

МИНИСТЕРСТВО ОБРАЗОВАНИЯ ПРИМОРСКОГО КРАЯ
Краевое государственное автономное учреждение дополнительного
образования «Региональный модельный центр Приморского края»

Рассмотрена на заседании
Педагогического совета
КГАУ ДО «РМЦ Приморского края»
Протокол № 2
от «25» 05 2026

Утверждаю

И.о. директора КГАУ ДО
«РМЦ Приморского края»

Ю.В. Шукурова

2026



ПРОГРАММИРОВАНИЕ: ОТ БЛОКА КОДА ДО ИГРОВОГО ДВИЖКА. ПРОЕКТНАЯ ДЕЯТЕЛЬНОСТЬ

Дополнительная общеобразовательная общеразвивающая программа
технической направленности

Возраст учащихся: 12-17 лет

Объём программы: 1 год

Веригин Александр Михайлович,
педагог дополнительного
образования
Казарян Алёна Анатольевна,
методист

Владивосток
2026

Раздел № 1. ОСНОВНЫЕ ХАРАКТЕРИСТИКИ ПРОГРАММЫ

1.1. Пояснительная записка

Актуальность программы

Первый год обучения позволил обучающимся освоить базовый синтаксис языков программирования и принципы работы в игровых движках через решение изолированных учебных кейсов. Второй же год носит выраженный практико-ориентированный характер и направлен на преодоление разрыва между «умением писать код» и «умением создавать программные продукты». Центральным элементом обучения является годовая работа над комплексным проектом: каждая подгруппа разрабатывает собственный продукт, проходя полный цикл создания программного обеспечения.

Актуальность программы второго года обусловлена необходимостью формирования у обучающихся компетенций уровня Junior-разработчика. Переход от кейсового к проектному подходу моделирует реальные условия работы в IT-компаниях, где задачи не имеют единственно верного алгоритмического решения, а требуют проектного мышления, итеративной разработки и постоянной обратной связи.

Ключевым отличием второго года является внедрение методологии гибкой разработки (Agile/Scrum) в образовательный процесс. Обучающиеся учатся декомпозировать крупные задачи, вести техническую документацию, использовать системы контроля версий (Git) в командной среде и проводить код-ревью. Это позволяет трансформировать разрозненные навыки программирования (полученные в Scratch, Python и Godot) в целостную профессиональную культуру разработки.

Таким образом, программа второго года обеспечивает качественный скачок в профессиональном становлении обучающихся: от исполнителя отдельных задач к полноправному участнику команды разработчиков, способному создать сложный, законченный цифровой продукт.

Направленность программы – техническая.

Язык реализации – государственный язык РФ – русский.

Уровень освоения – базовый.

Отличительные особенности

Программа рассчитана на обучающихся, успешно завершивших годовой вводный курс. Для зачисления необходимо владеть стартовыми навыками программирования Scratch, Python и работы с движком Godot.

Ключевая особенность программы - последовательная проектная работа в течение всего учебного года. Команды проходят полный цикл создания проекта: разработку прототипа, экспертную оценку, пилотное тестирование и итоговую защиту. Темы проектов согласуются с представителями реальных отраслей Приморского края, что гарантирует их практическую применимость.

Конкурсы и хакатоны не являются дополнительными активностями, а встроены в учебный план как контрольные точки. Участники получают системное сопровождение: отраслевое менторство, отработку питчей, формирование портфолио и доведение демоверсий до уровня, готового к внедрению. В результате выпускники выходят на рынок не с учебными прототипами, а с проверенными продуктами, адаптированными под специфику региональной экономики.

Адресат программы:

Программа предназначена для обучающихся Приморского края в возрасте от 12 до 17 лет, прошедших обучение по вводной программе «Программирование от блока кода до игрового движка».

Особенности организации образовательного процесса

Учебные группы формируются в количестве 13-15 человек. Программа реализуется в очном формате. Занятия проводятся – 2 раза в неделю по 2 академических часа с пятнадцатиминутным перерывом, что определяется требованиями Санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи».

Объем программы: 144 академических часа (2 занятия в неделю по 2 академических часа каждое).

Срок реализации: 1 год.

Используются методы проблемного обучения, разбор реальных бизнес-кейсов и архитектурных решений, анализ технических инцидентов, защита проектов.

1.2. Цель и задачи программы

Цель программы – Формирование базовых навыков создания цифровых проектов у обучающихся 12-17 лет.

Задачи программы

Воспитательные:

1. Воспитывать высокую степень профессиональной ответственности за результат командной работы, формируя понимание того, что создаваемый цифровой продукт предназначен для реальных пользователей и будет оцениваться экспертами индустрии.

2. Формировать навыки командного взаимодействия и культуру профессиональной коммуникации, необходимые для работы в IT-командах (в том числе навыки конструктивной критики на код-ревью и аргументированного отстаивания технических решений).

3. Формировать интерес к развитию технологического потенциала Приморского края и чувство причастности к профессиональному IT-сообществу региона через участие в открытых защитах проектов, хакатонах и взаимодействие с экспертами.

Развивающие:

1. Развивать системное и архитектурное мышление через решение сложных практических задач, требующих проектирования масштабируемой структуры приложений и прогнозирования последствий технических решений.

2. Развивать навыки проектного управления в условиях неопределённости, включая декомпозицию сложных задач, работу со спринтами, управление рисками и адаптацию планов при изменении требований к продукту.

3. Развивать коммуникативные навыки и способность к публичной презентации сложных технических решений перед аудиторией, стейкхолдерами

и представителями экспертного сообщества.

Обучающие:

1. Формировать навыки этичного и эффективного использования инструментов искусственного интеллекта в проектной деятельности.

2. Обучать основам проектного управления: постановке цели, планированию этапов, распределению ролей в команде и контролю сроков реализации.

3. Формировать навыки итеративной разработки прототипов: последовательного улучшения продукта на основе тестирования и экспертной обратной связи.

4. Обучать подготовке и проведению презентаций проектов, выступлений для жюри, наставников и отраслевых экспертов.

5. Формировать навыки проектной рефлексии: анализа успехов и ошибок, внесения корректировок в рабочий процесс и планирования дальнейшего развития продукта.

1.3. Содержание программы

Учебный план

№ п/п	Название раздела, темы	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
1	Основы работы с Искусственным Интеллектом	22	8	14	
1.1	Техника безопасности в компьютерном классе. Организация рабочего места	2	2	0	Устный опрос
1.2	Введение в генеративный ИИ. Промпт-инжиниринг для программистов	6	2	4	Практическая работа
1.3	ИИ как помощник в кодировании: рефакторинг, поиск багов, генерация прототипов	6	2	4	Проверочная работа
1.4	Генерация контента и этика использования ИИ. Авторское право и безопасность данных	8	2	6	Творческая работа
2	Проектная деятельность	122	28	94	

2.1	Знакомство с проектной деятельностью. Способы генерации идей. Мозговой штурм	4	1	3	Творческая работа
2.2	Способы поиска информации. Источники информации	4	1	3	Проверочная работа
2.3	Основы командной работы и инструменты для командной разработки проекта	4	1	3	Практическая работа
2.4	Распределение ролей в команде. Настройка систем контроля версий (Git) и таск-трекеров	4	2	2	Практическая работа
2.5	Анализ продуктов. Поиск проблемы. Выявление актуальности	4	1	3	Устный опрос
2.6	Постановка цели и задач проекта. Разработка концепции	6	2	4	Творческое задание
2.7	Поиск аналогов. Формирование гипотез и ожиданий от продукта	4	1	3	Практическая работа
2.8	Создание и заполнение проектной карты	6	1	5	Контрольная работа
2.9	Основы взаимодействия с заказчиком	4	1	3	Творческая работа
2.10	Сбор требований и составление Технического задания. Декомпозиция задач	6	2	4	Практическая работа
2.11	Проектирование архитектуры приложения и выбор технологического стека	6	2	4	Проверочная работа
2.12	Планирование спринтов. Тайм-менеджмент в разработке	4	1	3	Творческое задание
2.13	Создание прототипа основных механик (MVP). Программирование базовой логики	12	2	10	Практическая работа
2.14	Интеграция контента, UI/UX и настройка физики/логики	14	2	12	Наблюдение
2.15	Продвинутые механики, оптимизация производительности и сетевое взаимодействие	16	4	12	Проверочная работа

2.16	Тестирование и отладка (QA). Работа с баг-трекером. Полировка продукта	12	2	10	Контрольная работа
2.17	Подготовка к релизу: экспорт, упаковка, создание презентации и документации	10	2	8	Практическая работа
2.18	Итоговая защита года. Презентация продукта экспертам и партнерам	2	0	2	Публичная защита проекта
Итого:		144	34	110	

Содержание учебного плана

1. Раздел: Основы работы с Искусственным Интеллектом

1.1. Тема: Техника безопасности в компьютерном классе. Организация рабочего места

Теория. Требования СанПиН к работе за ПК. Электробезопасность, эргономика, организация рабочего пространства. Правила поведения в учебной лаборатории и цифровой гигиена.

1.2. Тема: Введение в генеративный ИИ. Промпт-инжиниринг для программистов

Теория. Принципы работы LLM и генеративных моделей. Структура эффективного запроса (контекст, роль, задача, ограничения). Изучение доступных ИИ-инструментов для разработки.

Практика 12-14. Создание простой коллекции промптов для генерации игровых идей и базового кода. Сравнение ответов ИИ на простые запросы в формате викторины.

Практика 15-17. Систематизация библиотеки промптов с тегами, контекстом и ограничениями. Глубокий сравнительный анализ ответов моделей на сложные технические задачи с оценкой точности, структуры и безопасности кода.

1.3. Тема: ИИ как помощник в кодинге: рефакторинг, поиск багов, генерация прототипов

Теория. Инструменты ИИ-ассистентов в среде разработки. Риски использования: «галлюцинации» кода, устаревший синтаксис, уязвимости

безопасности. Принцип «Доверяй, но проверяй».

Практика 12-14. Поиск очевидных ошибок в простом коде с помощью ИИ-ассистента. Замена неоптимальных строк на сгенерированные варианты с обязательной проверкой наставником.

Практика 15-17. Самостоятельный рефакторинг сложных фрагментов кода через ИИ. Критический аудит сгенерированных решений, выявление «галлюцинаций», проверка на уязвимости и исправление логических ошибок.

1.4. Тема: Генерация контента и этика использования ИИ. Авторское право и безопасность данных

Теория. Генерация изображений и звука для цифровых продуктов. Правовые аспекты: авторское право на ИИ-контент, политики платформ распространения. Безопасность персональных данных при работе с облачными сервисами.

Практика 12-14. Генерация простых спрайтов и звуков по готовым шаблонам промптов. Заполнение карточки ассета с указанием нейросети и базовых правил использования.

Практика 15-17. Создание тематического пакета ассетов с единым визуальным стилем. Подготовка полного пакета метаданных, проверка лицензий и составление документации по правовому использованию ИИ-контента.

2. Раздел: Проектная деятельность

2.1. Тема: Знакомство с проектной деятельностью. Способы генерации идей. Мозговой штурм

Теория. Понятие проекта и продуктового подхода. Отличие учебного задания от реального продукта. Методики креативного мышления: мозговой штурм, метод ассоциаций, интеллект-карты. Критерии оценки идеи: новизна, реализуемость силами команды, ресурсоемкость.

Практика 12-14. Мозговой штурм в группах с использованием карточек-подсказок. Создание 5–7 простых идей, выбор одной по голосованию и обсуждению с наставником.

Практика 15-17. Структурированный мозговой штурм с методиками ассоциаций. Генерация 10 концепций, оценка по матрице

«Интерес/Сложность/Ресурсы», аргументированный выбор финальной темы.

2.2. Тема: Способы поиска информации. Источники информации

Теория. Классификация источников: официальная документация, технические форумы, базы знаний, видеоуроки. Навыки формулирования поисковых запросов для нахождения технических решений. Оценка достоверности, актуальности и применимости найденной информации.

Практика 12-14. Поиск решения простой технической задачи по заданному алгоритму. Сравнение видеоурока и официального руководства. Создание личной папки с проверенными ссылками.

Практика 15-17. Самостоятельный поиск реализации алгоритма/эффекта. Критическое сравнение 3+ источников (документация, форумы, статьи). Формирование базы знаний с пометками о достоверности и актуальности.

2.3. Тема: Основы командной работы и инструменты для совместной разработки проекта

Теория: Роли в команде разработчиков: программист, дизайнер интерфейсов, художник, тестировщик, менеджер проекта. Принципы эффективной коммуникации в цифровой среде. Изучение инструментов для общения и управления задачами.

Практика 12-14. Распределение в команды по 3–4 человека с назначением ролей при помощи наставника. Создание общего чата и папки для файлов под контролем преподавателя.

Практика 15-17. Самостоятельное формирование команд, анализ компетенций и распределение ролей. Настройка коммуникационного хаба и облачного хранилища с системой прав доступа и каналами по задачам.

2.4. Тема: Распределение ролей в команде. Настройка систем контроля версий (Git) и таск-трекеров

Теория. Основы работы с системами контроля версий: репозиторий, сохранение изменений (коммит), ветвление, загрузка и выгрузка данных. Важность исключения временных файлов из проекта. Принцип работы трекера задач: создание карточек, назначение исполнителей, статусы выполнения.

Практика 12-14. Создание репозитория по инструкции с помощью

преподавателя. Настройка .gitignore. Добавление первой простой задачи в трекер. Тестовая отправка и получение одного файла.

Практика 15-17. Самостоятельная инициализация репозитория и настройка .gitignore под движок/ЯП. Создание бэклога в трекере, назначение приоритетов. Выполнение первого пулл-реквеста/слияния кода командой.

2.5. Тема: Анализ продуктов. Поиск проблемы. Выявление актуальности

Теория. Понятие целевой аудитории. Какую задачу решает продукт или какую потребность закрывает? Анализ существующих решений на рынке. Формулировка уникального преимущества проекта.

Практика 12-14. Заполнение простого шаблона «Кто будет играть/пользоваться». Описание проблемы своими словами. Краткая устная защита актуальности перед наставником.

Практика 15-17. Детальное описание целевой аудитории (демография, потребности, сценарии). Формулировка проблемы и ценности продукта. Письменное обоснование актуальности с анализом рыночных трендов.

2.6. Тема: Постановка цели и задач проекта. Разработка концепции

Теория. Принципы постановки целей: конкретность, измеримость, достижимость, значимость, ограниченность во времени. Структура документа с описанием концепции: жанр/тип продукта, ключевые функции, визуальный стиль, сюжет или сценарий использования.

Практика 12-14. Заполнение шаблона концепции (0.5–1 стр.). Определение 1 главной цели и 3–4 простых шагов. Обсуждение в группе с фиксацией правок на доске.

Практика 15-17. Написание развернутого концепт-документа (1–2 стр.) по принципу SMART. Декомпозиция цели на измеримые подзадачи. Критический разбор концепции командой с выявлением рисков.

2.7. Тема: Поиск аналогов. Формирование гипотез и ожиданий от продукта

Теория. Конкурентный анализ: прямые и косвенные аналоги. Что можно позаимствовать, а что нужно улучшить. Формирование ожиданий от

пользовательского опыта: как пользователь будет взаимодействовать с продуктом.

Практика 12-14. Поиск 2–3 похожих игр/приложений. Заполнение упрощенной таблицы «Что нравится/Что не работает». Составление списка из 3 обязательных функций для старта.

Практика 15-17. Анализ 3–5 прямых и косвенных аналогов. Заполнение детальной сравнительной матрицы функционала.

2.8. Тема: Создание и заполнение проектной карты

Теория. Структура паспорта проекта: тема, участники, сроки, необходимые ресурсы, потенциальные риски, планируемые результаты. Значение документа для согласования плана работ с руководителем.

Практика 12-14. Заполнение упрощенной карты с помощью шаблона. Планирование этапов на 2–3 недели. Оценка базовых ресурсов. Краткая защита с корректировкой сроков наставником.

Практика 15-17. Самостоятельное заполнение детального паспорта проекта. Построение календарного плана на весь срок. Оценка рисков и ресурсов. Формальная защита карты перед комиссией/преподавателем.

2.9. Тема: Основы взаимодействия с заказчиком

Теория. Роль заказчика в проекте (преподаватель, внешний эксперт или представитель пользователей). Техники выяснения требований: активное слушание, уточняющие вопросы. Управление ожиданиями: как говорить «нет» нереалистичным требованиям.

Практика 12-14. Ролевое обсуждение с упрощенным сценарием. Составление 3–5 базовых вопросов. Презентация идеи наставнику, запись 1–2 советов по доработке.

Практика 15-17. Деловая ролевое обсуждение с реальными ограничениями. Составление структурированного брифа вопросов. Презентация концепции, ведение протокола встречи, фиксация правок и согласование ожиданий.

2.10. Тема: Сбор требований и составление Технического задания.

Декомпозиция задач

Теори.: Структура технического задания: общие сведения, требования к функциям, требования к интерфейсу, технические ограничения. Описание требований с точки зрения пользователя («Пользователь может...»). Приоритизация задач по степени важности.

Практика 12-14. Заполнение бланка ТЗ по образцу. Описание 2–3 сценариев в формате «Я могу...». Разбивка задач на 4–6 простых шагов в трекере. Оценка времени с помощью наставника.

Практика 15-17. Написание структурированного ТЗ с требованиями к UI/UX и функционалу. Описание задач. Декомпозиция в трекере с оценкой трудозатрат и установлением дедлайнов.

2.11. Тема: Проектирование архитектуры приложения и выбор технологического стека

Теория. Выбор инструментов разработки: язык программирования, среда разработки, игровой движок или фреймворк. Принципы организации кода и файлов: модульность, разделение данных и логики. Схематичное изображение связей между основными частями программы.

Практика 12-14. Выбор движка/языка из предложенного списка. Создание простой блок-схемы структуры папок. Развертывание готового шаблона проекта в среде.

Практика 15-17. Самостоятельное обоснование технологического стека. Проектирование модульной архитектуры (диаграмма компонентов). Настройка чистого шаблона с разделением логики, ассетов и конфигураций.

2.12. Тема: Планирование спринтов. Тайм-менеджмент в разработке

Теория. Поэтапная разработка (спринты): планирование коротких отрезков работы (1–2 недели). Оценка сложности задач. Балансировка нагрузки между учёбой и проектом. Личная эффективность и борьба с прокрастинацией.

Практика 12-14. Составление простого плана на 1 неделю. Распределение легких задач в группе. Согласование 1–2 проверочных точек с преподавателем.

Практика 15-17. Планирование спринта (1–2 недели) с учётом загрузки и

зависимостей задач. Балансировка нагрузки через матрицу ответственности. Установка контрольных точек, ревью и дедлайнов.

2.13. Тема: Создание прототипа основных механик (MVP). Программирование базовой логики

Теория. Понятие минимально жизнеспособного продукта (прототипа). Разработка без финальной графики (использование простых геометрических фигур). Реализация основного цикла взаимодействия пользователя с программой. Базовая физика и обработка событий.

Практика 12-14. Сборка прототипа на базовых примитивах. Реализация 1–2 основных действий (движение, клик). Тестирование идеи.

Практика 15-17. Разработка MVP без финальной графики. Программирование основного цикла взаимодействия, физики и обработки ввода. Отладка и валидация геймплейной/функциональной петли.

2.14. Тема: Интеграция контента, UI/UX и настройка физики/логики

Теория. Основы проектирования интерфейса: удобство, понятность, обратная связь пользователю. Работа с графическими элементами, звуками и текстом. Привязка визуального оформления к программному коду.

Практика 12-14. Замена примитивов на готовые ассеты. Создание базового меню и HUD (здоровье, счёт) через визуальный конструктор. Настройка простых анимаций переходов.

Практика 15-17. Интеграция оптимизированных ассетов. Вёрстка адаптивного UI/UX, привязка данных к коду. Реализация системы состояний экранов и плавных анимационных переходов.

2.15. Тема: Продвинутые механики, оптимизация производительности и сетевое взаимодействие

Теория. Анализ скорости работы программы: поиск «узких мест», нагружающих процессор или память. Способы ускорения работы: оптимизация изображений, улучшение алгоритмов, отключение неиспользуемых функций. Основы сохранения и загрузки данных пользователя.

Практика 12-14. Использование простых профайлеров/наблюдения за FPS. Замена «тяжелых» объектов на оптимизированные. Добавление простого

сохранения через файл/JSON. Запись краткого отчета «что улучшили».

Практика 15-17. Профилирование кода и ресурсов. Рефакторинг алгоритмов, пулинг объектов, оптимизация текстур/звука. Реализация системы сохранения с валидацией данных. Технический отчет с метриками «до/после».

2.16. Тема: Тестирование и отладка (QA). Работа с баг-трекером. Полировка продукта

Теория. Виды проверки качества: поиск ошибок в коде, проверка удобства использования. Правила оформления сообщения об ошибке: что делали, что случилось, как повторить. Улучшение ощущений от использования продукта: визуальные и звуковые эффекты, плавность анимаций.

Практика 12-14. Тестирование соседних проектов по чек-листу. Запись багов в простую таблицу. Исправление вылетов и явных ошибок. Добавление 2–3 визуальных/звуковых эффектов для «оживления».

Практика 15-17. Проведение кросс-тестирования с оформлением баг-репортов (шаги, ожидаемый/фактический результат). Приоритизация и фикс критических багов. Полировка: фидбек, сглаживание, микроанимации, улучшение UX.

2.17. Тема: Подготовка к релизу: экспорт, упаковка, создание презентации и документации

Теория. Процесс создания готовой версии программы для пользователя (сборка). Требования к названию, иконке и окну запуска. Структура сопроводительных документов: руководство пользователя, описание проекта для портфолио. Подготовка выступления.

Практика 12-14. Сборка проекта по инструкции для ПК или браузера. Тест на другом устройстве. Написание простой инструкции «Как запустить и играть». Подготовка 3–5 слайдов с картинками.

Практика 15-17. Настройка конфигурации билда, сборка релизной версии (ПК/Веб/Мобайл). Кросс-платформенное тестирование. Написание пользовательской документации (README/руководство). Создание структурной презентации с фокусом на технические решения.

2.18. Тема: Итоговая защита года. Презентация продукта экспертам и

партнёрам

Практика 12-14. Демонстрация проекта перед классом/наставниками. Краткое выступление (3–5 мин) по структуре «Что сделали/Что получилось». Ответы на простые вопросы. Обсуждение в группе: «Что понравилось/Что было сложно». Оформление портфолио за год.

Практика 15-17. Профессиональная защита перед экспертной комиссией. Live-demo продукта с акцентом на архитектуру, вызовы и решения. Аргументированные ответы на технические вопросы. Ретроспектива команды: анализ метрик, извлеченные уроки, план развития. Оформление портфолио за год.

1.4. Планируемые результаты

Личностные результаты

Обучающиеся будут:

- осознавать ценность профессиональной ответственности за результат командной работы, понимая взаимозависимость участников проекта и влияние качества своего кода на успех всего продукта;
- проявлять готовность к непрерывному самообразованию в условиях быстро меняющихся технологий, демонстрируя интерес к освоению новых инструментов (в том числе генеративного ИИ) и архитектурных паттернов;
- соблюдать нормы цифровой этики и авторского права, ответственно подходя к использованию сторонних ассетов, открытых библиотек и результатов работы нейросетей;
- испытывать чувство причастности к локальному IT-сообществу через участие в публичных защитах проектов, хакатонах и взаимодействие с экспертами индустрии Владивостока;
- демонстрировать культуру профессиональной коммуникации в IT-команде: формулировать конструктивную критику на код-ревью, воспринимать обратную связь как инструмент развития, а также аргументированно и уважительно отстаивать технические решения в ходе командных обсуждений;

Метапредметные результаты

Обучающиеся будут уметь:

- осуществлять системный поиск, анализ и критическую оценку информации, необходимой для решения нестандартных технических задач и выбора оптимальных архитектурных решений;

- применять методы проектного управления (Agile/Scrum) для планирования этапов разработки, декомпозиции сложных задач, управления бэклогом и соблюдения дедлайнов в условиях неопределённости;

Обучающиеся приобретут:

- навыки эффективной коммуникации и конструктивного взаимодействия в кросс-функциональной команде, включая проведение код-ревью, разрешение рабочих конфликтов и распределение ролей;

Предметные результаты

Обучающиеся будут знать:

- современные подходы к использованию искусственного интеллекта в проектной деятельности;

- принципы подготовки и проведения презентаций проектов, выступлений;

Обучающиеся будут уметь:

- формулировать цель и задачи проекта, делить их на выполняемые этапы и распределять зоны ответственности в команде;

- применять принципы итеративной разработки прототипов;

Обучающиеся будут владеть:

- навыками проектной рефлексии: анализа успехов и ошибок, внесения корректировок в рабочий процесс и планирования дальнейшего развития проекта;

РАЗДЕЛ № 2. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ

2.1. Ресурсное обеспечение программы

Материально-техническое обеспечение

Для успешной реализации программы необходимо наличие специализированного компьютерного класса, соответствующего санитарным нормам (СанПиН) и требованиям пожарной безопасности.

1. Аппаратное обеспечение:

- компьютерный класс из 10-15 автоматизированных рабочих мест (АРМ) для обучающихся и 1 АРМ для педагога;
- технические характеристики ПК должны обеспечивать комфортную работу в среде разработки Godot Engine и IDE для Python (рекомендуется: процессор не ниже Intel Core i3 / AMD Ryzen 3, оперативная память от 8 ГБ, наличие дискретной или производительной встроенной видеокарты);
- стабильное высокоскоростное подключение к сети Интернет (не менее 100 Мбит/с) для работы с облачными сервисами, системами контроля версий (Git) и API искусственного интеллекта;
- мультимедийное оборудование: проектор или интерактивная панель для демонстрации кода, презентаций и проведения разборов кейсов;
- периферийные устройства: принтер для печати методических материалов и дипломов, веб-камеры и гарнитур (для возможности дистанционного взаимодействия с представителями компаний-партнёров).

2. Программное обеспечение:

- операционная система (Windows 10/11 или Linux);

- среда визуального программирования Scratch (offline-редактор или доступ к онлайн-платформе);
- интерпретатор Python последней стабильной версии и интегрированная среда разработки (PyCharm / VS Code);
- игровой движок Godot Engine (последняя стабильная версия);
- клиент системы контроля версий (Git) и доступ к платформе хостинга кода (GitHub / GitLab / Яндекс.Код);
- антивирусное обеспечение и средства контент-фильтрации.

3. Мебель и оснащение:

- регулируемые ученические столы и стулья, соответствующие росту обучающихся;
- достаточное искусственное и естественное освещение.

Учебно-методическое и информационное обеспечение

Реализация программы обеспечивается комплексом учебно-методических документов и информационных ресурсов, адаптированных под кейс-метод обучения и региональный компонент (Владивосток).

1. Нормативно-правовые и организационные документы:

- Указ Президента Российской Федерации «Об утверждении приоритетных направлений научно-технологического развития и перечня важнейших наукоемких технологий» от 18 июля 2024 года № 529;
- Распоряжение Правительства Российской Федерации о реализации II этапа Концепции развития дополнительного образования детей до 2030 года (2025-2030 годы), от 1 июля 2025 года № 1745-р;
- Приказ Министерства образования Приморского края «Об утверждении методических рекомендаций по составлению дополнительных общеобразовательных общеразвивающих программ» от 17 февраля 2026 года № пр.23а-135;
- Федерального закона от 29 декабря 2012 года № 273-ФЗ (ред. от 30.12.2021) «Об образовании в Российской Федерации»;

- Концепции развития дополнительного образования детей, утверждённой распоряжением Правительства Российской Федерации от 31 марта 2022 года № 678-р;

- Приказа Министерства просвещения Российской Федерации от 27.07.2022 № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам» (Зарегистрирован 26.09.2022 № 70226);

- Методических рекомендаций по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы), на основании письма Министерства образования и науки РФ от 18 ноября 2015 года №09-3242;

- Санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи», утвержденных постановлением Главного государственного санитарного врача Российской Федерации от 28 сентября 2020 года № 28.

2. Учебно-методические материалы:

- **сборник кейсов «IT-Владивосток»:** комплект заданий от партнёров, содержащий техническое задание (ТЗ), критерии оценки и примеры ожидаемых результатов для каждого раздела (Scratch, Python, Godot, AI);

- методические рекомендации для педагога по организации проектной деятельности и работе с системами контроля версий в группе.

3. Информационные ресурсы:

- доступ к официальной документации языков программирования и движков (документация Python, Godot Docs);

- доступ к российским образовательным платформам (Stepik, Яндекс.Учебник, Учи.ру) для самостоятельного изучения дополнительных тем;

- электронный журнал или система управления обучением (LMS) для отслеживания прогресса обучающихся и хранения портфолио.

2.2. Оценочные материалы и формы аттестации

Система оценки качества освоения программы построена на основе

компетентностного подхода и включает в себя, текущий, промежуточный и итоговый контроль. Особенностью программы является оценка не только технических навыков, но и способности применять их для решения практических задач бизнеса (кейсов).

Формы контроля:

Текущий контроль: осуществляется на каждом занятии в форме проверки выполненных практических заданий, викторин, наблюдения за работой в среде разработки и творческих заданий.

Промежуточная аттестация: проводится по окончании каждого тематического раздела. Обучающиеся презентуют реализованный фрагмент продукта или завершённый спринт перед педагогом и учебной группой. Защита проходит в формате, моделирующем процедуру код-ревью и демо-встречу в IT-команде.

Итоговый контроль: проводится при освоении всей программы в форме публичной защиты проектов.

Формы контроля:

Устный опрос:

- знание правил распорядка в теме Техника безопасности в компьютерном классе. Организация рабочего места;
- выбор темы проекта в теме Анализ продуктов. Поиск проблемы. Выявление актуальности.

Практическая работа:

- создание библиотеки эффективных промптов для разных задач в теме Введение в генеративный ИИ. Промпт-инжиниринг для программистов;
- распределение на команды в теме Основы командной работы и инструменты для командной разработки проекта;
- настройка репозитория в теме: Распределение ролей в команде. Настройка систем контроля версий (Git) и таск-трекеров;
- поиск и анализ аналогов для своего проекта в теме Поиск аналогов. Формирование гипотез и ожиданий от продукта;

- заполнение шаблона ТЗ в теме Сбор требований и составление Технического задания. Декомпозиция задач;

- создание прототипа проекта в теме Создание прототипа основных механик (MVP). Программирование базовой логики;

- составление инструкций пользователя в теме Подготовка к релизу: экспорт, упаковка, создание презентации и документации

Проверочная работа:

- оптимизация «грязного» кода с помощью ИИ с последующим ручным аудитом в теме ИИ как помощник в кодировании: рефакторинг, поиск багов, генерация прототипов;

- поиск информации о решении задач в теме Способы поиска информации. Источники информации;

- схема архитектуры в теме Проектирование архитектуры приложения и выбор технологического стека;

- отчет об оптимизации (FPS, использование памяти) в теме Продвинутые механики, оптимизация производительности и сетевое взаимодействие

Творческая работа:

- создание пакета ассетов (спрайты/звуки) с помощью ИИ для будущего проекта в теме Генерация контента и этика использования ИИ. Авторское право и безопасность данных;

- генерация идей на различные тематики в теме Знакомство с проектной деятельностью. Способы генерации идей. Мозговой штурм;

- описание концепции проекта в теме Постановка цели и задач проекта. Разработка концепции;

- составление списка вопросов и уточнений для заказчика в теме Основы взаимодействия с заказчиком;

- составление проектной карты в теме Планирование спринтов. Тайм-менеджмент в разработке

Контрольная работа:

- защита проектной карты в теме Создание и заполнение проектной карты;

- список найденных и исправленных критических ошибок в теме;
- тестирование и отладка (QA). Работа с баг-трекером. Полировка продукта

Наблюдение:

- проверка качества интеграции ассетов и интерфейса в теме Интеграция контента, UI/UX и настройка физики/логики

Публичная защита проекта:

- демонстрация продукта, ответы на вопросы комиссии. Портфолио за год. (Проекты, скриншоты, код, сертификаты) в теме Итоговая защита года. Презентация продукта экспертам и партнерам

Критерии оценивания

Критерий	Описание требований
Функциональность	Продукт работает стабильно, без критических ошибок (багов). Реализованы все функции, заявленные в техническом задании (ТЗ)
Качество кода	Соблюдение стандартов кодирования (например, PEP8 для Python). Высокая читаемость и логичная структура. Наличие необходимых комментариев и документации
Релевантность кейсу	Решение точно отвечает на конкретный запрос компании-партнера. Учтены потребности целевой аудитории (например, удобство интерфейса для клиента банка)
Презентация	Качество устного выступления и аргументации. Умение грамотно ответить на вопросы комиссии. Наглядность и работоспособность демо-версии

Оценочные материалы:

- тестовые задания для входного контроля (логические задачи);
- чек-листы для проверки функционала программ (для каждого кейса);
- карты наблюдения за деятельностью обучающегося в группе;
- листы оценки проектов (экспертные листы) для представителей компаний-партнёров;
- портфолио обучающегося (ссылки на репозитории, скриншоты работ, сертификаты).

2.3. Методические материалы

Для реализации программы разработан комплект методических материалов, обеспечивающий единство требований и поддерживающий кейс-метод обучения. Материалы адаптированы для возрастной группы 12–17 лет и

разделены на категории для педагога и обучающегося.

1 Методы обучения: наглядный практический, объяснительно-иллюстративный, исследовательский, проектный, игровой и др

2 Методы воспитания: убеждение, поощрение, упражнение, стимулирование, мотивация и др

3 Формы организации учебного занятия: беседа, встреча с интересными людьми, выставка, защита кейсов, игра, конкурс, лабораторное занятие, лекция, мастер-класс, «мозговой штурм», эксперимент

4 Педагогические технологии: технология группового обучения, технология коллективного взаимообучения, технология развивающего обучения, технология проблемного обучения, технология исследовательской деятельности, технология проектной деятельности, технология игровой деятельности, коммуникативная технология обучения, технология коллективной творческой деятельности, технология развития критического мышления

5 Дидактические материалы: раздаточные материалы, инструкционные, технологические карты, задания, упражнения, образцы изделий и тп

№	Наименование материала	Назначение	Формат
1	Рабочая программа и календарный учебный график	Планирование учебного процесса, соблюдение сроков прохождения тем	Документ (PDF/Word)
2	Сборник кейсов «IT-Приморье»	Описание практических задач от партнёров (ДНС, ФЕСКО, Банк «Приморье» и др.). Содержит вводные данные, требования и критерии успеха	Брошюра / Электронный ресурс
3	Методические рекомендации для педагога	Сценарии занятий, советы по организации проектной работы, инструкции по работе с Git и AI-инструментами	Документ (PDF)
4	Рабочая тетрадь обучающегося	Конспекты теории, шаблоны для описания алгоритмов, листы для планирования спринтов и рефлексии	Печатная тетрадь / Онлайн-блокнот
5	Инструкционные карты по ПО	Пошаговые руководства по установке и настройке Scratch, Python, Godot, Git, IDE.	Видеоуроки / Текстовые гайды
6	Шаблоны проектной документации	Образцы Технического Задания (ТЗ), Дорожной	Файлы-шаблоны

		карты (Roadmap), Отчета о тестировании	
7	Библиотека кода и ассетов	Подборка безопасных и легальных ресурсов (спрайты, звуки, snippets кода) для использования в учебных проектах	Облачное хранилище
8	Защита проектов	Регламент проведения итоговой защиты, критерии оценки, права и обязанности участников и экспертов	Документ (PDF)

2.4. Календарный учебный график

Примечание: График составлен на основе 36 учебных недель (сентябрь–май), 4 часа в неделю (2 занятия по 2 часа), всего 144 часа в год.

Календарный учебный график

Этапы образовательного процесса		1 год
Продолжительность образовательного процесса, неделя		36
Количество учебных дней		72
Продолжительность учебных периодов	1 полугодие	15.09.2026-30.12.2026
	2 полугодие	09.01.2027-30.05.2027
Возраст детей, лет		12-17
Продолжительность занятия, час		2
Режим занятия		2 раза/нед
Годовая учебная нагрузка, час		144

2.5. Рабочая программа воспитания

Пояснительная записка

Рабочая программа воспитания является неотъемлемым компонентом дополнительной общеразвивающей программы «ИТ-Квантум: Программирование от блока кода до игрового движка». Воспитательная работа органично интегрирована в учебный процесс через проектную деятельность, решение кейсов от партнёров и командную разработку, обеспечивая единство обучения и воспитания.

Воспитательные задачи усиливают достижение предметных результатов программы, формируя личностные качества, критически важные для ИТ-

специалиста: ответственность за код, умение работать в команде и соблюдение профессиональной этики. Например, защита проектов перед партнёрами формирует не только технические навыки, но и коммуникативную культуру и ответственность за слово.

Принципы реализации

1. Личностно-ориентированный подход: учёт индивидуальных интересов обучающихся в выборе технологического стека и роли в команде.

2. Единство обучения и воспитания: формирование ценностей происходит непосредственно в процессе создания цифровых продуктов.

3. Профессиональная ориентация: погружение в реальную среду IT-индустрии через взаимодействие с компаниями-партнёрами.

3. Особенности воспитательной работы в контексте программы: специфика технической направленности обуславливает акцент на дисциплине, точности и ответственности за результат. Ключевые ценностные ориентиры: цифровая этика, безопасность данных, уважение к авторскому праву, культура командной разработки (Code Review, работа с Git).

Ожидаемая динамика развития обучающихся

Начальный этап: формирование интереса к профессии, соблюдение правил безопасности и дисциплины.

Основной этап: развитие навыков командного взаимодействия, ответственности за этап проекта, понимание этики использования ИИ.

Итоговый этап: готовность к профессиональному самоопределению, умение презентовать себя и свой продукт, рефлексия профессионального роста.

Цель рабочей программы воспитания – воспитание профессиональной ответственности, необходимой для успешного освоения инструментов разработки программного обеспечения и реализации командных проектов.

Задачи рабочей программы воспитания

1. Формировать ценностное отношение к профессии разработчика и технологическому развитию региона.

2. Развивать личностные качества, востребованные в IT-сфере

(настойчивость, внимательность, критическое мышление).

3. Осваивать нормы сотрудничества и коммуникации в процессе совместной проектной деятельности.

4. Воспитывать ответственность за качество кода, соблюдение дедлайнов и безопасность данных.

5. Стимулировать познавательную активность через решение реальных кейсов от бизнес-партнеров.

Планируемые результаты воспитания

Обучающиеся будут способны:

- демонстрировать ответственность за взятые обязательства в процессе реализации командных проектов;
- соблюдать нормы цифровой этики и авторского права при работе с программным обеспечением и данными;
- конструктивно взаимодействовать с участниками команды и экспертами в процессе защиты проектов;
- рефлексировать собственные профессиональные достижения и планировать траекторию дальнейшего развития.

Краткое пояснение календарного плана воспитательной работы

Мероприятия проводятся регулярно (1 раз в месяц) и интегрированы в учебные занятия. Логика распределения связана с этапами освоения программы: на начальном этапе акцент делается на мотивацию и знакомство с профессией, на основном этапе – на командную работу и решение кейсов, на завершающем – на рефлексии и профессиональное самоопределение. Используются формы работы: встречи с экспертами, деловые игры, проектные сессии, мастер-классы.

Технопредки в Кванториуме – День открытых дверей

Организация интерактивного пространства с демонстрацией ключевых направлений и возможностей. Проведение экскурсий и мастер-классов для привлечения новой аудитории и мотивации обучающихся к техническому творчеству.

Участие в межрегиональном конкурсе «Голдберг шоу»

Подготовка и реализация инженерных проектов в формате цепных реакций. Развитие навыков проектного мышления, командного взаимодействия и умения эффективно презентовать сложные технические решения экспертному жюри.

Краевой чемпионат по игровым дисциплинам «День отца»

Проведение семейного турнира, объединяющего поколения через совместную техническую деятельность. Формирование уважения к семейным ценностям и развитие коммуникативных навыков через публичное представление совместных продуктов.

Цифровой подарок маме в рамках проекта «Технопредки»

Серия практических занятий по созданию персонализированных цифровых продуктов. Воспитание заботы о близких через применение технических навыков для решения творческих задач и изготовления индивидуальных подарков.

Новогодний хакатон: Чудеса кода

Интенсивная командная работа по разработке тематических программных продуктов в сжатые сроки. Отработка навыков распределения ролей, соблюдения дедлайнов и коллективной ответственности за конечный результат проекта.

Участие в VI Всероссийском турнире «РОБО БИТВА 2026»

Интенсивная подготовка робототехнических комплексов и отработка алгоритмов управления. Развитие стрессоустойчивости, алгоритмического мышления и навыков конкурентной борьбы в условиях регламентированных соревнований.

День Российской науки: Код будущего

Проведение научно-популярных экспериментов и лекций, связывающих фундаментальную науку с прикладными технологиями. Стимулирование исследовательского интереса и осознание роли научных открытий в современном технологическом прогрессе.

Участие в конкурсе «Инженеры FESCO» (отборочный этап)

Разработка инженерных решений с акцентом на практико-ориентированный подход и промышленный дизайн. Освоение принципов прототипирования, технико-экономического обоснования идей и навыков профессиональной презентации проектов.

Фестиваль «Код: космос» ко Дню космонавтики

Тематический интенсив, интегрирующий изучение алгоритмов с моделированием задач космической отрасли. Демонстрация связи образовательных компетенций с реальными инженерными вызовами и историей освоения космоса.

День открытых дверей «Лето в Кванториуме»

Организация выставки лучших годовых проектов с привлечением опытных обучающихся в роли наставников. Развитие лидерских качеств, навыков менторства и способности структурированно передавать знания менее опытным участникам.

2.6. Календарный план воспитательной работы

Дата / период	Название мероприятия	Форма проведения	Планируемый воспитательный результат
Сентябрь	«Технопредки в Кванториуме» — День открытых дверей	Интерактивная экскурсия с демонстрацией программирования	Обучающиеся проявят интерес к техническому творчеству и осознают возможности развития в сфере программирования
Октябрь	Участие в межрегиональном конкурсе «Голдберг шоу»	Проектная сессия / инженерное моделирование	Обучающиеся разовьют навыки проектного мышления и умение презентовать техническое решение
Октябрь	Краевой чемпионат по игровым дисциплинам «День отца»	Турнир / семейный фестиваль	Обучающиеся проявят уважение к семейным ценностям и научатся представлять свой продукт аудитории
Ноябрь	«Цифровой подарок маме» в рамках проекта «Технопредки»	Практикум / мастер-класс	Обучающиеся проявят заботу о близких через создание персонализированного цифрового продукта
Декабрь	«Новогодний хакатон: Чудеса кода» (к Новому году)	Командная проектная сессия (создание тематической игры/анимации)	Обучающиеся будут способны работать в команде над общим продуктом, соблюдая дедлайны и распределяя роли в условиях ограниченного

			времени
Январь	Участие в VI Всероссийском турнире «РОБО БИТВА 2026»	Подготовка и участие в конкурсе	Обучающиеся разовьют навыки алгоритмического мышления и работы в соревновательной среде
Февраль	«День Российской науки: Код будущего»	Научно-популярная лаборатория	Обучающиеся осознают роль науки в развитии технологий и проявят интерес к исследовательской деятельности
Март	Участие в конкурсе «Инженеры FESCO» (отборочный этап)	Проектная работа / инженерный дизайн	Обучающиеся освоят принципы инженерного подхода и научатся презентовать технические идеи
Апрель	Фестиваль «Код: космос» ко Дню космонавтики	Тематический интенсив	Обучающиеся свяжут изучаемые алгоритмы с реальными инженерными задачами космической отрасли
Май	День открытых дверей «Лето в Кванториуме»	Интерактивная выставка проектов	Обучающиеся научатся менторству и передаче знаний младшим участникам

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Godot Engine. Официальная документация [Электронный ресурс]. – Режим доступа: <https://docs.godotengine.org/ru/stable/> (дата обращения: 01.03.2026).
2. GitHub Docs. Документация для начинающих [Электронный ресурс]. – Режим доступа: <https://docs.github.com/ru/get-started> (дата обращения: 01.03.2026).
3. Python Software Foundation. Документация Python 3 на русском языке [Электронный ресурс]. – Режим доступа: <https://docs.python.org/3/> (дата обращения: 01.03.2026).
4. Google AI Education. Основы искусственного интеллекта [Электронный ресурс]. – Режим доступа: <https://ai.google/education> (дата обращения: 01.03.2026).