| **Разработка приложений с ИИ (урок 3)**  **Методические рекомендации по организации урочной и внеурочной  деятельности в рамках урока**  **Авторы**  Команда экспертов Академии искусственного интеллекта для школьников Благотворительного фонда Сбербанка «Вклад в будущее» |
| --- |
| **Для возраста**  7–11 класс  **Трудоемкость**  45 минут |

**Урок 2 «От идеи к приложению: создание прототипа»**

**Цель:** Научиться создавать прототип Telegram-приложения, используя базовые принципы UX/UI-дизайна. Освоить практику работы с ИИ-инструментами для генерации визуальных элементов, текстов и кода (HTML, CSS).

**Задачи:**

* Ознакомить обучающихся с основами простого и удобного дизайна и принципами составления корректных промптов для ИИ.
* Продемонстрировать работу ИИ-инструментов (например, Kandinsky для создания иконок, фонов и кнопок и GigaChat для генерации текстов и кода) и дать возможность обучающимся самостоятельно разработать прототип.
* Развить навыки генерации шаблонов и корректировки результатов ИИ.
* Стимулировать командное взаимодействие.
* Развить творческий подход в разработке.

**Оборудование:**

* Компьютеры, проектор, интерактивная доска.
* Доступ к интернету для демонстрации работы ИИ-инструментов.
* Подготовленные примеры промптов для Kandinsky и GigaChat.
* Шаблоны прототипов для работы (см. общий комплект файлов).
* Флипчарт или доска для записи идей во время мозгового штурма.
* Рабочие листы для каждого обучающегося.

*Упоминание сервисов Kandinsky, GigaChat, Telegram и других в материалах сделано не на правах рекламы в образовательных целях.*

**План урока**

1. **Разминка (10 минут)**

*Здравствуйте, ребята! Откройте, пожалуйста, свои рабочие листы. Сегодня мы начнём урок с творческого задания «Генератор идей».*

*На слайде вы видите примеры забавных идей для Telegram-приложений. Запишите в своих рабочих листах хотя бы одну оригинальную идею.*

*Не бойтесь быть креативными! У вас есть минута, чтобы всё записать. Поехали!*

**Активность «Генератор идей»**

*Каждый из вас должен записать в рабочем листе одну или две забавные/необычные идеи для Telegram-приложения.*

На слайде появляются примеры (генератор мемов, бот-рассказчик историй, приложение с необычными эмодзи).

После 40–50 секунд самостоятельной работы пригласить обучающихся по очереди озвучить свои идеи, фиксируя их на интерактивной доске или в общем документе.

*Отлично! Теперь, когда вы записали свои идеи, давайте перейдём ко второй части нашей разминки. Посмотрите на слайд «Что у нас в приложении?». Я хочу, чтобы каждый из вас вспомнил своё любимое приложение или какую-нибудь классную фишку, которую вы видели в приложениях. Запишите коротко в рабочих листах, что именно вам понравилось и почему. После этого я попрошу несколько обучающихся поделиться своими наблюдениями с классом. Главное — делитесь, не бойтесь высказывать свои мысли, каждая идея важна!*

**Активность «Что у нас в приложении?»**

Попросить обучающихся коротко рассказать о любимых приложениях или интересных функциях, которые они заметили в приложениях.

При этом обучающиеся снова делают записи в рабочих листах, чтобы позже сравнить свои мысли с тем, что назовёт класс.

Прокомментировать каждую озвученную идею, отмечая плюсы и побуждая к дальнейшему обсуждению.

**Запасной план. Готовая идея приложения «УчисьДелись»**

*Ребята, вижу, что сейчас у нас тихо. Может быть, идеи для приложения пока не появляются. Не переживайте, это нормально! У меня есть запасная идея.*

**«УчисьДелись»** — это инновационное Telegram-приложение, которое превращает обмен навыками в увлекательное приключение. Приложение объединяет людей, готовых делиться своими знаниями и обучаться новому, создавая живую, динамичную платформу для взаимного обучения. Здесь каждый может предложить свой уникальный навык или найти того, кто научит его новому мастерству: от программирования до кулинарии, от цифрового маркетинга до рисования.

*Возьмите свои рабочие листы, часть три. Напишите там, что, по-вашему, должно быть в таком приложении. Например, как оно может выглядеть, какие темы будет полезно добавить или как сделать рейтинги и комментарии удобными.*

*Потом я попрошу несколько человек озвучить свои идеи. Мы вместе подумаем, как сделать это приложение ещё лучше и интереснее!*

Обучающиеся работают с третьей частью рабочего листа, где они должны сделать заметки о функционале и особенностях приложения «УчисьДелись».

Записывают, какие разделы и функции, по их мнению, должны присутствовать в приложении (раздел с вопросами и ответами, лента рекомендаций, возможность оценивать полезность материалов и т. п.).

Важно объяснить, что данная идея — отличный пример того, как можно объединить обучение и обмен знаниями в одном приложении. Также необходимо задать вопросы для анализа и доработки идеи.

1. **Введение (5 минут)**

**Объяснение важности UX/UI-дизайна**

*UX (пользовательский опыт) и UI (пользовательский интерфейс) критически важны для успеха мобильных приложений, включая Telegram. Исследования показывают, что простота, удобство и визуальное единство повышают удовлетворённость пользователей, увеличивают вовлечённость и удержание.*

*Подумайте, почему ещё эти качества важны, когда вы пользуетесь приложениями? Запишите в рабочие листы, что для вас означает удобство в интерфейсе.*

**Роль ИИ-инструментов**

*А теперь давайте поговорим об инструментах, которые помогут воплотить наши идеи!*

*Первый — Kandinsky. Он как волшебная кисть, которая за несколько секунд создаст яркие иконки, фон и кнопки для нашего приложения.*

*Второй — GigaChat. Он поможет нам составить тексты и даже основу для кода.*

*Не переживайте, если вы никогда не работали с HTML или CSS: сегодня всё будет на простом уровне, и, главное, мы узнаем, как наши идеи превращаются в прототип.*

**Концепция прототипа (MVP)**

*Далее поговорим о прототипе, который мы будем создавать.*

***MVP (minimum viable product)****— это минимально жизнеспособный продукт. Представьте, что вы рисуете эскиз, который потом можно дорабатывать и улучшать. Наш MVP-прототип не должен быть идеальным с первого раза, он нужен для того, чтобы мы смогли протестировать идею и внести изменения.*

**Связь с предыдущими уроками**

Напомнить, как ранее изучались основы искусственного интеллекта и создания дизайна и текстов. Объяснить, как сегодня эти знания будут применяться на практике.

*Ребята, помните, как на предыдущих уроках мы изучали основы искусственного интеллекта и учились создавать дизайн и тексты? Сегодня мы пойдём дальше и используем наши знания, чтобы разработать прототип Telegram-приложения.*

1. **Обсуждение и мозговой штурм (5 минут)**

**Мозговой штурм**

*Сейчас мы перейдём к обсуждению идей из начала нашего урока. Подумайте, какой функционал и интерфейс должен быть у нашего Telegram‑приложения.*

*На слайде перед вами вы видите три ключевых вопроса:*

1. *Какие элементы (например, кнопки, меню, визуальные эффекты) сделают приложение интуитивно понятным?*
2. *Какие функции обязательно должны быть в нашем MVP‑прототипе?*
3. *Какие проблемы могут возникнуть при генерации контента с помощью ИИ и как их можно быстро исправлять?*

*Откройте, пожалуйста, свои рабочие листы. В течение минуты индивидуально запишите по две-три идеи на каждый из этих вопросов. Помните: ваши мысли важны и никаких глупых идей нет!*

**Ключевые вопросы для дискуссии**

* Какие особенности дизайна сделают приложение удобным для пользователя?
* Какие элементы интерфейса должны присутствовать в MVP-прототипе?
* Какие проблемы могут возникнуть при генерации ИИ-контента и как их можно оперативно решать?

**Методологический акцент**

Кратко упомянуть идеи известных работ — например, Don’t Make Me Think, Revisited из одноимённого труда и концепцию MVP из The Lean Startup.

**Don’t Make Me Think, Revisited** (Стив Круг) — принципы интуитивной навигации. Эта книга посвящена основам удобного веб-дизайна и пользовательского опыта (UX).   
  
**Главная идея** — интерфейс должен быть настолько простым, чтобы пользователю не приходилось задумываться о его использовании.

**Основные принципы:**

* Пользователь сканирует страницу, а не читает её. Информация должна быть чётко структурирована.
* Чем меньше кликов, тем лучше. Интуитивные пути экономят время.
* Простота и предсказуемость навигации повышают удовлетворённость пользователей.

**The Lean Startup** (Эрик Рис) — концепция MVP и итеративная разработка. В этой книге описан метод стартапа, который позволяет быстро тестировать идеи и адаптироваться к изменениям.

**MVP (minimum viable product)** — минимально жизнеспособный продукт. Это базовая версия продукта, включающая только ключевые функции, которые можно протестировать на реальных пользователях.

**Цикл Build-Measure-Learn («создать — измерить — научиться»)** — постоянное тестирование и доработка продукта на основе обратной связи.

**Быстрая адаптация и гибкость** — если что-то не работает, изменяем подход, а не ждём идеального результата.

Форма работы: открытая дискуссия, где каждый может внести свою идею.

1. **Разбор примера и практическое задание (40 минут)**

**Демонстрация работы ИИ-инструментов**

Продемонстрировать, как GigaChat генерирует код для макета (HTML, CSS) и текстовое наполнение, поясняя, что результат можно корректировать.

**Подготовка к работе**

Открыть интерфейс GigaChat и показать, как вставить адаптированный промпт для генерации HTML-кода веб-страницы приложения. Напомнить о ключевых секциях промпта (заголовок с названием, кнопки главного меню).

**Генерация кода**

Продемонстрировать, как ИИ генерирует код. Веб-страница приложения должна содержать заголовок «УчисьДелись» и кнопки: «Предложить навык», «Найти наставника», «О проекте», «Контакты».

Рассказать о том, что CSS-стили обеспечивают адаптивный, современный и минималистичный дизайн. Предложить их сгенерировать.

Оформить два файла: index.html и styles.css. Поместить в новую директорию только index.html. Открыть его через браузер и показать, что это просто скелет. Добавить в директорию styles.css, показать, как изменился файл. Сделать акцент на том, что HTML — это скелет, а CSS — оформление. Если уровень знаний обучающихся и время позволяют, продемонстрируйте работу JS.

**Анализ и корректировка**

Показать, как внимательно просмотреть полученный код: проверить правильность структуры (теги, разделение секций) и стиль оформления (цвета, шрифты, отступы).

Продемонстрировать, как внести простые корректировки вручную, если результат не соответствует ожиданиям.

Подчеркнуть, что итоговый код можно дорабатывать в зависимости от идей, выдвинутых в ходе дискуссии.

**Практическое задание**

Обучающиеся ведут записи в рабочем листе (части 6, 7), в котором указана последовательность действий для публикации приложения.

**Часть 6. Генератор приложения**  
  
Обучающиеся генерируют HTML- и CSS-код для прототипа приложения «УчисьДелись» с помощью GigaChat, используя предоставленный шаблон промпта.

**Часть 7. Публикация приложения**  
  
Обучающиеся выполняют действия для публикации сгенерированного приложения на платформе Amvera. Затем регистрируют его в Telegram через BotFather.

**Пояснения по работе MiniApp-приложений в Telegram**

В комплекте материалов урока находятся файлы, которые представляют собой пример MVP MiniApp-приложения:

* **index.html** задаёт структуру страницы и подключает файл стилей.
* **styles.css** определяет оформление элементов интерфейса.
* **main.py** запускает сервер, который при обращении по корневому URL-адресу отображает HTML-страницу со стилями CSS.
* **requirements.txt** подключает необходимые библиотеки для работы сервера.

Это MiniApp-приложение можно запускать на сервере, тестировать в браузере, интегрировать с Telegram.

**Подробное описание файлов  
  
index.html**

Это основной HTML-документ MiniApp-приложения из примера «УчисьДелись». Он задаёт структуру веб-страницы.

Документ начинается с объявления типа (DOCTYPE) и установки языка (ru).

В секции <head> задаются метатеги кодировки и адаптивности, подключается внешний файл стилей styles.css, указывается заголовок страницы («УчисьДелись»).

Тело документа (<body>) разделено на несколько секций:

* Header — заголовок приложения.
* Main — контентная область с меню, в котором есть кнопки «Предложить навык», «Найти наставника», «О проекте», «Контакты».
* Footer — информация о правах: © УчисьДелись 2025.

**styles.css**

Это внешний файл стилей, который определяет вид интерфейса (цвета, шрифты, отступы и адаптивность).

Файл отвечает за визуальное оформление интерфейса. Он делает страницу привлекательной и удобной для пользователя, что важно для UX/UI.

**main.py**

Это серверный скрипт на языке Python, использующий фреймворк Flask для включения веб-сервера. Файл отвечает за серверную часть приложения, обработку запросов и передачу HTML-страницы пользователю. Он позволяет запустить MiniApp как полноценное веб-приложение.

Инициализация приложения происходит через объект Flask (с указанием текущей директории как каталога с шаблонами страниц приложения).

Основной маршрут (/) возвращает HTML-страницу из файла index.html через функцию render\_template. Дополнительный маршрут для файла стилей (/styles.css) обеспечивает корректную подачу CSS-файла через функцию send\_from\_directory. Таким образом информация из этого файла будет доступна браузеру, а цвета и стили будут корректно отображены.

Приложение запускается со включённым режимом отладки (debug=True) на хосте 0.0.0.0 и порту 80. Это делает приложение доступным в локальной сети и на сервере.

**requirements.txt**

Это файл, содержащий список того, что нужно для работы приложения.

В нашем примере файл requirements.txt содержит один пакет, Flask==3.0.3. Это фреймворк Flask версии 3.0.3 для запуска веб-сервера.

При развёртывании приложения, например, на платформе Amvera, этот файл используется для установки нужной версии Flask через менеджер пакетов (pip).

1. **Подведение итогов (5 минут)**

**Резюме урока**

Кратко повторить основные моменты: важность UX/UI, роль ИИ-инструментов, понятие MVP, итеративный процесс разработки.

**Обсуждение**

Задать вопросы:

* Что нового вы узнали?
* Какие моменты оказались наиболее интересными?
* Как можно улучшить предложенные прототипы?

**Обратная связь**

Предложить обучающимся поделиться впечатлениями и обсудить, какие идеи можно развить в дальнейшем.

**Заключение**

Подвести итоги, отметив творческий подход и активное участие.

При необходимости обозначить домашнее задание: например, доработать прототип, учитывая полученные комментарии, и написать краткое эссе о применении инструментов ИИ в разработке приложений.

*Урок подготовлен экспертами Академии искусственного интеллекта для школьников Благотворительного фонда Сбербанка «Вклад в будущее».*

*Академия ИИ для школьников — Всероссийский образовательный проект, реализуемый Благотворительным фондом «Вклад в будущее» при поддержке Сбера с 2018 года. Цель проекта — формирование интереса и развитие прикладных навыков у школьников в сфере искусственного интеллекта. В рамках Академии ИИ проводятся просветительские активности, организуются соревнования, создаются образовательные курсы и вводные уроки, а также ведется работа с сообществом педагогов и амбассадоров Академии ИИ.*

*Все материалы опубликованы в открытом доступе на сайте Академии искусственного интеллекта для школьников и могут быть использованы для образовательных некоммерческих целей.*

*Ссылка:* [*https://ai-academy.ru*](https://ai-academy.ru)