

АВТОРСКАЯ ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА

ТЕХНОЛОГИИ БУДУЩЕГО

Понимать. Создавать.

Практические модули по программированию, искусственному интеллекту,
цифровым продуктам и кибербезопасности

Автор и ведущий	Руслан Бексултанов — Senior Flutter-разработчик, 6+ лет коммерческого опыта, преподаватель Compass College
Основная аудитория	Ученики 7–11 классов; отдельные форматы — для педагогов, родителей и корпоративных команд
Формат	Практические воркшопы по 90 минут, проектный интенсив и интерактивные Tech Talks
Редакция	2026 / содержание актуализируется перед каждым запуском

Программа связывает фундаментальные знания школьной информатики с практиками и инструментами, которые используются в современной IT-индустрии.

1. Паспорт программы

Название	«Технологии будущего: понимать и создавать»
Направления	Программирование, AI literacy, цифровые продукты, кибербезопасность, профориентация
Возраст	Основная программа — 13–17 лет; конкретный модуль адаптируется под класс и уровень
Язык	Русский; отдельные элементы и профессиональная терминология — на английском
Группа	Практика — до 15 учеников; Tech Talk — 50–200 слушателей
Продолжительность	Один воркшоп — 90 минут; модуль — 4–5 воркшопов; Tech Talk — 60–90 минут
Форма	Очно; возможно смешанное сопровождение материалами и заданиями
Итог	Работающий артефакт: программа, модель, прототип, опубликованный проект, аудит безопасности или индивидуальный план развития

2. Место программы в школьном обучении

Ключевой принцип. Программа не заменяет школьную информатику. Она использует полученную учениками базу и показывает, как фундаментальные концепции работают в актуальных коммерческих продуктах.

Школьный курс формирует необходимые основы: алгоритмическое мышление, работу с информацией, базовый синтаксис, устройство компьютера и цифровую грамотность. Авторские модули добавляют индустриальный контекст: клиент-серверные приложения, мобильная разработка, API, базы данных, современные AI-инструменты, продуктовый цикл и практики цифровой безопасности.

Содержание прикладных блоков актуализируется перед каждым запуском. Это особенно важно для искусственного интеллекта: инструменты, модели, интерфейсы и правила безопасного использования меняются быстрее, чем традиционная длительная учебная программа.

В качестве отраслевых примеров используются публично наблюдаемые механики современных приложений и обобщённый опыт коммерческой разработки, включая банковские

продукты. Внутренний код, данные пользователей и конфиденциальная информация компаний не используются.

3. Цель и образовательные задачи

Цель программы — помочь подросткам сформировать целостное и реалистичное представление о современной IT-индустрии, научиться осмысленно использовать цифровые инструменты и получить опыт создания собственного продукта.

Задачи программы:

- связать школьные знания по информатике с реальными сценариями разработки приложений;
- снять страх перед кодом и технологической сложностью;
- научить раскладывать проблему на последовательность проверяемых шагов;
- показать возможности и ограничения современных AI-систем;
- сформировать навыки проверки информации, защиты данных и безопасной работы в интернете;
- дать опыт командной разработки, публикации и презентации цифрового продукта;
- помочь ученику осознанно оценить интерес к IT-профессиям.

4. Планируемые результаты обучения

После прохождения основной программы ученик сможет:

- объяснить, как программа превращается в последовательность инструкций для компьютера;
- использовать переменные, условия, циклы и функции для решения небольшой задачи;
- различать интерфейс, бизнес-логику, сервер, API и базу данных;
- проследить путь пользовательского действия в современном приложении;
- объяснить базовый принцип машинного обучения и роль данных;
- объяснить, почему языковая модель может ошибаться и зависеть от контекста;
- составить содержательный запрос к AI-инструменту и проверить полученный результат;
- распознать основные признаки фишинга, социальной инженерии и манипулятивного контента;
- создать в команде прототип цифрового продукта и опубликовать его в интернете;
- кратко представить проблему, решение и результат своей работы.

5. Методология

Программа основана на Project-Based Learning и обучении через действие. Концепции вводятся короткими циклами: объяснение → демонстрация → практическая задача → обсуждение

результата. Не менее 50 минут 90-минутного воркшопа занимает самостоятельная или командная деятельность учеников.

1. Контекст и вопрос — 5–10 минут. Зачем тема нужна и где она встречается в жизни.
2. Короткий блок концепции — 10–15 минут. Только необходимая теория.
3. Демонстрация реального примера — 10–15 минут.
4. Практическая работа — 45–55 минут с промежуточными контрольными точками.
5. Рефлексия и проверка результата — 10 минут.

Продолжительность отдельных этапов может меняться в зависимости от возраста, уровня группы и технической сложности задания.

6. Входная диагностика и адаптация

До начала модуля проводится короткая диагностика: класс, изученные темы, языки программирования, наличие школьных аккаунтов, ограничения на внешние сервисы и уровень уверенности учеников за компьютером. При возможности программа согласуется с преподавателем информатики.

- Начальный уровень: готовые шаблоны, визуальные инструменты, работа парами, минимальный объём синтаксиса.
- Средний уровень: самостоятельные изменения кода, API и простая работа с данными.
- Продвинутый уровень: расширенные требования, декомпозиция, тестирование, защита архитектурного решения.

7. Учебный модуль 1. Как работает программирование

Рекомендуемые классы	7–9
Объём	5 воркшопов по 90 минут
Подготовка	Не требуется; уверенная работа с браузером и файлами желательна
Итог модуля	Мини-программа и опубликованный учебный веб-проект

Модуль даёт цельную картину: от выполнения инструкции компьютером до обмена данными между интерфейсом и сервером. Акцент сделан не на заучивании языка, а на причинно-следственных связях.

1.1. Алгоритм и выполнение программы

Цель занятия. Показать, как задача превращается в точную последовательность команд.

По завершении ученик сможет:

- отличить задачу от алгоритма;
- записать алгоритм с условиями;
- найти неоднозначность или пропущенный шаг;
- перевести бытовой алгоритм в псевдокод.

Ключевые понятия. алгоритм, команда, последовательность, условие, состояние, ввод и вывод.

Связь с индустрией. Почему приложение должно однозначно обрабатывать каждый пользовательский сценарий; примеры регистрации, перевода и восстановления доступа.

Практическая работа. Команды проектируют сценарий регистрации на школьное мероприятие, тестируют его на исключениях и переводят в псевдокод.

Результат	Проверенный алгоритм и блок-схема.
Проверка результата	Алгоритм имеет начало, результат, обработку обязательных условий и минимум два исключения.
Инструменты	Доска или онлайн-доска, браузер, редактор схем.

1.2. Код: данные, условия и функции

Цель занятия. Дать первый опыт написания и изменения исполняемого кода.

По завершении ученик сможет:

- создать переменную и выбрать подходящий тип данных;
- написать условие if/else;
- вынести повторяемую проверку в функцию;
- прочитать сообщение об ошибке и исправить простой дефект.

Ключевые понятия. переменные, строки, числа, логические значения, условия, функции, ошибка выполнения.

Связь с индустрией. Валидация номера телефона, возраста и обязательных полей в реальных формах.

Практическая работа. Создание валидатора регистрации: возраст, формат телефона, наличие согласия и свободных мест.

Результат	Работающая консольная или браузерная мини-программа.
Проверка результата	Программа корректно обрабатывает минимум пять тестовых сценариев.

Инструменты

Браузерная среда программирования или заранее настроенный редактор.

Самостоятельная работа. Добавить собственное правило проверки и объяснить его назначение.

1.3. Из чего состоит современное приложение

Цель занятия. Сформировать системное представление о слоях цифрового продукта.

По завершении ученик сможет:

- различать интерфейс, данные и бизнес-логику;
- объяснить роль клиента и сервера;
- описать путь действия пользователя;
- найти место, где должна выполняться конкретная проверка.

Ключевые понятия. UI, клиент, сервер, бизнес-логика, состояние, база данных.

Связь с индустрией. Разбор публичного пользовательского сценария мобильного банка: вход, просмотр баланса, перевод, получение статуса операции.

Практическая работа. Ученики собирают карту компонентов приложения и связывают каждое действие с нужным слоем.

Результат

Схема архитектуры знакомого приложения.

Проверка результата

Ученик корректно объясняет назначение каждого компонента и путь одного запроса.

Инструменты

Проектор, карточки компонентов, готовое демонстрационное приложение.

1.4. Сеть, API и данные

Цель занятия. Показать, как приложения получают и отправляют данные.

По завершении ученик сможет:

- объяснить запрос и ответ;
- прочесть простой JSON;
- получить данные из учебного API;
- обработать успешный ответ и ошибку.

Ключевые понятия. HTTP, запрос, ответ, API, JSON, код состояния, задержка сети.

Связь с индустрией. Почему мобильное приложение не хранит актуальный баланс только на телефоне; как интерфейс получает данные и сообщает об ошибке.

Практическая работа. Отправка запроса к безопасному учебному API, чтение ответа и вывод выбранных полей в интерфейсе.

Результат	Экран или страница с данными из API и состоянием ошибки.
Проверка результата	Данные отображаются корректно; предусмотрены загрузка, успех и ошибка.
Инструменты	Браузер, учебный API, стартовый проект.

1.5. От изменения к публикации

Цель занятия. Познакомить с циклом выпуска небольшой версии продукта.

По завершении ученик сможет:

- сформулировать небольшое изменение;
- внести и проверить изменение в готовом проекте;
- использовать AI-помощника как инструмент, а не источник истины;
- опубликовать проект и проверить ссылку.

Ключевые понятия. требование, версия, тестирование, ошибка, деплой, обратная связь.

Связь с индустрией. Как команда превращает запрос пользователя в изменение приложения; почему даже небольшой релиз проходит проверку.

Практическая работа. Команды изменяют готовый мини-проект, проверяют основные сценарии и публикуют результат.

Результат	Опубликованный учебный веб-проект.
Проверка результата	Работает основная функция, отсутствуют блокирующие ошибки, ссылка открывается на другом устройстве.
Инструменты	Стартовый репозиторий или web-редактор, AI-инструмент, учебный хостинг.

8. Учебный модуль 2. Искусственный интеллект без магии

Рекомендуемые классы	8–11
Объём	5 воркшопов по 90 минут
Подготовка	Не требуется; задания усложняются для технических классов

Итог модуля

Проверенный прототип AI-функции и памятка безопасного использования

Модуль развивает AI literacy: понимание принципов, практическое применение, проверку результатов и безопасное поведение. Конкретные сервисы и демонстрации обновляются перед запуском.

2.1. Что называют искусственным интеллектом

Цель занятия. Отделить базовые понятия от маркетинговых обещаний и сформировать карту AI-технологий.

По завершении ученик сможет:

- различать правила, машинное обучение, нейросети и генеративный ИИ;
- подобрать тип системы под задачу;
- назвать ограничения AI-решения;
- объяснить, почему ChatGPT — лишь один из видов AI-продуктов.

Ключевые понятия. AI, машинное обучение, нейронная сеть, генеративная модель, классификация, рекомендация.

Связь с индустрией. Рекомендации, антифрод, распознавание документов, поиск, генерация текста и кода в современных приложениях.

Практическая работа. Классификация набора реальных продуктовых задач: где достаточно правил, где требуется ML, а где генеративная модель.

Результат

Карта «задача → подход → риск».

Проверка результата

Для каждого кейса выбран обоснованный подход и назван минимум один риск.

Инструменты

Карточки кейсов, проектор, браузер.

2.2. Как модель учится на данных

Цель занятия. Показать роль данных, признаков и проверки качества модели.

По завершении ученик сможет:

- разделить примеры на обучающую и проверочную выборки;
- выбрать признаки для классификации;
- объяснить переобучение на простом примере;
- сравнить ошибочные и правильные предсказания.

Ключевые понятия. данные, признак, метка, обучение, тестирование, классификация, переобучение, смещение.

Связь с индустрией. Почему качество антифрода, рекомендаций и распознавания зависит от состава данных.

Практическая работа. Ученики собирают небольшой датасет и обучают визуальную модель различать два класса объектов или сигналов.

Результат	Мини-модель и таблица ошибок.
Проверка результата	Модель проверена на новых примерах; ученик объясняет хотя бы одну ошибку и способ улучшения данных.
Инструменты	Визуальный ML-инструмент, камера или подготовленный датасет.

2.3. Как работает языковая модель

Цель занятия. Дать рабочую ментальную модель генерации текста без избыточной математики.

По завершении ученик сможет:

- объяснить роль токенов и контекста;
- описать вероятностное продолжение;
- вызвать и распознать неподтверждённый ответ;
- улучшить запрос с помощью контекста и критериев.

Ключевые понятия. токен, контекст, вероятность, языковая модель, системная инструкция, галлюцинация.

Связь с индустрией. Использование LLM для анализа требований, черновиков, кода и поддержки; необходимость проверки человеком.

Практическая работа. Серия контролируемых экспериментов: изменение контекста, ограничений и примеров; фиксация различий в ответах.

Результат	Журнал экспериментов с выводами.
Проверка результата	Ученик объясняет, почему ответы изменились, и отделяет подтверждённые факты от предположений модели.
Инструменты	Доступный школе AI-инструмент, подготовленный набор запросов.

2.4. Безопасность, дипфейки и ответственное использование

Цель занятия. Научить оценивать риски при работе с AI-контентом и личными данными.

По завершении ученик сможет:

- определить, какие данные нельзя передавать внешнему сервису;
- проверить подозрительный материал по нескольким признакам;
- объяснить, почему визуальная правдоподобность не является доказательством;
- составить безопасный учебный сценарий использования ИИ.

Ключевые понятия. персональные данные, конфиденциальность, дипфейк, источник, фактчекинг, авторство, человеческий контроль.

Связь с индустрией. Риски загрузки клиентских данных, исходного кода и документов во внешние AI-сервисы; политика доступа в компаниях.

Практическая работа. Разбор кейсов: поддельное сообщение, синтетическое изображение, сомнительный ответ модели и запрос с персональными данными.

Результат

Чек-лист проверки AI-контента и безопасного запроса.

Проверка результата

В чек-листе есть проверка источника, данных, цели, доказательств и последствий ошибки.

Инструменты

Проектор, подготовленные безопасные кейсы, браузер.

2.5. Прототип AI-функции

Цель занятия. Пройти путь от пользовательской проблемы до проверяемого AI-прототипа.

По завершении ученик сможет:

- сформулировать задачу и критерий полезности;
- выбрать роль AI в продукте;
- создать цепочку запросов или подключить учебный API;
- проверить результат на нормальных и сложных примерах;
- описать ограничения прототипа.

Ключевые понятия. пользовательская задача, промпт, API, критерий качества, тестовый сценарий, ограничение.

Связь с индустрией. Как AI-функции добавляются в существующий продукт: интерфейс, запрос к модели, обработка ответа, ограничения и мониторинг.

Практическая работа. Команды создают прототип помощника для учёбы, классификатора обращений или другого согласованного кейса.

Результат

Демонстрируемый AI-прототип и карточка ограничений.

Проверка результата	Прототип решает заявленный сценарий, протестирован минимум на пяти примерах и не скрывает ограничения.
Инструменты	Визуальный конструктор или стартовый проект, разрешённый AI-сервис, учебный API при необходимости.

9. Учебный модуль 3. Создай свой цифровой продукт

Рекомендуемые классы	9–11
Объём	4 воркшопа по 90 минут
Формат	Команды по 2–3 человека; до 15 учеников
Подготовка	Уверенная работа с браузером; базовое программирование желательно, но не обязательно
Итог модуля	Опубликованный прототип и трёхминутная защита

Модуль предназначен для школьной проектной деятельности. Главный результат — не количество изученных инструментов, а завершённый цикл: проблема → решение → прототип → проверка → публикация → презентация.

3.1. Проблема, пользователь и прототип

Цель занятия. Выбрать достаточно узкую проблему и спроектировать проверяемое решение.

По завершении ученик сможет:

- описать конкретного пользователя;
- отделить проблему от предполагаемого решения;
- сформулировать основной сценарий;
- создать кликабельный или бумажный прототип;
- получить обратную связь.

Ключевые понятия. проблема, пользователь, гипотеза, пользовательский путь, интерфейс, прототип.

Связь с индустрией. Как продуктовые команды проверяют идею до дорогой разработки.

Практическая работа. Команды выбирают проблему школьной или городской среды, рисуют путь пользователя и собирают прототип ключевых экранов.

Результат	Карта проблемы, пользовательский сценарий и прототип.
------------------	---

Проверка результата	Другой ученик может пройти основной сценарий без объяснения автора.
Инструменты	Figma или бумажные шаблоны, доска, браузер.

3.2. Интерфейс и логика

Цель занятия. Превратить прототип в работающую первую версию.

По завершении ученик сможет:

- разделить продукт на небольшие задачи;
- создать основные экраны;
- реализовать ключевой сценарий;
- использовать AI-помощника с конкретными требованиями;
- проверить работу интерфейса.

Ключевые понятия. MVP, компонент, состояние, событие, требование, итерация.

Связь с индустрией. Как разработчик получает задачу, уточняет критерии и вносит изменение без разрушения остальных сценариев.

Практическая работа. Создание интерфейса по прототипу и реализация основной логики переходов и действий.

Результат	Работающая локальная версия ключевого сценария.
Проверка результата	Основной сценарий завершается; элементы имеют понятные состояния и подписи.
Инструменты	Web-конструктор или стартовый проект, AI-помощник, браузер.

3.3. Данные и умная функция

Цель занятия. Добавить продукту сохранение данных или внешнюю интеграцию.

По завершении ученик сможет:

- определить минимальную структуру данных;
- подключить разрешённое хранилище или API;
- обработать загрузку и ошибку;
- добавить одну полезную AI- или data-функцию;
- не размещать секретные ключи в открытом коде.

Ключевые понятия. модель данных, хранилище, API, интеграция, секрет, состояние загрузки.

Связь с индустрией. Почему данные, доступы и обработка ошибок определяют надёжность реального приложения.

Практическая работа. Команды сохраняют пользовательские данные или подключают учебный API, затем добавляют одну функцию, усиливающую основную ценность продукта.

Результат	Версия продукта с данными или интеграцией.
Проверка результата	Функция воспроизводима, ошибки не блокируют весь продукт, чувствительные данные не публикуются.
Инструменты	Supabase/Firebase или учебный API по согласованию, стартовый проект.

3.4. Тестирование, релиз и питч

Цель занятия. Завершить продукт, опубликовать его и представить аудитории.

По завершении ученик сможет:

- провести проверку по тестовым сценариям;
- исправить блокирующие дефекты;
- опубликовать проект;
- сформулировать проблему, решение и демонстрацию;
- ответить на вопросы об ограничениях.

Ключевые понятия. тестовый сценарий, дефект, релиз, обратная связь, питч, демонстрация.

Связь с индустрией. Как команды принимают решение о готовности версии и объясняют ценность продукта заинтересованным сторонам.

Практическая работа. Финальное тестирование, деплой и Demo Day: трёхминутная презентация каждой команды.

Результат	Публичная ссылка, демонстрация и краткая карточка проекта.
Проверка результата	Основной сценарий работает по ссылке; команда укладывается во время и честно называет ограничения.
Инструменты	Учебный хостинг, проектор, форма обратной связи.

10. Отдельный мастер-класс. Кибербезопасность в эпоху ИИ

Аудитория	7–11 классы; адаптированная версия — родители, педагоги и сотрудники организаций
------------------	--

Продолжительность	90 минут
Формат	Интерактивное объяснение, безопасные симуляции, индивидуальный аудит настроек
Подготовка	Не требуется
Итог	Персональный план усиления цифровой безопасности

Цель — сформировать практические привычки, которые снижают риск фишинга, кражи аккаунтов, утечки данных и манипуляций с использованием AI-контента.

Результаты обучения:

- объяснить разницу между паролем, двухфакторной аутентификацией и кодом восстановления;
- выявить типовые признаки фишингового сообщения и поддельной страницы;
- проверить адрес отправителя, домен, контекст и срочность запроса;
- назвать данные, которые нельзя передавать незнакомцу или AI-сервису;
- составить безопасный алгоритм действий при подозрении на взлом.

Практические станции:

6. Пароли и 2FA: оценка собственных привычек без раскрытия паролей ведущему или группе.
7. Фишинг: разбор подготовленных примеров и поиск признаков манипуляции.
8. Дипфейки и AI-контент: проверка происхождения, источника и контекста.
9. План защиты: три конкретных действия, которые ученик выполнит после занятия.

Безопасность проведения. На занятии не выполняется реальная атака, не собираются пароли, коды, персональные данные или доступы учеников. Все симуляции используют учебные материалы и вымышленные аккаунты.

11. Массовый формат. Tech Talk

Тема: «Профессии будущего: как подготовиться к рынку труда в эпоху ИИ»

Аудитория	8–11 классы, 50–200 человек
Продолжительность	60–90 минут
Оборудование	Проектор, экран, звук, презентационный компьютер, интернет желательно
Формат	Динамичная лекция, демонстрации, вопросы залу, короткие голосования

Итог

Индивидуальный чек-лист развития на ближайшие 6–12 месяцев

Содержание:

- что современные AI-системы действительно умеют сегодня и где проходит граница демонстрации и надёжного продукта;
- какие рабочие задачи уже автоматизируются в разработке, дизайне, маркетинге, аналитике, образовании и других областях;
- какие сценарии развития возможны и почему невозможно честно назвать точный список «исчезающих профессий»;
- какие навыки сохраняют ценность: фундаментальные знания, системное мышление, коммуникация, ответственность и проверка результата;
- как школьнику выбрать направление, собрать первые проекты и проверить собственный интерес без преждевременного выбора профессии.

12. Оценивание и обратная связь

Оценивание формирующее: оно помогает ученику увидеть прогресс и следующую точку развития, а не ранжирует детей между собой. Для кратких форматов используется проверка артефакта и рефлексия; для проектного модуля — рубрика.

Критерий	Начальный	Базовый	Уверенный
Проблема	Сформулирована широко	Понятны пользователь и задача	Есть проверенная гипотеза и ограничения
Работоспособность	Есть отдельные элементы	Работает основной сценарий	Обработаны ошибки и несколько сценариев
Применение знаний	Нужна постоянная помощь	Ключевые решения объяснены	Решения обоснованы и улучшены после теста
Безопасность	Риски не учтены	Соблюдены основные правила	Риски названы и предусмотрены меры
Презентация	Идея описана частично	Проблема и решение понятны	Есть демонстрация, доказательства и честные ограничения

13. Рекомендации ученику и родителям

При длительном формате ведущий может предоставить краткую индивидуальную обратную связь: сильные стороны, наблюдаемые затруднения и рекомендуемый следующий шаг. Это не психологическая диагностика и не обещание профессионального успеха.

- интерес к логике и программированию;
- самостоятельность при поиске ошибки;
- умение объяснить принятое решение;
- работа в команде и реакция на обратную связь;
- способность довести небольшой проект до демонстрируемого результата.

14. Организационные и технические требования

Компьютеры	Один компьютер на 1–2 учеников; заранее проверенные браузер и доступ к необходимым сайтам
Интернет	Стабильное подключение; желательно отдельное тестирование гостевой сети до занятия
Аудитория	Проектор или интерактивная панель, доска, возможность свободно работать в командах
Аккаунты	Создаются заранее только при необходимости и с учётом правил школы и возрастных ограничений сервисов
Доступы	Ученики не передают ведущему личные пароли, банковские данные, коды подтверждения или закрытые документы
Подготовка школы	Назначить координатора, подтвердить состав группы, сообщить ограничения сети и установленного ПО
Подготовка ведущего	Адаптировать задания, предоставить стартовые файлы, проверить демонстрации и резервный сценарий без интернета

15. Цифровая безопасность и этика

- Для практики используются учебные, обезличенные или синтетические данные.
- Не публикуются фамилии, контакты и другие персональные данные учеников без согласованной необходимости.
- При работе с AI-сервисами обсуждаются правила школы, возрастные ограничения и условия конкретного инструмента.

- Сгенерированный результат проверяется человеком; ИИ не представляется как безошибочный источник.
- Кибербезопасность преподаётся в защитном контексте; реальные атаки и получение несанкционированного доступа не выполняются.
- Публичная публикация проектов проводится только после проверки контента и отсутствия секретов в коде.

16. Варианты комплектации программы

Формат	Для кого	Продолжительность	Результат
Ознакомительный	Большая аудитория	1 Tech Talk	Кругозор и личный план действий
Практический	Группа до 15	1 воркшоп	Один завершённый учебный артефакт
Тематический модуль	7–11 классы	4–5 воркшопов	Система знаний и итоговая работа
Проектный интенсив	9–11 классы	4 воркшопа	Опубликованный командный продукт
Школьный клуб	Мотивированная группа	Регулярные занятия	Портфолио проектов и углубление

17. Порядок запуска в школе

10. Короткая встреча с руководителем или координатором: цели, классы, расписание и ограничения.
11. Сверка с программой информатики и выбор уровня сложности.
12. Подтверждение формата, количества участников и технических требований.
13. Проверка аудитории, сети, аккаунтов и демонстрационных материалов.
14. Проведение занятия или модуля с текущей обратной связью.
15. Итоговая демонстрация, сбор отзывов и рекомендации по продолжению.

18. Об авторе и ведущем

Руслан Бексултанов — Senior Flutter-разработчик с более чем шестилетним опытом коммерческой разработки и преподаватель Compass College. Работал над мобильными продуктами, включая банковские приложения, участвовал в проектировании архитектуры,

разработке пользовательских сценариев, интеграциях, выпуске приложений и работе команд разработки.

В образовательной программе профессиональный опыт используется для объяснения общих принципов современной разработки и публично наблюдаемых механик цифровых продуктов. Примеры адаптируются под возраст и не содержат конфиденциальной информации работодателей или пользователей.

Контакты. Сайт: <https://edu.hostme.live> • Telegram: <https://t.me/brevol> • WhatsApp: +996 707 501 488