

План курса: Введение в искусственный интеллект и создание собственного чат-бота

Для школьников 13–16 лет

Общая информация о курсе

Материалы курса находятся тут: [Github](#)

Цель курса: Дать общее представление об искусственном интеллекте (ИИ), его направлениях и практическом применении. Научить использовать современные нейросетевые инструменты (включая большие языковые модели), познакомить с основами Python и разработать собственный проект (чат-бот, веб-приложение) с интеграцией ИИ.

Формат и длительность:

- 9 занятий, каждое по 2 часа 45 минут (165 минут).
- Каждое занятие включает несколько блоков (примерно по 30–40 минут), а также один перерыв 10–15 минут.
- В конце каждого занятия — короткая рефлексия (5–10 минут) для закрепления материала и обсуждения вопросов.

Основные результаты по окончании курса:

- Понимание, что такое ИИ, какие бывают направления (генеративные модели, чат-боты, компьютерное зрение и др.).
- Базовые навыки программирования на Python (переменные, циклы, функции), умение работать в Google Colab и в IDE.
- Умение использовать ИИ в учёбе (поиск информации, объяснение решений, генерация идей).
- Опыт творческой генерации контента (музыка, изображения, видео) с помощью нейросетей.
- Знания о промпт-инжиниринге (Chain of Thought, Few-shot), а также принципах Retrieval-Augmented Generation (RAG).
- Создание собственного проекта (или веб-приложения) с ИИ и его презентация.

Содержание

1	Занятие 1: Введение в искусственный интеллект (2:45)	3
2	Занятие 2: Основы Python и GitHub (2:45)	3
3	Занятие 3: Использование ИИ для учебных целей (2:45)	4
4	Занятие 4: Творческие проекты с нейросетями (2:45)	4
5	Занятие 5: Промпт-инжиниринг и создание чат-бота (2:45)	5
6	Занятие 6: Retrieval-Augmented Generation (RAG) и векторные базы данных (2:45)	5
7	Занятие 7: Проектирование чат-бота: логика, структура, база. (2:45)	6
8	Занятие 8: Реализация и интерфейс чат-бота (2:45)	7
9	Занятие 9: Пред-финал проекта: презентация и финальная доработка (2:45)	7

1 Занятие 1: Введение в искусственный интеллект (2:45)

Цель

Понять, что такое ИИ, где он применяется, а также познакомиться с генеративными моделями (изображения, аудио, видео, музыка) и опробовать их на практике.

Блок 1 (30–40 мин): Теория и история ИИ. Краткая история развития искусственного интеллекта (Алан Тьюринг, первые нейронные сети и т. д.). Основные направления: чат-боты, компьютерное зрение, генеративные модели, системы рекомендаций. Небольшое обсуждение: какие ИИ-сервисы уже знакомы школьникам.

Блок 2 (30–40 мин): Практика с генеративными моделями. Демонстрация различных сервисов (генерация изображений, музыки и т. д.). Самостоятельное тестирование: в парах или группах пробуют сгенерировать короткую музыку или картинку.

(Перерыв 10–15 мин)

Блок 3 (30–40 мин): Обсуждение результатов и примеры применения. Просмотр/прослушивание того, что получилось у участников. Короткая лекция: почему модели иногда ошибаются, как формируются «странные» результаты. Ответы на вопросы о возможностях и ограничениях нейронных сетей.

Блок 4 (30–40 мин): Итоговая рефлексия и «шаги вперёд». Краткая дискуссия об этике (deepfake, фейковые изображения). Итог занятия: что нового узнали и где в будущем можно применять ИИ.

2 Занятие 2: Основы Python и GitHub (2:45)

Цель

Освоить базовые конструкции языка Python (переменные, типы, циклы, функции), научиться работать в Google Colab (или подобной среде), а также сделать первые шаги в Git и GitHub - создать репозиторий и сохранить мини-проект.

Блок 1 (30–40 мин): Введение в Python. Настройка среды (Colab, Jupyter, repl.it). Настройка среды (Colab, Jupyter, repl.it). Переменные, типы данных (числа, строки, списки, словари). Первая программа: «Hello, World!», простые арифметические операции.

Блок 2 (30–40 мин): Управляющие конструкции и упражнения. `if`, `for`, `while`. Условные операторы и циклы: `if`, `for`, `while`. Практикум: вывести чётные числа, посчитать сумму списка. Работа в парах: помощь друг другу, обсуждение подходов к решению.

(Перерыв 10–15 мин)

Блок 3 (30–40 мин): Функции и мини-проект. Создание функций (`def`). Создание собственных функций (`def`). Мини-проект: «Калькулятор» или «Генератор случайных паролей». Поиск ошибок, отладка с `print`.

Блок 4 (10–20 мин): GitHub. Затем - введение в Git и GitHub: регистрация аккаунта; создание нового репозитория; добавление .ру файлов; оформление первого README.md.

Блок 5 (20–30 мин): Итоги и расширение. Короткий опрос или квиз по пройденным темам. Рефлексия: с какими сложностями столкнулись, что понравилось.
(Опционально) Задание на дом: придумать и реализовать маленькую программу.

3 Занятие 3: Использование ИИ для учебных целей (2:45)

Цель

Показать, как нейросети помогают в учёбе: от поиска информации и проверки фактов до объяснения решений.

Блок 1 (30–40 мин): Обзор ИИ-инструментов для учёбы. ChatGPT (или аналоги), сервисы для генерации рефератов, презентаций, тезисов. Примеры: попросить ИИ объяснить задачу, проверить грамматику, подобрать аргументы.

Блок 2 (30–40 мин): Групповой практикум. Каждая команда выбирает реальную задачу (математика, литература и т.д.). Формулирует вопрос к ИИ, анализирует ответ на корректность и полноту. Обсуждение: почему нужно перепроверять фактологические сведения.

(Перерыв 10–15 мин)

Блок 3 (30–40 мин): Расширенные кейсы использования. Как ИИ может помогать в подготовке докладов и презентаций (генерация слайдов, тем и т.д.).

Блок 4 (30–40 мин): Подведение итогов. Дискуссия: плюсы и минусы применения ИИ в учебе, границы «честного» использования. Итог занятия: как и когда применять ИИ-инструменты для учёбы.

4 Занятие 4: Творческие проекты с нейросетями (2:45)

Цель

Показать, как можно использовать нейросети для генерации музыки, изображений, видео и вдохновить на креативные эксперименты.

Блок 1 (30–40 мин): Виды творческих моделей. Генерация музыки (Amper Music, Boomy), изображений (DALL·E, Midjourney-подобные), видео (Runway). Примеры/демо: показать готовые ролики, треки, рисунки, попробовать в реальном времени нейросети.

Блок 2 (30–40 мин): Практикум генерации контента. Участники делятся по интересам: музыка, картинки, видео. Эксперименты с настройками (жанры, стили, ключевые слова). Обсуждение: как получить более точные результаты.

(Перерыв 10–15 мин)

Блок 3 (30–40 мин): Улучшение и доработка. Исправление неудачных результатов, подбор оптимальных промптов. Совместный разбор: почему возникают искажения или «странные» детали.

Блок 4 (30–40 мин): Итоговая презентация. Каждая группа показывает итог (музыка, серия картинок, небольшой видеоролик). Обсуждение: какие идеи появились для будущих проектов.

5 Занятие 5: Промпт-инжиниринг и создание чат-бота (2:45)

Цель

Научиться формулировать эффективные промпты для языковых моделей и сделать простой прототип чат-бота на Python через API.

Блок 1 (30–40 мин): Основы промпт-инжиниринга. Что такое промпт, почему важна точность формулировки. Структура запроса: контекст, уточнение деталей, желаемый формат ответа. Примеры: сравнение «короткого» и «подробного» промпта.

Блок 2 (30–40 мин): Создание простого текстового бота. Подключение к API (например, OpenAI). Минимальный код: отправить запрос к модели, получить ответ. Задание: сделать «ролевого» бота (продавец, учитель, шутник).

(Перерыв 10–15 мин)

Блок 3 (30–40 мин): Практические эксперименты. Отладка и тестирование разных «системных» сообщений. Анализ качества ответов, доработка промптов.

Блок 4 (30–40 мин): Итоги занятия. Демонстрация разных версий бота. Обсуждение: как промпт влияет на характер ответов.

6 Занятие 6: Retrieval-Augmented Generation (RAG) и векторные базы данных (2:45)

Цель

Понять, как языковая модель может использовать внешние источники знаний (тексты, документы) для генерации более точных и осмысленных ответов. Освоить базовые принципы векторного поиска и эмбедингов, а также научиться использовать FAISS и локальные LLM.

Блок 1 (30–40 мин): Теория RAG. Что такое Retrieval-Augmented Generation (RAG): поиск информации + генерация ответа. Преимущества и ограничения: более актуальные, точные ответы, но сложнее реализация.

Блок 2 (30–40 мин): Chain of Thought и Few-shot. Chain of Thought (CoT): разбивка задачи на последовательные шаги. Few-shot: показ нескольких примеров формата «вопрос–ответ» для повышения качества генерации. Практика: сравнить ответы с CoT/без CoT, с примерами/без примеров.

(Перерыв 10–15 мин)

Блок 3 (20–30 мин): Векторные базы данных. Что такое эмбединги: превращаем текст в вектор чисел; Инструменты: SentenceTransformers, FAISS; Практика: загрузка набора текстов (файлы/вопросы), индексация в FAISS.

Блок 4 (20–30 мин): Мини-RAG на практике. Заготовленный набор текстов (или векторная база). Поиск релевантного параграфа и передача его в промпт модели. Простой код, иллюстрирующий концепцию.

Блок 5 (20–30 мин): Итог и обсуждение. Где и когда применять RAG, CoT, Few-shot. Ограничения: модель может «выдумывать» ответы, если нет подходящих данных.

7 Занятие 7: Проектирование чат-бота: логика, структура, база. (2:45)

Цель

Определиться с темой и задачами будущего чат-бота, продумать его логику, структуру диалогов, интерфейс и хранилище данных. Сформировать рабочую команду и подготовить проект к технической реализации.

Блок 1 (30–40 мин): Генерация идей и выбор проекта. Обсуждение разных видов ботов: продавец, учитель, квест, викторина и т. д. Формирование команд: кто отвечает за код, интерфейс, тексты, тестирование.

Выбор проекта для реализации:

1. Telegram-бот с локальной языковой моделью
2. Бот-суммаризатор текста / презентаций
3. Генератор изображений по описанию
4. Предсказатель цен на квартиры
5. Бот для распознавания животных на фото

Блок 2 (30–40 мин): Создание структуры и логики. Общая схема диалога: ветвления, ключевые функции. Скетч (блок-схема) основных сценариев.

(Перерыв 10–15 мин)

Блок 3 (30–40 мин): Распределение задач и ресурсов. Планирование проекта и создание репозитория. Создание GitHub-репозитория Название, описание, README.md Распределение ролей в команде:

1. Программисты

2. дата инженеры

3. ml-инженеры

Блок 4 (30–40 мин): Результат занятия. Каждая команда формирует план (список задач, сроки). Короткая презентация планов, обмен идеями, возможно — советы по реализации.

8 Занятие 8: Реализация и интерфейс чат-бота (2:45)

Цель

Перейти к практической разработке: написать код бота, подключить интерфейс (веб или Telegram), проверить основные сценарии.

Блок 1 (30–40 мин): Старт разработки. Настройка окружения: GitHub, Colab, локальная IDE. Подключение к LLM внутри IDE локально, тестовый запрос.

Блок 2 (30–40 мин): Основной код и логика диалога. Обработка входящих сообщений, генерация ответа, хранение контекста (при необходимости). Тестовые диалоги, быстрая проверка.

(Перерыв 10–15 мин)

Блок 3 (30–40 мин): UX/UI и тестирование. Telegram: бот через aiogram — проверка команд, приветствия и т. д. Обмен ботами между командами, поиск багов, сбор обратной связи.

Блок 4 (30–40 мин): Текущее состояние и задачи на доработку. Составить список всех багов, «хотелок» и улучшений. Итог дня: что уже получилось и что нужно доделать к финалу.

9 Занятие 9: Пред-финал проекта: презентация и финальная доработка (2:45)

Цель

Завершить проект, отрепетировать выступление, записать видеопрезентацию и подготовиться к защите.

Блок 1 (30–40 мин): Подготовка презентации. Структура доклада: идея, функции бота, используемые технологии (CoT, RAG и т. д.). Слайды или короткая демо-версия, распределение ролей в выступлении.

Блок 2 (30–40 мин): Репетиция выступления. Каждая команда пару раз прогоняет презентацию: показывает бота, объясняет логику. Обратная связь: преподаватель и участники указывают, что улучшить.

(Перерыв 10–15 мин)

Блок 3 (30–40 мин): Финальная доработка кода. Исправление критических багов. (По возможности) Добавление мелких улучшений или «фишек».

Блок 4 (30–40 мин): Запись видеопрезентации и итог. Короткий ролик (2–3 минуты) с демонстрацией бота и комментариями команды. Итог: проект готов к финальному показу.