

Общество с ограниченной ответственностью
«Новое инженерное образование»



Директор МПБО «Гимназия №107
«Открытие»

А.С. Акмаева

2024 г.



Управляющий

А.А. Свирина

2024 г.

ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА
ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ

Мастер программирования

(1-11 класс)

Разработчик программы:
преподаватель Пищулин М.С.

Казань 2024 г.

СОДЕРЖАНИЕ

№	Наименование разделов	Стр.
1	Общие положения	3
2	Характеристика результатов обучения по образовательной программе	5
3	Компетенции выпускника образовательной программы	6
4	Документы, регламентирующие содержание и организацию образовательного процесса при реализации образовательной программы	7
5	Ресурсное обеспечение образовательной программы	13
6	Нормативно-методическое обеспечение системы оценки качества освоения образовательной программы	14

1. Общие положения

Образовательная программа определяет требования по реализации образовательной деятельности по профилю подготовки «Мастер программирования».

Образовательная программа является программой дополнительного образования.

Развитие цифровых профессий и их доминирование на рынке труда, а также востребованность навыков программирования в сферах, не связанных напрямую с разработкой программных продуктов, диктует необходимость формирования программ дополнительного образования, направленных на создание устойчивых междисциплинарных компетенций по наиболее востребованным направлениям профессиональной деятельности в диджитал-сфере. Наиболее универсальной профессией в данной области является разработчик программного обеспечения, владеющий востребованными языками программирования. Универсальный программист является одной из самых востребованных профессий на рынке труда, а потребность в соответствующих специалистах присутствует как в реальном, так и в финансовом секторе экономики. В соответствии с вышеизложенным, реализация образовательной программы по профилю «Мастер программирования» является обоснованной.

Образовательная программа дополнительного образования (далее – ОП ДО) представляет собой систему документов, разработанную и утвержденную образовательной организацией с учетом потребностей рынка труда, требований органов исполнительной власти и соответствующих отраслевых требований, а также с учетом структуры федеральных государственных образовательных стандартов начального общего, основного общего, среднего общего образования.

ОП ДО регламентирует цели, ожидаемые результаты, содержание, условия и технологии реализации образовательного процесса, оценку освоения образовательной программы и включает в себя: учебно-тематический план, аннотации программ дисциплин (модулей) и другие материалы, обеспечивающие реализацию соответствующей образовательной технологии.

Реализация образовательной деятельности осуществляется на основании требований следующих основных документов:

- Федеральный закон от 29.12.2012 №273-ФЗ «Об образовании в Российской Федерации»;
- Федеральный государственный образовательный стандарт начального общего образования.
- Федеральный государственный образовательный стандарт основного общего образования.
- Федеральный государственный образовательный стандарт среднего общего образования.
- Устав ООО «Новое инженерное образование»;
- Распорядительные акты ООО «Новое инженерное образование».

Форма обучения: **очная, дистанционная**

Срок освоения: **11 лет** (возможно частичное освоение)

Требования к обучающемуся:

Обучающийся должен обучаться по программе начального/основного/среднего общего образования; владеть государственным языком общения.

Подробная информация об условиях приема на образовательную программу определяется распорядительными актами ООО «Новое инженерное образование».

Основной целью подготовки по образовательной программе является:

- обеспечение комплексной и качественной подготовки квалифицированных, конкурентоспособных выпускников в области программирования и разработки программного обеспечения на основе сочетания современных образовательных технологий и методик для формирования профессиональных и личностных качеств, развития творческого потенциала обучающихся, который будет удовлетворять требованиям рынка труда и позволит добиться успеха в профессиональной и дальнейшей образовательной деятельности.

Целями образовательной программы являются:

- формирование личностных результатов обучения на основе гуманитарных, социальных, правовых, экономических, математических и естественнонаучных знаний, позволяющих ему успешно работать в сфере разработки программного обеспечения и быть конкурентоспособным на рынке труда;

- формирование метапредметных результатов обучения на основе интеграции знаний, умений и навыков из различных сфер, соответствующих профилю профессии разработчика программного обеспечения и индивидуальной образовательной траектории обучающегося;

- формирование предметных результатов обучения на основе интеграции знаний, умений и навыков из различных сфер, соответствующих профилю разработчика программного обеспечения (математическая подготовка, подготовка в области естественных наук, интернет вещей, подготовка в области создания и обработки программного обеспечения с использованием наиболее распространённых языков программирования, программирование игры и веб-программирование) и применимых в профильной деятельности.

Образовательная программа имеет сформулированные задачи (ожидаемые результаты обучения), согласованные с целями образовательной программы:

- формирование личностных качеств;
- формирование личностных компетенций;
- формирование метапредметных компетенций;
- формирование предметных компетенций;
- подготовка к будущей профессиональной деятельности;
- формирование знаний и умений в объеме, достаточном для продолжения обучения по программам профессионального образования.

Формирование результатов обучения происходит по годам освоения образовательной программы.

2. Характеристика результатов обучения по образовательной программе

Ожидаемые результаты обучения по блокам освоения образовательной программы представлены в таблице ниже.

	ПОНИМАНИЕ	РАЗРАБОТКА	ВНЕДРЕНИЕ	ПРИМЕНЕНИЕ	СОЗДАНИЕ
1-2 класс (Начинающий уровень)	Понимает основы программирования Майнкрафт, Роблокс	Может применять основы программирования в игровой среде ПиктоМир, AppInventor, thinkable	-	-	-
3-4 класс (Базовый уровень)	Понимает основы трехмерного моделирования	Может программировать в среде Scratch	Может создавать 3D-модели в TinkerCad	-	-
5-6 класс (Неуверенный пользователь)	Понимает основы языка Python Понимает механизм работы оконных приложений	Может программировать в Python Умеет использовать оконные приложения C#.NET	Может выбрать оптимальный инструмент к решению задачи программирования	-	-
7-8 класс (Пользователь среднего уровня)	Понимает основы языков Python, Java, JS, C++	Может генерировать программный код в Python, Java, JS, C++ на основе шаблонов	Может выбрать оптимальный для условий задачи язык программирования	Может решить стандартную задачу программирования на Python	-
9 класс (Уверенный пользователь)	-	Может генерировать программный код в Python, Java, JS, C++ самостоятельно	Может выбрать оптимальный для условий задачи язык программирования	Может решить нестандартную задачу программирования на Java, JS, C++	Находит клиента для применения своих навыков
10 класс (Высокий уровень)	Понимает основы языка Unity, Web-программирования	Может генерировать программный код на Unity	Может самостоятельно создавать игры и веб-сайты	Может решить нестандартную задачу программирования на Unity	Может подготовить техническое задание проекта по ИТ-разработке
11 класс (Профессиональный уровень)	-	Умеет программировать при отсутствии шаблонных решений	Умеет подбирать оптимальный инструментарий программного решения	Может реализовать проект по разработке ИТ-решения	Может организовать исполнение проекта по разработке ИТ-решения

3. Компетенции выпускника образовательной программы

В результате освоения образовательной программы выпускник соответствии с целями и задачами образовательной программы дополнительного образования, а также профилем подготовки данной образовательной программы должен обладать следующими **компетенциями**:

№	Описание и тип компетенции
Личностные	
1.	способность использовать основы научного мировоззрения для решения профессиональных задач различного уровня сложности
2	способность к коммуникации в устной и письменной формах на русском и иностранном языках для решения задач межличностного взаимодействия
3	способность работать в коллективе, толерантно воспринимая социальные, этнические, конфессиональные и культурные различия
4.	способность к самоорганизации и самообразованию
Метапредметные	
5.	умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи разработки программного обеспечения
6.	умение самостоятельно планировать пути достижения целей, в том числе альтернативные, осознанно выбирать наиболее эффективные способы решения задач разработки программного обеспечения
7.	умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности в процессе достижения результата, определять способы действий в рамках предложенных условий и требований, корректировать свои действия в соответствии с изменяющейся ситуацией в разработке программного обеспечения
8.	умение определять понятия, создавать обобщения, устанавливать аналогии, классифицировать, самостоятельно выбирать основания и критерии для классификации, устанавливать причинно-следственные связи, строить логическое рассуждение, умозаключение (индуктивное, дедуктивное и по аналогии) и делать выводы при разработке программного обеспечения
9.	умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками; работать индивидуально и в группе: находить общее решение и разрешать конфликты на основе согласования позиций и учета интересов; формулировать, аргументировать и отстаивать свое мнение в инженерной сфере
10.	умение осознанно использовать речевые средства в соответствии с задачей коммуникации для планирования и регуляции своей деятельности в разработке программного обеспечения
11.	формирование и развитие компетентности в области использования информационно-коммуникационных технологий (далее ИКТ- компетенции); развитие мотивации к овладению культурой активного пользования словарями и другими поисковыми системами для решения задач разработки программного обеспечения
Предметные	
12.	умение проектировать и создавать программное обеспечение, в том числе в игровом пространстве
13.	умение программировать на языках Python, Java, Javascript, C++, Unity
14.	умение создавать программное обеспечение в специфических средах, включая интернет вещей, игровые миры, веб-ресурсы

15.	умение применять навыки программирования для решения стандартных и нестандартных задач, в том числе применяя low-code и no-code решения
-----	---

4. Документы, регламентирующие содержание и организацию образовательного процесса при реализации образовательной программы

Сводный график освоения образовательной программы (в академических часах, из расчета 2 занятия в неделю по 2 академических часа) представлен в таблице ниже.

Класс	1 четверть	2 четверть	3 четверть	4 четверть	Итого часов
1	36	30	46	32	144
2	36	30	46	32	144
3	36	30	46	32	144
4	36	30	46	32	144
5	36	30	46	32	144
6	36	30	46	32	144
7	36	30	46	32	144
8	36	30	46	32	144
9	36	30	46	32	144
10	36	30	46	32	144
11	36	30	46	32	144

Допускается перенос часов между четвертями при сохранении общего объема часов освоения образовательной программы.

Состав учебной группы – до 18 человек.

Расписание занятий формируется для учебной группы и утверждается генеральным директором ООО «Новое инженерное образование».

Целью изучения дисциплин образовательной программы является формирование у обучающихся ключевых компетенций, необходимых для успешного решения профессиональных задач в области разработки программного обеспечения на основе применения широкого спектра естественно-научных, технических и программных решений.

Основными задачами изучения профиля являются:

- формирование технического мышления на основе понимания законов естественно-научного развития;
- освоение инструментов блочного программирования, включая специализированные игровые инструменты;
- освоение инструментов разработки программного обеспечения, включая специализированные инструменты;
- освоение основных языков программирования, применяемых при решении инженерных задач (Python, Java, Javascript, C++, Unity и т.п.);

- формирования навыков создания программного обеспечения, соответствующего целям и задачам, сформулированным заказчиком.

Учебно-тематический план представлен в таблице ниже

№	Наименование раздела	Год обучения	Количество часов	Итоговая аттестация	
Основы профессии разработчика программного обеспечения					
1	Основы компьютерной грамотности. Применение основ программирования в игровой среде Minecraft	1	72	Защита проекта	IT-
2	Основы компьютерной грамотности. Применение основ программирования в игровой среде Роблокс	1	72		
3	Применение основ программирования в игровой среде ПиктоМир	2	72	Защита проекта	IT-
4	Применение основ программирования в игровой среде AppInventor/thunkable	2	72		
5	Программирование в Scratch (базовый уровень)	3	72	Защита проекта	IT-
6	Компьютерное трехмерное моделирование TinkerCad (базовый уровень)	3	72		
7	Программирование в Scratch (продвинутый уровень)	4	72	Защита проекта	IT-
8	Компьютерное трехмерное моделирование TinkerCad (продвинутый уровень)	4	72		
Освоение основных инструментов профессии					
9	Применение объектно-ориентированных языков программирования: введение в Python (базовый уровень)	5	72	Защита проекта	IT-
10	Оконные приложения C#.NET (базовый уровень)	5	72		
11	Применение объектно-ориентированных языков программирования: введение в Python (средний уровень)	6	72	Защита проекта	IT-
12	Оконные приложения C#.NET (средний уровень)	6	72		
13	Программирование на Python (решение типовых задач)	7	36	Защита проекта	IT-
14	Программирование на Java (решение типовых задач)	7	36		
15	Программирование на Javascript (решение типовых задач)	7	36		
16	Программирование на C++ (решение типовых задач)	7	36		
17	Программирование на Python (решение нетиповых задач)	8	36	Защита проекта	IT-
18	Программирование на Java (решение	8	36		

	нетиповых задач)			
19	Программирование на Javascript (решение нетиповых задач)	8	36	
20	Программирование на C++ (решение нетиповых задач)	8	36	
21	Программирование игр на Unity	9	72	
22	Веб-программирование	9	72	Защита ИТ-проекта
Развитие в профессии				
23	Технологии искусственного интеллекта	10	72	Защита ИТ-проекта
24	Создание простых прикладных ИТ-решений	10	72	
25	Оптимизация технологии создания ИТ-решения	11	72	Защита ИТ-проекта
26	Создание сложных прикладных ИТ-решений	11	72	
Итого часов			1584	

Содержание разделов.

1. Основы компьютерной грамотности. Применение основ программирования в игровой среде Minecraft.

Инструктаж по правилам безопасности. Основные понятия. История компьютерной техники. Разновидности компьютеров и их компоненты. Азбука интернета. Начало программирования. Знакомство с программированием внутри Minecraft. Создание программы для постройки каких-либо объектов в Minecraft. Изучение циклов и использование их в постройке с помощью черепашки. Основы программирования на черепашке. Изучение переменных и циклов.

2. Основы компьютерной грамотности. Применение основ программирования в игровой среде Роблокс.

Программное обеспечение. Типы файлов. Структура файловых пространств. Знакомство с Roblox Studio. Создание Obby. Изучение Terrain Editor. Скрипты и режимы игры.

3. Применение основ программирования в игровой среде ПиктоМир

Знакомство со средой разработки ПиктоМир. Интерфейс программирования. Понятие «робот», виды роботов, их назначение. Понятия «команда», «программа», «командная строка».

Упражнения на построение алгоритмов с использованием дидактического материала. Порядок выполнения команд в простейших программах. Особенности и варианты записи линейной программы. Построение линейной программы с использованием команд учебной программной среды ПиктоМир. Упражнения на построение линейных программ с использованием пиктограммного лото. Упражнения на нахождение и исправление ошибок в записанных линейных программах. Особенности и варианты записи цикла. Построение программы, со-

держщей цикл, с использованием команд учебной программной среды ПиктоМир. Использование нескольких подпрограмм в одной программе.

4. Применение основ программирования в игровой среде AppInventor/thunkable.

Знакомство со средой разработки AppInventor/Thunkable. Изучение интерфейса. Работа по установке приложений на мобильные устройства. Изучение математических операторов, переменных. Работа с текстовым и цветовым блоками. Массивы. Функции и процедуры. Медиа блок. Рисование и анимация.

5. Программирование в Scratch (базовый уровень)

Демонстрация примеров проектов, сделанных в среде Scratch. Знакомство со средой программирования Scratch. Установка Scratch на домашнем компьютере. Интерфейс и главное меню Scratch. Понятия «скрипт», «сцена», «спрайт». Система команд исполнителя Scratch. Блоки и команды. Движение, звук, цвет спрайтов. Управление и контроль над спрайтом, анимация. Понятие проекта, его структура и реализация в среде Scratch. Этапы разработки и выполнения проекта (постановка задачи, составление сценария, программирование, тестирование, отладка) с помощью Scratch. Дизайн проекта. Примеры поэтапной разработки проекта. Создание и защита проекта, созданного в среде программирования Scratch. Графика в Scratch.

6. Компьютерное трехмерное моделирование Tinkercad (базовый уровень)

Знакомство с интерфейсом Tinkercad. Терминология. Функциональные возможности программной среды. Примеры использования 3D печати. Редактор для моделирования Tinkercad.

7. Программирование в Scratch (продвинутый уровень).

Проект в Scratch. Изучение и реализация проектов «Игра с геометрическими фигурами», «Игра с буквами», «Игра со случайными надписями», «Сказка», «Квест». Разработка собственного проекта, его программирование, дизайн, оформление и защита. Публикация собственного проекта на сайте <http://scratch.mit.edu>. Скачивание и использование чужих проектов, доступных пользователям данного сайта, авторские права.

8. Компьютерное трехмерное моделирование Tinkercad (продвинутый уровень)

Редактор для моделирования Tinkercad. Базовые функции создания эскизов. Булевы операции. Создание эскизов с параметризацией. Функции готовых простых фигур. Создание сборки из нескольких простых деталей. Привязка.

9. Применение объектно-ориентированных языков программирования: введение в Python (базовый уровень)

Редактор кода Visual Studio Code. Переменные. Типы данных. Способы работы с ними и их преобразование. Операторы сложения, вычитания, умножения, деления, взятия остатка от деления и целочисленное деление. Оператор if. Конструкция if...else. Конструкция

if...elif...else. Вложенные условия. Операторы сравнения. Логический тип bool. Особенности проверки истинности в логических выражениях. Введение в алгебру логики. Операторы AND, OR, NOT.

10. Оконные приложения C#.NET (базовый уровень)

Знакомство с элементами окон. Создание окна. Рассмотрение основных свойств. Динамическое добавление элементов, элементы GroupBox, Panel и FlowLayoutPanel, размеры элементов и их позиционирование в контейнере. Создание главного меню. Создание MDI-приложений. Перечисление MdiLayout. Вырезание, копирование и вставка текстовых фрагментов. Контекстное меню. Диалоговые окна. Панель инструментов ToolStrip, MenuStrip.

11. Применение объектно-ориентированных языков программирования: введение в Python (продвинутый уровень)

Упорядоченные и неупорядоченные последовательности. Их виды в Python. Общие особенности и методы. Сравнение последовательностей. Слияние. Срезы. Цикл while. Цикл for. Тип Range: диапазон чисел. Синтаксис str. Методы строк. Работа с регистром символов. Поиск и замена. f-строки. Синтаксис list. Распаковка списков. Методы списков. Синтаксис кортежей (неизменяемых списков). Преимущества кортежей. Упаковка и распаковка кортежей. Сортировка кортежей. Удаление элемента. Тип dict. Объявление словаря. Функция zip(). Методы словарей. Метод get(). Добавление нового элемента в словарь. Объединение словарей. Удаление элемента из словаря. Получения ключа и значения из словаря. Копирование словарей.

12. Оконные приложения C#.NET (продвинутый уровень).

Асинхронное программирование в .NET Framework. Методы EndOperation, Pooling, Callback. Асинхронный запуск произвольного метода. Обновление интерфейса. Реализация линий, графических примитивов в Windows Forms. Анимация окон и графических примитивов. Проектная деятельность.

13. Программирование на Python (решение типовых задач)

Создание новых классов в Python. Инициализация полей класса. Как «напечатать» класс. Магические методы __init__ и __str__. Наследование. Понятия «родительский класс», «дочерний класс» и «интерфейс класса». Полиморфизм. Type Hints: аннотация типов данных. Синтаксис аннотирования встроенных типов данных. Type Hints для функций. Type Hints для классов. Проверка аннотации типов через консоль.

14. Программирование на Java (решение типовых задач)

История и эволюция языка Java. JDK и JVM. Введение в среду разработки приложений IntelliJ Idea. Типы данных Java. Методы. Массивы одномерные, многомерные. Строки. Методы работы со строками. Оболочечные классы. Класс Math, его методы и константы. Опера-

торы циклов. Операторы ветвления. Операторы выбора. Операторы прерывания, перехода и возврата. Объектная модель Java.

15. Программирование на Javascript (решение типовых задач)

Структура кода. Переменные. Типы данных. Взаимодействие. Преобразование типов. Базовые операторы языка. Операторы сравнения. Логические операторы. Функции. Отличительные особенности Javascript.

16. Программирование на C++ (решение типовых задач)

Знакомство с языком C++. Инструкции управления. Система ввода-вывода. Массивы и строки. Указатели. Функции. Структуры и объединения

17. Программирование на Python (решение нетиповых задач)

API. Форматы обмена данными. JSON. XML. Взаимодействие программ по сети: протокол SOAP. Архитектура REST: ресурсы, эндпоинты и HTTP-методы. HTTP. URL. Кодирование URL. HTTP-ответы. Заголовки ответов. HTTP-запросы. Python вместо браузера. Передача параметров в URL. Заголовки запросов и ответов. Обработка ошибок. Client API для Телеграм. Bot API для Телеграм. Обработка входящих сообщений. Отправка сообщений из кода. Объект update. Реализация графических интерфейсов. Отправка медиа. Хранение токенов. Обработка исключений. Журнал ошибок.

18. Программирование на Java (решение нетиповых задач)

Поточная модель организации I/O в Java. Классы InputStream и OutputStream. Классификация исключений. Объявление контролируемых исключений. Генерация исключений. Создание классов исключений. Перехват исключений. Регистрация ошибок. Логирование. Интерфейсы коллекций в Java – Set, List, Queue, Map. Реализации данных интерфейсов в классах.

19. Программирование на Javascript (решение нетиповых задач)

Прототипы, наследование. Классы. Обработка ошибок. Промисы. Генераторы. Продвинутая итерация. Модули. Юникод.

20. Программирование на C++ (решение нетиповых задач)

Понятие перегрузки. Использование перегрузки. Понятие принципа наследования. Применение в ООП. Понятие и реализация шаблонов функций. Понятие структур данных и применение. Сортировка данных.

21. Программирование игр на Unity.

Введение в разработку игр и их проектирование. Начало работы с Unity. Интерфейс и взаимодействие объектов на сцене. Изучение основ и базы C# для создания игр. Материалы и анимация игровых объектов. Основы физики и префабы. Импорт 3D-моделей из редакторов, добавление музыки. Работа с эффектами. Формирование и создание главного меню игры.

22. Веб-программирование

Введение в HTML. Применение CSS. Применение Javascript в веб-программировании. Применение PHP в веб-программировании. Адаптация веб-сайтов к мобильным устройствам.

23. Технологии искусственного интеллекта.

Введение в машинное обучение, основные понятия, области применения. Основные понятия нейронных сетей, области применения. Виды нейронных сетей, сверточные сети. Компьютерное зрение.

24. Создание простых прикладных ИТ- решений.

Решение задач разработки программного обеспечения по кейсам работодателей. Разработка элементов кода (уровень джун).

25. Оптимизация технологии создания ИТ-решения

Выбор оптимального подхода к созданию ИТ-решения. Критерии оптимизации. Выбор оптимального языка программирования при создании ИТ-решения. Техническое задание ИТ-проекта.

26. Создание сложных прикладных ИТ- решений.

Решение задач разработки программного обеспечения по кейсам работодателей. Разработка элементов кода (уровень мидл).

5. Ресурсное обеспечение образовательной программы

Материально-техническое обеспечение изучения профиля (предоставляется по месту оказания услуг в рамках договора аренды)

1. Учебные помещения
2. Специализированное оборудование в учебных помещениях
3. Учебная литература

Рекомендуемая литература.

1. Иван Жуков. Компьютер! Большой понятный самоучитель. Все подробно и «по полочкам», Стандарт, 2017.
2. Дэвид Вэйл, Мартин О'Хэнлон. Minecraft. Програмируй свой мир. – СПб.: Питер, 2016. – 224 с.
и объектов реального мира средствами конструктора LEGO). — М.:ЛИНКА-ПРЕСС, 2001
3. Голиков Денис, Голиков Артем Книга юных программистов на Scratch 2013 г.
4. Тонни Гэддис Начинаем программировать на Python. - 4-е изд. - СПб.: BHV, 2019. - 768 с
5. Эрик Мэтиз. Изучаем Python. Программирование игр, визуализация данных, веб-приложения. - СПб.: Питер, 2020. - 512 с.
6. Биллиг В.А. Основы объектного программирования на C# (C# 3.0, Visual Studio 2008)
[Электронный ресурс] : учебное пособие / В.А. Биллиг. — Электрон. текстовые данные. —

Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Вузовское образование, 2017. — 583 с.

7. Блох, Джошуа. Java: эффективное программирование, 3-е изд.: Пер. с англ. — СПб.: ООО «Диалектика», 2019. — 464 с.

8. Павловская, Т.А. C/ C++. Программирование на языке высокого уровня / Т.А. Павловская. — СПб.: Питер, 2011. — 461 с.

9. Джозеф Хокинг. Unity в действии. Мультиплатформенная разработка на C#. 2-е межд. изд. — СПб.: Питер, 2019. — 352 с.: ил. — (Серия «Для профессионалов»). ISBN 978-5-4461-0816-9

Интернет ресурсы:

1. <https://ru-minecraft.ru/instrukcii-gajdy-minecraft/34020-computercraft-tutorial1.html>
2. <https://ru-minecraft.ru/instrukcii-gajdy-minecraft/34045-computercraft-tutorial2.html>
3. <https://ru-minecraft.ru/instrukcii-gajdy-minecraft/34045-computercraft-tutorial3.html>
4. <http://scratch.mit.edu>.
5. <https://visualstudio.microsoft.com/ru/>
6. www.python.org

Кадровое обеспечение профиля

Кадровое обеспечение профиля формируется руководителем образовательной программы из числа лиц, которые имеют высшее профессиональное образование, опыт педагогической деятельности не менее года, могут иметь учению степень и/или ученое звание.

Материалы образовательной программы при освоении программы с применением дистанционных образовательных технологий, размещаются в личных кабинетах пользователей на веб-ресурсе <https://niogroup.nethouse.ru/>

6. Нормативно-методическое обеспечение системы оценки качества освоения образовательной программы

Освоение образовательной программы, в том числе отдельной части или всего объема дисциплины (модуля), сопровождается текущим контролем успеваемости и итоговой аттестацией обучающихся.

Текущий контроль успеваемости обеспечивает последовательность формирования компетенций.

Освоение модулей представленной образовательной программы завершается итоговой аттестацией в форме защиты проектной работы. По итогам освоения ступени обучающийся вправе пройти независимый экзамен на сформированность компетенции.

Итоговая аттестация

Итоговая аттестация проходит в форме защиты проекта по окончании каждого года обучения. Оценивание происходит в формате «зачтено / не зачтено» в соответствии со следующими критериями:

Словесное выражение	Описание
Зачтено	Обучающийся выполнил весь объем проектной работы и ответил на вопросы, возникшие на защите проекта, полностью или частично
Не зачтено	Обучающийся не выполнил объем проектной работы и/или не ответил ни на один из вопросов, возникших на защите проекта

По 6 уровням освоения профиля слушатели вправе пройти независимый экзамен на уровень владения профессией по следующей классификации:

Буквенное выражение	Описание
A1	Начинающий (владение базовым набором инструментов)
A2	Элементарный (применение базового набора инструментов в усложненных задачах)
B1	Пороговый уровень (готовность самостоятельно работать в профессии)
B2	Продвинутый пороговый уровень (готовность самостоятельно работать в профессии, разрабатывая собственные решения)
C1	Профессиональное владение (готовность к полноценному трудоустройству на начальном уровне)
C2	Полное владение (готовность к решению нетиповых профессиональных задач)