дата обращения: 15.12.2022)
4. Васильева К. Н., Хусаинова Г. Я. Реляционные базы данных // Colloquium-journal. 2020. №2 (54). URL: <https://cyberleninka.ru/article/n/relyatsionnye-bazy-dannyh> (дата обращения: 15.12.2022).

# Development of telegram bot software for students and applicants of GBOU VO NGUEU

S. Zueva, A. Krasnov, A. Krasnova

**Abstract** – This article analyzes the rating of programming languages and the graph of changes in the popularity of programming languages. Based on the analysis, a project was created and implemented, the main purpose of which is to help students during their studies by separating the required data from the main source of information into a single portal with the possibility of using a Telegram chatbot. The relevance of the development and the possibility of implementing this project in the daily use of NGUE students is explained by the fact that the Telegram messenger is not an official source for obtaining information. This platform is the most relevant for creating bots with a set of response chains, since its essence lies in the interaction of the user with the application using text messages. At the moment, NGIEU has about 5 official sources of information, one of which is the web portal "ngieu.ru ", it is also the main center for collecting, notifying and receiving data. In this regard, it was decided to create and implement a project whose main goal will be to help students during their studies by separating the required data from the main source of information into a single portal with the possibility of using a Telegram bot.

**Keywords** – telegram, chatbot, reference information system, Python, function.

## REFERENCES

1. What are chatbots and what they are [Electronic resource]. –Access mode: <https://www.carrotquest.io/chatbot/chatbot-types/dostavki.html> (accessed: 15.12.2022)
2. Matveeva N.Yu., Zolotaryuk A.V. Technologies of creation and application of chatbots // Scientific notes of young researchers. 2018. No. 1. URL: <https://cyberleninka.ru/article/n/tehnologii-sozdaniya-i-primeneniya-chat-botov> (accessed: 14.12.2022).
3. Rating of programming languages 2021 [Electronic resource]. –Access mode: <https://habr.com/ru/post/543346/> (accessed: 15.12.2022)
4. Vasilyeva K. N., Khusainova G. Ya. Relational databases // Colloquium-journal. 2020. No. 2 (54). URL: <https://cyberleninka.ru/article/n/relyatsionnye-bazy-dannyh> (accessed: 15.12.2022).