

**XXXI Межрегиональная открытая научно-исследовательская конференция  
обучающихся «Будущее Карелии»  
Направление «Проектная деятельность (проектирование)»**

**СОЗДАНИЕ УЧЕБНОГО БОТА НА PYTHON**

*Бедин Кирилл Вадимович*

*МБОУ КГО «СОШ № 3», 9«а» класс, г. Костомукша*

*E-mail: ripka10clone@yandex.ru; тел. +7(921)458-16-90*

*Руководитель Шумкина Наталья Сергеевна*

*учитель информатики*

*МБОУ КГО «СОШ №3», г. Костомукша*

*E-mail: shumns@yandex.ru; тел. +7(921)222-19-06*

г. Костомукша

2026

## Оглавление

Введение .....	3
Глава 1.....	5
1.1 Чат-боты. Их использование и возможности. ....	5
1.2 Тестовые технологии .....	5
Глава 2.....	7
2.1 Этапы проектирования .....	7
2.2 Концептуальное проектирование .....	7
2.3 Логическое проектирование .....	7
2.4 Физическое проектирование.....	8
2.5 Анализ полученных результатов .....	9
Заключение .....	11
Список использованных источников .....	12
Приложение 1 .....	13
Приложение 2 .....	14

## Введение

Бот — компьютерная программа, которая может выполнять различные задачи автоматически<sup>1</sup>.

Сегодня мы живем в мире стремительно развивающихся технологий, поэтому различные боты стали частью нашей жизни.

Можно сколько угодно спорить о пользе и вреде данных помощников, но факт остается фактом: боты проникли во все сферы жизни и деятельности человека, возможности их постоянно расширяются, они способны обрабатывать огромное количество информации, тем самым избавляя нас от рутины. И да, различные риски есть. Но ведь за компьютером сидит человек, который четко должен представлять «что такое хорошо и что такое плохо»

Итак, на сегодняшний день существует множество ботов – простых и сложных, которые применяются в различных сферах нашей жизни. Вот некоторые из них: - мультимодальные боты (работают и с текстом и с голосом). Например, GigaChat от «Сбера», или Med-Gemini-3D (обучен трактовать медицинские данные) или боты-консультанты для интернет-магазинов...

-специализированные боты, которые создаются под конкретные задачи. Например, для бронирования билетов или Sales Bot – помогает торговым компаниям или ChatGPT – бот для общения ит.д.

- интеллектуальные боты (с использованием искусственного интеллекта)

Таким образом, возможности и разнообразие ботов постоянно увеличиваются, а это предоставляет новые возможности для человека и его взаимодействию с технологиями.

Изучив Python и библиотеку aiogram(см. приложение), я попробовал создать учебный бот на Python.

**Актуальность моего проекта в том,** что сделана попытка автоматизировать и облегчить процесс контроля над пониманием учебного материала, а также использование образовательно-информационных технологий открывают новые возможности как для учеников так и для учителей в освоении и преподавании дисциплин, кроме того бот разработан для мессенджеров, так как это наиболее популярный и массовый способ организации группового тестирования.

**Практическая значимость** моего проекта состоит в том, чтобы показать возможность самостоятельного изучения программирования.

---

<sup>1</sup> Бот <https://blog.skillfactory.ru/glossary/bot/> (дата обращения: 05.01.2026)

**Проблема:** необходимость получения новых знаний и разработка собственного бота, отличающегося от других.

**Цель проекта:** написание программы для бота на Python.

**Задачи:** 1) изучить библиотеку aiogram; 2) написать пробный проект для закрепления знаний; 3) создать собственного учебного бота; 4) сделать анализ трудностей, с которыми пришлось столкнуться.

**Объект исследования:** боты, взаимодействующие с пользователем через сторонние сервисы.

**Предмет исследования:** учебный бот.

**Продукт:** учебный бот на языке программирования Python.

**Методы исследования:** теоретические – изучение документации, статей, интернет-ресурсов по данной теме;

Эмпирические – программирование и тестирование.

Тип проекта: смешанный.

## Глава 1

### 1.1 Чат-боты. Их использование и возможности.

Чат-боты это виртуальные помощники, часть которых выполняет примитивные действия (напоминания, бронирование, запись информации, автоответчик), а другая построена на основе искусственного интеллекта. Основоположителем можно считать Алана Тьюринга и его идею создания интеллектуальных машин<sup>2</sup>.

Чат-боты могут быть простыми, выполняющими одну какую-то функцию, а могут обладать искусственным интеллектом. Их можно настроить в мессенджерах и соц.сетях.

Функционал чат-ботов также разнообразен. Чат-бот для продаж – консультирует покупателей и помогает им подобрать нужный товар, бот-информатор помогает искать и анализировать информацию, бот-ассистент в состоянии автоматизировать ответы и взаимодействовать с клиентами и т.д. Таким образом, чат-боты помогают найти нужные сведения, бронируют билеты и заказы, собирают отзывы, консультируют, обрабатывают денежные транзакции.

Сферы деятельности, где используются чат-боты расширяются. Как правило, чат-боты удобны для клиентов. Они быстро ищут информацию, у них нет выходных, работают быстро.

### 1.2 Тестовые технологии

Многочисленные изменения, которые происходят в обществе, не могли не отразиться и на школе. Меня заинтересовал вопрос технологии тестирования знаний. Познакомившись с рядом статей о тестовых технологиях в образовании, я пришел к выводу о том что: тесты – это оперативная проверка знаний по пройденному материалу, это возможность получить объективный результат, это определенный контроль над качеством усвоения материала. Как пишет Гладких Алексей Андреевич в своей публикации «Тестовые технологии как средство повышения качества образования»... XXI век предъявляет три главных требования к *тестовой технологии*: это адаптивность, качество и эффективность. Адаптивность технологий предполагает приоритет личности учащихся и необходимость создания таких технологий, которые способны реагировать на индивидуальные различия испытуемых, регулируя меру трудности заданий зависимости от успешности ответов на предыдущие задания...

---

<sup>2</sup> Всё о чат-ботах: виды, преимущества, успешные примеры. Электронный ресурс <https://practicum.yandex.ru/blog/chto-takoe-chat-bot/> (дата обращения: 07.01.2026)

Качество технологии связано преимущественно с надежностью и валидностью тестовых результатов. Эффективность технологий предполагает уменьшение отношения затраты/ результаты.»...

Итак, компьютерное тестирование позволяет быстро получить результат, это менее трудозатратно, чем например, бумажное тестирование, ученики могут легко выявить пробелы в своих знаниях и предпринять меры, а работа учителя упрощается<sup>3</sup>.

Такие же негативные явления компьютерного тестирования как, например, психологический дискомфорт или технические неполадки или невозможность видеть учителю ход решения можно уменьшить при дальнейшем развитии этого направления.

---

<sup>3</sup> Гладких А.А. «Тестовые технологии как средство повышения качества образования». Электронный ресурс <https://nsportal.ru/shkola/raznoe/library/2020/11/09/testovye-tehnologii-kak-sredstvo-povysheniya-kachestva-obrazovaniya> (дата обращения: 05.01.2026)

## Глава 2

### 2.1 Этапы проектирования

В начале работы были рассмотрены аналоги чат-ботов. На основе сделанных выводов определились функции будущего проекта. Анализируя способы реализации данного проекта, решил остановиться на боте на языке программирования Python с использованием библиотек: aiogram(взаимодействие с пользователями), sqlite3(СУБД) и numpy(оптимизация некоторых расчётов).

При работе над проектом я прошел следующие этапы:

1. Концептуальное проектирование
2. Логическое проектирование
3. Физическое проектирование
4. Анализ полученных результатов.

### 2.2 Концептуальное проектирование

На данном этапе я изучил литературу по проекту компьютерного тестирования, продумал идею создания собственной системы тестирования. Определился с целью проекта и внешним оформлением. К реализации принял учебного бота, как помощника учителю. Задания будут заранее разработаны администраторами и загружены в систему. задания будут прикрепляться к классу, для их выдачи ученикам, учителем. Учитель сможет выбрать тесты и изменить ограничения для них. Именно бот способен обеспечить скорость, массовость, сокращение трудозатрат.

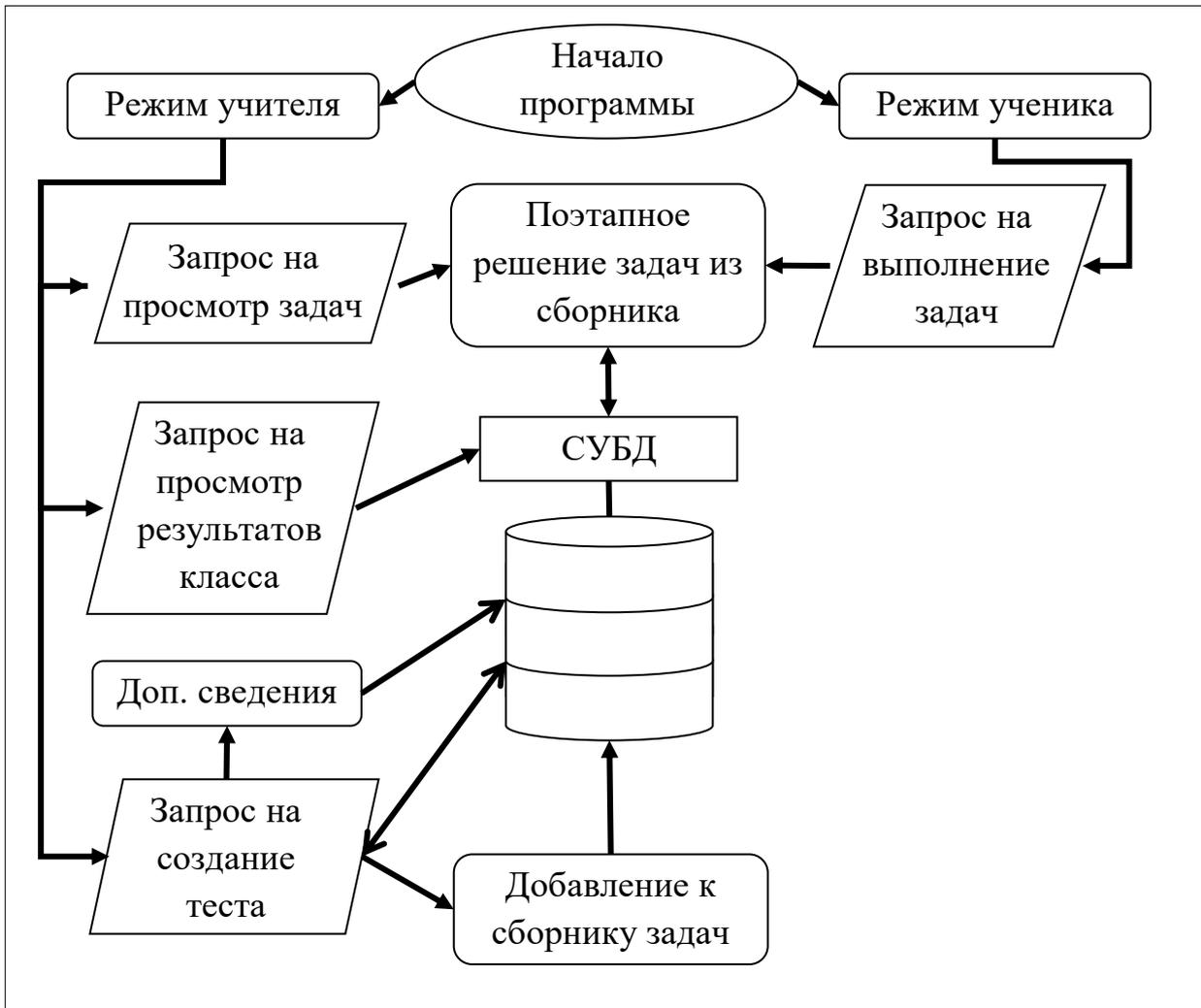
### 2.3 Логическое проектирование

На этом этапе я разработал алгоритмы работы учебного бота. Он должен быть способен быстро отвечать на запросы пользователей, иметь гибкий и отказоустойчивый код, должен поддерживать многопоточность и быстрый доступ к базе данных.

Для работы с системой, учитель должен выбрать тематику теста и тест из предложенного списка, по желанию он сможет изменить ограничения

теста. После этого учитель должен прикрепить тест к сборнику задач или создать его, выбрав классы к которым будут выданы задачи и время на их выполнение. После входа ученика в систему, он сможет просмотреть все невыполненные им сборники задач и прорешать их до истечения срока выполнения.

Результатом логического проектирования стала схема ниже.



Пояснение к схеме: возврат в начало программы может быть осуществлён на любом этапе выполнения программы.

## 2.4 Физическое проектирование

На этапе физического проектирования был разработан учебный бот. Его целевая аудитория: учителя и ученики общеобразовательных школ.

Бот генерирует тесты для проверки текущих знаний. Предполагаемая платформа для внедрения проекта – мессенджеры.

Программная часть бота представляет собой многофайловую структуру программного кода и виртуальной оболочки `ruvenv`. Данная особенность помогает быстро развёртывать программное обеспечение.

Препятствием в создании иерархической многопапочной файловой структуры являются ограничения библиотеки `sqlite3` и необходимость в обработке ошибок, что вынесены в отдельный файл.

В проекте также была использована библиотека `aiogram3`, являющаяся основой в передаче данных между программой и пользователем через мессенджер. Она была выбрана из-за полного соответствия с оригинальной документацией core API используемого мессенджера, асинхронностью и возможностью компилирования на PyPy.

Код был разделён на 8 файлов в единственном каталоге. Точка входа, `middleware`, файл с СУБД и обработкой исключений, проверки, утилита для создания страниц из `inline-кнопок` - все вынесены в отдельные файлы. `authorization.py`, `collproblems.py`, `tests.py` содержат логику из пункта 2.2 Логическое проектирование.

Так как при написании кода будет использоваться библиотека `sqlite` с динамическим распределением типов, то стоит однозначно определить структуру базы данных.

Создание таблиц базы данных и регистрирование обработчиков происходит в файле `utils.py`. Всего создаётся 4 таблицы: `authorized` – содержит ID пользователя и в какой аккаунт он может войти, `tests_table` – содержит все сгенерированные учителем тесты, `collections_table` – хранит сборники задач и тесты к ним, `classes_table` – хранит ID учителя и ученика, а также название класса.

Было добавлен ещё один тип данных – JSON, и изменён DATE, так как стандартные обработчики этого типа устарели, начиная с Python 3.12. JSON поддерживает хранение как списков, так и словарей. Следует обратить внимание, что везде, кроме в `tests_table` в `doneBy`, JSON подразумевает хранение в списках, а в `tests_table` `doneBy` хранение осуществляется в словаре.

Результат физического проектирования, описание работы можно посмотреть в видео файлах <https://disk.yandex.ru/i/t0eIr-7PX11Vlg> <https://disk.yandex.ru/i/1j-3akQCE9imZA>

Некоторые скриншоты кода см. **Приложение 2**

Ссылка на код: <https://github.com/KirillPipka/teacher-s-helper-bot>

## 2.5 Анализ полученных результатов

### Результаты работы

В ходе работы над реализацией проекта был создан учебный бот на языке программирования Python. Бот позволяет создавать тесты учителям и проверять знания учеников, зарегистрированных в системе. Пользователи бота могут проверять свои знания и знания других людей, если были предоставлены полномочия учителя.

### Перспективы использования результатов работы

Распространить функционал для создания тестов, добавить расширенные функции.

При использовании учебного бота, его успешной работы – следует учитывать материальные (наличие смартфона, планшета и др. с выходом в интернет) и профессиональные ресурсы (понимание логики использования программы).

## Заключение

В ходе работы над проектом «Создание учебного бота на Python», я создал собственного бота. Таким образом, намеченная цель достигнута, а поставленные задачи выполнены.

В заключении я пришел к выводу, что сделать самостоятельно бота – реально. А если подойти к изучению этой темы более детально, то возможно и создание более сложных ботов.

Работа над проектом была начата мною в 2024 году. На сегодняшний день, планирую учесть пожелания и потребности пользователей.

Предполагаю, что можно добавить:

- распечатывание тестов на бумаге,
- перевод на другие языки через fluelent,
- создание тестов учителями с помощью L<sup>A</sup>T<sub>E</sub>X,
- выдача индивидуальных заданий, создание docker-файла,
- облегчение кода для компиляции PyPy.

### Список использованных источников

1. Бот <https://blog.skillfactory.ru/glossary/bot/> (дата обращения: 05.01.2026)
2. Всё о чат-ботах: виды, преимущества, успешные примеры. Электронный ресурс: <https://practicum.yandex.ru/blog/что-такое-chat-bot/> (дата обращения: 07.01.2026)
3. Гладких А.А. «Тестовые технологии как средство повышения качества образования». Электронный ресурс: <https://nsportal.ru/shkola/raznoe/library/2020/11/09/testovye-tehnologii-kak-sredstvo-povysheniya-kachestva-obrazovaniya> (дата обращения: 05.01.2026)
4. aiogram <https://aiogram.dev>

**Приложение 1**

Результат физического проектирования:

- <https://disk.yandex.ru/i/t0eIr-7PX11Vlg>

- <https://disk.yandex.ru/i/1j-3akQCE9imZA>

Ссылка на код: <https://github.com/KirillPipka/teacher-s-helper-bot>

## Скриншоты элементов кода

## Реализация переброса событий в сцену для дальнейшей обработки

```

@dp.callback_query(F.data[:13] == "back_to_scene")
@flags.permission("all")
async def back_to_scene(data: types.CallbackQuery | types.Message,
                       scenes: ScenesManager, state: FSMContext) -> None:
    additional_info = (await state.get_data())["additional_info"]
    if type(data) == types.CallbackQuery:
        identification = data.data[13:]
    elif type(data) == types.Message:
        try:
            identification = additional_info["b2s_args"]["identification"]
        except KeyError:
            identification = ''
    else:
        logger.error("type(data) in back_to_scene is unknown: " + str(type(data)))
        return

    back_to: str
    step: int
    match additional_info["dataFor"]:
        case 9:
            back_to = additional_info["pagedview_args"]["back_to"]
            step = additional_info["pagedview_args"]["step"]
        case 10:
            back_to = additional_info["b2s_args"]["back_to"]
            step = additional_info["b2s_args"]["step"]
        case _:
            return
    kwargs = {"entered_step": step, "identification": identification}

    # This check is here because it lets re-enter scene without waiting
    if not isinstance(
        await scenes._get_active_scene(),
        type(await scenes._get_scene(back_to))
    ):
        return

    await scenes.enter(
        scene_type = back_to,
        _check_active = False,
        **kwargs
    )

```

## Реализация мидлваря для контроля уровня доступа к программе

```

from aiogram.dispatcher import flags
from typing import Any, Callable, Dict, Awaitable
from aiogram import types, BaseMiddleware
from utils import error_occured

class PermissionMiddleware(BaseMiddleware):
    async def __call__(
        self,
        handler: Callable[[types.TelegramObject, Dict[str, Any]], Awaitable[Any]],
        event: types.TelegramObject,
        data: Dict[str, Any]
    ) -> Any:
        if flags.get_flag(data, "skip_permission_middleware"):
            return await handler(event, data)
        logged_as = (await data["state"].get_data()).get("logged_as", -1)
        permission_for: str = flags.get_flag(data, "permission")
        if permission_for == "teacher" and logged_as == 1:
            return await handler(event, data)
        elif permission_for == "student" and logged_as == 0:
            return await handler(event, data)
        elif permission_for == "all" and logged_as != -1:
            return await handler(event, data)

        message: types.Message = data['event_update']
        if permission_for == "teacher":
            return await error_occured(message, "lt")
        elif permission_for == "student":
            return await error_occured(message, "ls")
        return await error_occured(message, "ln")

```