

ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ "ЦИФРА НИЖНИЙ
НОВГОРОД"

Утверждаю



**Дополнительная профессиональная
программа - переподготовки программа
«Разработка программного обеспечения»**

Нижний Новгород
2024г.

I. Общие положения

1. Дополнительная профессиональная программа (программа профессиональной переподготовки) ИТ-профиля «**Разработка программного обеспечения**» (далее – Программа) разработана в соответствии с нормами Федерального закона РФ от 29 декабря 2012 года № 273-ФЗ «Об образовании в Российской Федерации», с учетом требований приказа Минобрнауки России от 1 июля 2013 г. № 499 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам», с изменениями, внесенными приказом Минобрнауки России от 15 ноября 2013 г. № 1244 «О внесении изменений в Порядок организации и осуществления образовательной деятельности по дополнительным профессиональным программам, утвержденный приказом Министерства образования и науки Российской Федерации от 1 июля 2013 г. № 499», *приказа Министерства образования и науки РФ от 23 августа 2017 г. N 816 «Об утверждении Порядка применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ» (указать при необходимости);* паспорта федерального проекта «Развитие кадрового потенциала ИТ-отрасли» национальной программы «Цифровая экономика Российской Федерации»; постановления Правительства Российской Федерации от 13 мая 2021 г. № 729 «О мерах по реализации программы стратегического лидерства «Приоритет-2030» (в редакции постановления Правительства Российской Федерации от 14 марта 2022 г. № 357 «О внесении изменений в постановление Правительства Российской Федерации от 13 мая 2021 г. № 729»)

2. Профессиональная переподготовка заинтересованных лиц (далее – Слушатели), осуществляемая в соответствии с Программой (далее – Подготовка), имеющей отраслевую направленность **информационно-**

коммуникационные технологии, в соответствии с учебным планом в очной/очно-заочной форме обучения.

3. Разделы, включенные в учебный план Программы, используются для последующей разработки календарного учебного графика, учебно-тематического плана, рабочей программы, оценочных и методических материалов.

4. Программа регламентирует требования к профессиональной переподготовке в области создания и применения технологий больших данных.

Срок освоения Программы составляет 1152 академических часа.

К освоению Программы в рамках проекта допускаются лица:

- получающие высшее образование по очной (очно-заочной) форме, лица, освоившие основную профессиональную образовательную программу (далее – ОПОП ВО) бакалавриата – в объеме не менее первого курса (бакалавры 2-го курса), ОПОП ВО специалитета – не менее первого и второго курсов (специалисты 3-го курса). Также к освоению ДПП III допускаются лица, обучающиеся по программам магистратуры, которые не относятся к ИТ-профилю (согласно приложению к Методике расчета показателя граждан, прошедших обучение по дополнительным образовательным программам) и по программам ординатуры.

II. Цель

1. Целью подготовки слушателей по Программе является получение

компетенции, необходимой для выполнения нового вида профессиональной деятельности в области информационных технологий для обучающихся по специальностям и направлениям подготовки, не отнесенным к ИТ-сфере; приобретение новой квалификации «**Разработчик**».

2. Перечень профессиональных компетенций в рамках имеющейся квалификации, качественное изменение которых осуществляется в результате обучения.

Выпускник, освоивший программу профессиональной переподготовки, должен обладать следующими компетенциями, на которые ориентирована программа:

ПМ 01. Проектирование и разработка информационных систем

Уметь:

- осуществлять постановку задач по обработке информации; проводить анализ предметной области;
- осуществлять выбор модели и средства построения информационной системы и программных средств;
- использовать алгоритмы обработки информации для различных приложений;
- решать прикладные вопросы программирования и языка сценариев для создания программ;
- разрабатывать графический интерфейс приложения;
- создавать и управлять проектом по разработке приложения;
- проектировать и разрабатывать систему по заданным требованиям и спецификациям.

Знать:

- основные виды и процедуры обработки информации, модели и методы решения задач обработки информации;
- основные платформы для создания, исполнения и управления информационной системой;
- основные процессы управления проектом разработки;
- основные модели построения информационных систем, их структуру,

особенности и области применения;

- методы и средства проектирования, разработки и тестирования информационных систем;
- систему стандартизации, сертификации и систему обеспечения качества продукции.

ПМ 02. Разработка дизайна веб-приложений

Уметь:

- создавать, использовать и оптимизировать изображения для веб-приложений;
- выбирать наиболее подходящее для целевого рынка дизайнерское решение;
- создавать дизайн с применением промежуточных эскизов, требований к эргономике и технической эстетике;
- разрабатывать интерфейс пользователя для веб-приложений с использованием современных стандартов

Знать:

- нормы и правила выбора стилистических решений;
- современные методики разработки графического интерфейса;
- требования и нормы подготовки и использования изображений в информационно телекоммуникационной сети "Интернет" (далее - сеть Интернет);
- государственные стандарты и требования к разработке дизайна веб-приложений

ПМ 03. Проектирование, разработка и оптимизация веб-приложений

Уметь:

- разрабатывать программный код клиентской и серверной части веб-приложений;
- осуществлять оптимизацию веб-приложения с целью повышения его рейтинга в сети Интернет;
- разрабатывать и проектировать информационные системы.

Знать:

- языки программирования и разметки для разработки клиентской и серверной

части веб приложений;

- принципы функционирования поисковых сервисов и особенности оптимизации веб приложений под них;

- принципы проектирования и разработки информационных систем.

РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДОПОЛНИТЕЛЬНОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

Результатом освоения дополнительного профессионального образования является овладение обучающимися видами профессиональной деятельности Проектирование и разработка информационных систем, Разработка дизайна веб-приложений, Проектирование, разработка и оптимизация веб-приложений, в том числе профессиональными (ПК) и общими (ОК) компетенциями:

Код	Наименование результата обучения
ПК 1.1.	Собирать исходные данные для разработки проектной документации на информационную систему.
ПК 1.2.	Разрабатывать проектную документацию на разработку информационной системы в соответствии с требованиями заказчика
ПК 1.3.	Производить разработку модулей информационной системы в соответствии с техническим заданием
ПК 1.4.	Осуществлять тестирование информационной системы на этапе опытной эксплуатации с фиксацией выявленных ошибок кодирования в разрабатываемых модулях информационной системы
ПК 1.5.	Разрабатывать техническую документацию на эксплуатацию информационной системы
ПК 2.1	Разрабатывать дизайн-концепции веб-приложений в соответствии с корпоративным стилем заказчика
ПК 2.2	Формировать требования к дизайну веб-приложений на основе анализа предметной области и целевой аудитории.
ПК 2.3	Осуществлять разработку дизайна веб-приложения с учетом современных тенденций в области веб-разработки
ПК 3.1.	Разрабатывать техническое задание на веб-приложение в соответствии с требованиями заказчика
ПК 3.2.	Разрабатывать веб-приложение в соответствии с техническим заданием
ПК 3.3.	Разрабатывать интерфейс пользователя веб-приложений в соответствии с техническим заданием
ПК 3.4.	Производить тестирование разработанного веб приложения
ПК 3.5.	Осуществлять аудит безопасности веб-приложения в соответствии с регламентами по безопасности
ОК 1.	Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.
ОК 2.	Организовывать собственную деятельность, определять методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.

ОК 3.	Решать проблемы, оценивать риски и принимать решения в нестандартных ситуациях.
ОК 4.	Осуществлять поиск, анализ и оценку информации, необходимой для постановки и решения профессиональных задач, профессионального и личностного развития.
ОК 5.	Использовать информационно-коммуникационные технологии для совершенствования профессиональной деятельности.
ОК 6.	Работать в коллективе и команде, обеспечивать ее сплочение, эффективно общаться с коллегами, руководством, потребителями.
ОК 7.	Ставить цели, мотивировать деятельность подчиненных, организовывать и контролировать их работу с принятием на себя ответственности за результат выполнения заданий.
ОК 8.	Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.
ОК 9.	Быть готовым к смене технологий в профессиональной деятельности.

III. Организационно-педагогические условия реализации ДПП

1. Реализация Программы должна обеспечить получение компетенции, необходимой для выполнения нового вида профессиональной деятельности в области информационных технологий - создание алгоритмов и программ, пригодных для практического применения; приобретение новой квалификации «Разработчик».
2. Учебный процесс организуется с применением электронного обучения и дистанционных образовательных технологий, инновационных технологий и методик обучения, способных обеспечить получение слушателями знаний, умений и навыков в области создания и применение современных технологий.
3. Реализация Программы обеспечивается по средствам привлечения к образовательному процессу высококвалифицированных специалистов ИТ-сферы и/или дополнительного профессионального образования в части, касающейся профессиональных компетенций в области создания алгоритмов и программ, пригодных для практического применения, с обязательным участием представителей профильных организаций-работодателей. Возможно привлечение региональных руководителей цифровой трансформации (отраслевых ведомственных и/или корпоративных) к проведению итоговой аттестации.

IV. Учебный план ДПП

1. Объем Программы составляет 1152 академических часа.
2. Учебный план Программы определяет перечень, последовательность, общую трудоемкость разделов и формы контроля знаний.

Учебный план программы профессиональной переподготовки «Разработка программного обеспечения»

№	Название разделов (тем)	Всего часов	В том числе:		Форма итогового контроля
			Т	П	
Первый семестр					
1.	Основы информационных технологий	56	28	28	Экзамен
2.	Программирование на языке С#	176	88	88	
3.	Теория баз данных	36	18	18	
4.	Промежуточная аттестация	4		4	Зачет
Второй семестр					
5.	Работа с базами данных	48	24	24	
6.	Объектно-ориентированное программирование на С#	224	112	112	
7.	Паттерны проектирования	36	18	18	
8.	Промежуточная аттестация	4		4	Зачет
Третий семестр					
9.	Разработка web-страниц на языке разметки HTML5 с использованием каскадных таблиц стилей CSS3	56	28	28	Практическое задание
10.	Язык сценариев JavaScript. ООП в JavaScript, jQuery, AngularJS	88	44	44	Практическое задание
11.	Расширяемый язык разметки для создания Web-страниц — XML	36	18	18	Практическое задание
12.	Создание web-приложений, исполняемых на стороне сервера с помощью языка программирования PHP и технологии AJAX.	88	44	44	Практическое задание
13.	Промежуточная аттестация	4		4	Зачет
Четвертый семестр					
14.	Разработка приложений с применением технологии Клиент-Сервер	90	44	46	Практическое задание
15.	Создание приложений для платформы Microsoft .NET	88	44	44	Практическое задание
16.	Разработка web-приложений ASP.NET на языке С# с применением СУБД MS SQL Server, источников данных XML и технологии AJAX	36	18	18	Практическое задание
17.	Создание облачных решений с использованием Microsoft Azure	56	28	30	Практическое задание

1.	<p>Раздел 1. Основы информационных технологий <i>Основные темы:</i> Базовое аппаратное обеспечение PC IP Адреса, шлюз, DHCP Фаерволы BIOS, RAID Модель OSI Виртуализация Операционные системы Работа с Linux Написание скриптов Bash Сети. Основные понятия Сети. Практика. Коммутация и маршрутизация Сети. Практика. Основные протоколы TCP/IP <i>Краткое содержание</i> Программа предназначена для изучения основ информационных технологий, аппаратного обеспечения ПК, принципов работы сетей, сетей коммутации и маршрутизации, а также основных TCP/IP протоколов. Полученные знания помогут студентам лучше понимать принципы работы компьютерных систем и сетей, что является важным аспектом для ИТ-специалистов.</p>	56
2.	<p>Раздел 2. Программирование на языке C# <i>Основные темы:</i> IDE и среда разработки Алгоритмические основы программирования Переменные и типы данных Классы Условные операторы Циклы Типы данных Массив <i>Краткое содержание</i> Программа предназначена для обучения основам алгоритмического программирования и работы с различными типами данных, включая классы, переменные, массивы и другие. В рамках курса изучаются условные операторы и циклы, что позволяет студентам освоить базовые принципы разработки и получить навыки структурного подхода к решению задач. Освоение IDE и сред разработки дает возможность студентам научиться работать с кодом на практике и получить опыт создания собственных проектов.</p>	176
3	<p>Раздел 3. Теория баз данных Язык структурированных запросов SQL Создание запросов на SQL Проектирование баз данных Основы проектирования баз данных Создание баз данных Нормализация баз данных Нормальные формы Создание форм в СУБД Многотабличные и вложенные запросы, агрегатные функции. ASP.NET MVC Создание форм в СУБД <i>Краткое содержание</i> Программа предназначена для обучения разработке и проектированию баз данных с использованием языка структурированных запросов (SQL). Студенты изучат основные аспекты создания и нормализации баз данных, а также создание</p>	36

	форм в системах управления базами данных (СУБД). Кроме того, раздел включает обучение многотабличным запросам, агрегатным функциям и основам платформы ASP.NET MVC.	
	<p>Раздел 4 Работа с базами данных</p> <p><i>Основные темы:</i></p> <p>Операции с таблицами в SQL</p> <p>Групповые операции и агрегирование данных</p> <p>Практика операций с таблицами, выборки и агрегации</p> <p>Практика. Соединение таблиц</p> <p>Изучение операторов JOIN, множественных операций</p> <p>Транзакция</p> <p>Практика. Представления, временные таблицы и оконные функции</p> <p>Практика. Серверное программирование</p> <p><i>Краткое содержание</i></p> <p>Раздел предназначен для углубленного изучения операций с таблицами и данными в SQL, включая групповые операции, агрегирование и соединение таблиц. Студенты также изучат концепцию транзакций, представления, временные таблицы, оконные функции и основы серверного программирования.</p>	48
4	<p>Раздел 5. Объектно-ориентированное программирование на C#</p> <p><i>Основные темы:</i></p> <p>Принципы ООП на языке C#.</p> <p>Конструкторы.</p> <p>Деструктор.</p> <p>Виртуальные функции.</p> <p>Абстрактные классы.</p> <p>Коллекции</p> <p>Анонимные методы, лямбда, подробнее о делегатах</p> <p>Работа с JSON</p> <p>Введение в многопоточность</p> <p>Работа с CSV файлами</p> <p>Сериализация XML файлов</p> <p>Виртуализация Debian</p> <p>Поля, свойства, ключи</p> <p>WPF. Создание формы регистрации</p> <p>WPF. Привязка данных и шаблоны ListBox</p> <p><i>Краткое содержание</i></p> <p>Программа предназначена для освоения основ объектно-ориентированного программирования на языке C#. Студенты изучат принципы ООП, такие как инкапсуляция, наследование и полиморфизм, а также научатся использовать ключевые концепции языка, такие как конструкторы, деструкторы, виртуальные функции, абстрактные классы и коллекции. Курс также включает изучение работы с JSON, многопоточности, CSV и XML файлами, а также использование WPF для создания пользовательских интерфейсов. Виртуализация Debian и поля, свойства, ключи, также рассматриваются в рамках курса.</p>	224

5	<p>Раздел 6. Паттерны проектирования</p> <p><i>Основные темы:</i></p> <p>Структура и компоненты языка UML Диаграммы вариантов использования Диаграммы последовательности Диаграммы кооперации Диаграммы классов Диаграммы состояний Диаграммы деятельности Диаграммы компонентов Диаграммы развертывания Паттерны проектирования и их типы Порождающие Структурные Поведенческие</p> <p><i>Краткое содержание:</i></p> <p>Раздел посвящен изучению паттернов проектирования в контексте языка UML (Unified Modeling Language). Рассматриваются различные типы паттернов: порождающие, структурные и поведенческие. Для каждого типа паттерна обсуждаются его назначение, структура, участники и примеры использования. Также изучаются основные компоненты и диаграммы языка UML: варианты использования, последовательность, кооперация, классы, состояния, деятельность, компоненты и развертывание.</p>	36
	<p>Раздел 6. Разработка web-страниц на языке разметки HTML5 с использованием каскадных таблиц стилей CSS3</p> <p><i>Основные темы:</i></p> <p>Форматирование текста при помощи HTML. Работа в HTML. Форматирование при помощи CSS. Основы работы со стилями CSS. Работа с CSS. Списки. Списки и другие способы группировки контента. Использование таблиц. Формы - основы верстки контента. HTML5. Практика: Работа с HTML5. CSS3. Определение и подключение стилей на веб-страницу, важность и валидация кода CSS, каскадные таблицы стилей, определение селекторов. Практика: Работа с CSS.</p> <p><i>Краткое содержание:</i></p> <p>Раздел посвящен обучению разработке веб-страниц с использованием HTML5 и CSS3. Студенты изучат основы форматирования текста и работы с HTML, научатся применять стили CSS для оформления веб-страниц, использовать списки и таблицы, а также работать с формами. Практические задания помогут закрепить полученные знания и навыки.</p>	56
	<p>Раздел 7. Язык сценариев JavaScript. ООП в JavaScript, jQuery, AngularJS</p> <p><i>Основные темы:</i></p> <p>ООП в JavaScript. Работа с JavaScript. jQuery. jQuery: обзор и основные функции библиотеки. Работа с jQuery.</p>	88

	<p>AngularJS. Angular.JS: введение и основные концепции. Практика: Работа с AngularJS. <i>Краткое содержание:</i> Курс посвящен изучению основ объектно-ориентированного программирования в JavaScript, а также библиотек и фреймворков jQuery и AngularJS. Студенты научатся работать с данными инструментами, освоят их основные функции и получают практический опыт создания веб-приложений с использованием изученных технологий.</p>	
	<p>Раздел 8. Расширяемый язык разметки для создания Web-страниц — XML <i>Основные темы:</i> Расширяемый язык разметки для создания Web-страниц — XML. Структура XML документа. Применение XML. Практика: Работа с Web-страниц — XML. Создание web-приложений, исполняемых на стороне сервера с помощью языка программирования PHP и технологии AJAX. AJAX + PHP: применение, взаимодействие, пример. Практика: Работа с AJAX + PHP. <i>Краткое содержание:</i> Курс посвящен изучению расширяемого языка разметки XML и созданию веб-приложений на стороне сервера с использованием технологии AJAX и языка программирования PHP. Студенты узнают о структуре XML-документов, принципах их создания и применения, а также на практике освоят работу с XML-страницами и создание веб-приложений с использованием AJAX и PHP.</p>	36
	<p>Раздел 9. Создание web-приложений, исполняемых на стороне сервера с помощью языка программирования PHP и технологии AJAX <i>Основные темы:</i> Общие сведения о веб-службах ASP.NET AJAX. Вызов веб-служб с помощью ASP.NET AJAX. Конфигурация веб-служб. Создание веб-служб AJAX-Enabled. Использование атрибута ScriptService Создание веб-служб AJAX-Enabled. Взаимодействие с MS SQL Server. База данных MS SQL Server. Работа с базой данных MS SQL. <i>Краткое содержание:</i> Раздел посвящен созданию веб-приложений исполняемых на стороне сервера с использованием языка программирования PHP и технологии AJAX, а также взаимодействию с базой данных Microsoft SQL Server. Студенты изучают общие сведения о веб-службах ASP.NET AJAX, создание и настройку веб-служб, взаимодействие с базой данных и работу с данными при помощи AJAX.</p>	88
	<p>Раздел 10. Разработка приложений с применением технологии Клиент-Сервер. <i>Основные темы:</i> Введение в технологию “Клиент-Сервер”. Протоколы и стандарты технологии “Клиент-Сервер”. Основы серверных технологий: Apache, Nginx, IIS. Основы клиентских технологий: HTML, CSS, JavaScript. Разработка клиентской части приложения с использованием фреймворка Bootstrap. Разработка серверной части приложения на языке программирования PHP с использованием баз данных MySQL.</p>	90

	<p>Создание RESTful API для взаимодействия клиента и сервера. Работа с формами на клиентской стороне с использованием JavaScript и отправка данных на сервер. Обработка ошибок и исключений на сервере и клиенте. Безопасность и защита данных в приложениях “Клиент-Сервер”.</p> <p><i>Краткое содержание:</i> В блоке “Разработка приложений с применением технологии “Клиент-Сервер”” студенты изучают принципы и стандарты технологии “Клиент-Сервер”, разработку клиентской и серверной частей приложения, создание RESTful API, работу с формами и базами данных, обработку ошибок и безопасность данных. Обучение включает в себя разработку веб-приложения с использованием HTML, CSS, Bootstrap, JavaScript, PHP, MySQL, а также тестирование, отладку и оптимизацию клиент-серверных систем.</p>	
	<p>Раздел 11. Создание приложений для платформы Microsoft .NET</p> <p><i>Основные темы:</i> Базовый язык реализации .NET решений — C#. Библиотека базовых классов BCL. Разработка приложений с использованием Windows Forms. Взаимодействие с источниками данных с помощью ADO.NET. ADO.NET (ActiveX Data Object.NET) – набор классов, используемый для доступа к источникам данных в платформе .NET. Практика: Работа с ADO.NET. Платформа Windows Presentation Foundation (WPF). Основы создания клиентских приложений для настольных систем Windows с привлекательным пользовательским интерфейсом. Практика: Создание клиентских приложений. LINQ. Общие сведения о LINQ. LINQ (Language-Integrated Query) основные возможности выполнения запросов на уровне языка и API функции высшего порядка в C# и Visual Basic для написания выразительного декларативного кода. Практика: Создание запросов с использованием LINQ. Entity Framework. Практика: Работа с Entity Framework.</p> <p><i>Краткое содержание:</i> Курс посвящен созданию приложений для платформы Microsoft .NET с использованием языка C# и различных технологий, таких как Windows Forms, ADO.NET, LINQ и Entity Framework. Студенты изучают базовые классы BCL, разработку приложений с Windows Forms, взаимодействие с базами данных через ADO.NET и практикуются в создании клиентских приложений с использованием WPF. Курс также охватывает LINQ - декларативный язык запросов и Entity Framework - ORM-решение для работы с Entity Framework Core.</p>	88
	<p>Раздел 12. Разработка web-приложений ASP.NET на языке C# с применением СУБД MS SQL Server, источников данных XML и технологии AJAX</p> <p><i>Основные темы:</i> Введение в ASP.NET: основные понятия и компоненты. Работа с протоколом HTTP в ASP.NET приложениях. Основы языка программирования C#: типы данных, переменные, операторы, циклы и условные конструкции. Консольное приложение на C#: разработка, компиляция и запуск. XML: структура документа, элементы и атрибуты, валидация XML.</p>	36

	<p>AJAX: основные принципы работы, методы и объекты XMLHttpRequest.</p> <p>Создание web-приложения на ASP.NET с использованием C# и MS SQL Server: разработка и тестирование.</p> <p>Безопасность в web-приложениях: аутентификация, авторизация и шифрование данных.</p> <p>Отладка и оптимизация кода на C# и ASP.NET приложений.</p> <p><i>Краткое содержание:</i></p> <p>Курс посвящен разработке веб-приложений на платформе ASP.NET с использованием языка программирования C# и СУБД MS SQL Server. Студенты изучат основные компоненты и протоколы HTTP, разработают консольное приложение и научатся работать с XML и AJAX для создания безопасных и оптимизированных веб-приложений.</p>	
	<p>Раздел 13. Создание облачных решений с использованием Microsoft Azure</p> <p>Программирование с использованием технологии Java</p> <p><i>Основные темы:</i></p> <p>Потоки в Java.</p> <p>Практика: Работа с потоками Java.</p> <p>Spring.</p> <p>Практика: Работа с возможностями Spring.</p> <p>Hibernate.</p> <p>Работа с возможностями Hibernate.</p> <p>JSP.</p> <p>Практика: Работа с возможностями JSP.</p> <p>MVC.</p> <p>Архитектура MVC на Java с примером разработки приложения JSP.</p> <p>Практика: Работа с возможностями MVC.</p> <p><i>Краткое содержание:</i></p> <p>Раздел посвящен созданию облачных решений с использованием платформы Microsoft Azure и языка программирования Java. Студенты изучают основы многопоточного программирования, работу с Spring, Hibernate и JSP, а также рассматривают архитектуру MVC на примере разработки веб-приложения.</p>	56
7	<p>Основные темы: Итоговая аттестация</p> <p>Краткое содержание:</p> <p>В рамках итоговой аттестации слушатели выполняют проект, имеющий практическую направленность. Результаты выполнения проекта представляются к защите и позволяют оценить степень развития профессиональных компетенций.</p>	20

VI. Система оценки качества освоения образовательной программы

Текущий контроль знаний, обучающихся проводится на протяжении всего обучения по программе преподавателем, ведущим занятия в учебной группе.

Текущий контроль знаний включает в себя наблюдение преподавателя за учебной работой обучающихся и проверку качества знаний, умений и навыков, которыми они овладели на определенном этапе обучения посредством выполнения упражнений на практических занятиях и в иных формах, установленных преподавателем.

Промежуточная аттестация - Оценка качества усвоения обучающимися содержания учебного материала непосредственно по завершению его освоения, проводимая в форме Зачета посредством подготовки и презентации проекта, в соответствии с учебным планом и календарным графиком.

Итоговая аттестация - процедура, проводимая с целью установления уровня знаний, обучающихся с учетом прогнозируемых результатов обучения и требований к результатам освоения образовательной программы.

Итоговая аттестация обучающихся осуществляется в форме защиты проекта. Представленный проект демонстрирует качество полученных навыков, определяет уровень усвоения обучающимися учебного и практического материала и охватывает все содержание, установленное соответствующей дополнительной профессиональной образовательной программой.

Лицам, успешно прошедшим итоговую аттестацию, выдается Диплом.

Для аттестации обучающихся на соответствие их персональных достижений требованиям соответствующей ООП созданы фонды оценочных средств, включающие типовые задания, тесты и методы контроля, позволяющие оценить знания, умения и уровень приобретенных компетенций.

При разработке оценочных средств для контроля качества изучения дисциплин учтены все виды связей между включенными в них знаниями, умениями, навыками, позволяющие установить качество сформированных у обучающихся компетенций по видам деятельности и степень общей готовности выпускников к профессиональной деятельности.

В качестве оценочных средств для проведения текущего контроля и аттестации в целом по всем дисциплинам учебного плана используются как инновационные формы контроля (тесты (действия, ситуационные), модульно-рейтинговая система), так и классические (контрольные работы, модульно-рейтинговая система).

Фонды оценочных средств соответствуют целям и задачам дополнительной общеобразовательной программы, учебному плану и обеспечивают оценку качества компетенций, приобретаемых обучающимся.

Критерии оценки знаний обучающихся

Знания	Умения	Навыки (трудовые функции)
Разработка и отладка программного кода (А)		
<ul style="list-style-type: none"> Методы и приемы формализации задач Языки формализации функциональных спецификаций Методы и приемы алгоритмизации поставленных задач Нотации и программные продукты для графического отображения алгоритмов Алгоритмы решения типовых задач, области и способы их применения 	<ul style="list-style-type: none"> Использовать методы и приемы формализации задач Использовать методы и приемы алгоритмизации поставленных задач Использовать программные продукты для графического отображения алгоритмов Применять стандартные алгоритмы в соответствующих областях 	<p>Формализация и алгоритмизация поставленных задач А/01.3</p> <ul style="list-style-type: none"> Составление формализованных описаний решений поставленных задач в соответствии с требованиями технического задания или других принятых в организации нормативных документов Разработка алгоритмов решения поставленных задач в соответствии с требованиями технического задания или других принятых в организации нормативных документов Оценка и согласование сроков выполнения поставленных задач
<ul style="list-style-type: none"> Синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования Методологии разработки программного обеспечения Методологии и технологии проектирования и использования баз данных Технологии программирования Особенности выбранной среды программирования и системы управления базами данных Компоненты программно-технических архитектур, существующие приложения и интерфейсы взаимодействия с ними 	<ul style="list-style-type: none"> Применять выбранные языки программирования для написания программного кода Использовать выбранную среду программирования и средства системы управления базами данных Использовать возможности имеющейся технической и/или программной архитектуры 	<p>Написание программного кода с использованием языков программирования, определения и манипулирования данными А/02.3</p> <ul style="list-style-type: none"> Создание программного кода в соответствии с техническим заданием (готовыми спецификациями) Оптимизация программного кода с использованием специализированных программных средств Оценка и согласование сроков выполнения поставленных задач

<ul style="list-style-type: none"> • Инструментарий для создания и актуализации исходных текстов программ • Методы повышения читаемости программного кода • Системы кодировки символов, форматы хранения исходных текстов программ • Нормативные документы, определяющие требования к оформлению программного кода 	<ul style="list-style-type: none"> • Применять нормативные документы, определяющие требования к оформлению программного кода • Применять инструментарий для создания и актуализации исходных текстов программ • Применять имеющиеся шаблоны для составления технической документации 	<p>Оформление программного кода в соответствии с установленными требованиями А/03.3</p> <ul style="list-style-type: none"> • Приведение наименований переменных, функций, классов, структур данных и файлов в соответствие с установленными в организации требованиями • Структурирование исходного программного кода в соответствии с установленными в организации требованиями • Комментирование и разметка программного кода в соответствии с установленными в организации требованиями • Форматирование исходного программного кода в соответствии с установленными в организации требованиями
<ul style="list-style-type: none"> • Возможности используемой системы контроля версий и вспомогательных инструментальных программных средств • Установленный регламент использования системы контроля версий 	<ul style="list-style-type: none"> • Использовать выбранную систему контроля версий • Использовать вспомогательные инструментальные программные средства для обработки исходного текста программного кода • Выполнять действия, соответствующие установленному регламенту используемой системы контроля версий 	<p>Работа с системой контроля версий А/04.3</p> <ul style="list-style-type: none"> • Регистрация изменений исходного текста программного кода в системе контроля версий • Слияние, разделение и сравнение исходных текстов программного кода • Сохранение сделанных изменений программного кода в соответствии с регламентом контроля версий
<ul style="list-style-type: none"> • Методы и приемы отладки программного кода • Типы и форматы сообщений об ошибках, предупреждений • Способы использования технологических журналов, форматы и типы записей журналов • Современные компиляторы, отладчики и оптимизаторы программного кода • Сообщения о состоянии аппаратных средств 	<ul style="list-style-type: none"> • Выявлять ошибки в программном коде • Применять методы и приемы отладки программного кода • Интерпретировать сообщения об ошибках, предупреждения, записи технологических журналов • Применять современные компиляторы, отладчики и оптимизаторы программного кода 	<p>Проверка и отладка программного кода А/05.3</p> <ul style="list-style-type: none"> • Анализ и проверка исходного программного кода • программного кода на уровне программных модулей • Отладка программного кода на уровне межмодульных взаимодействий и взаимодействий с окружением • Оценка и согласование сроков выполнения поставленных задач
<p>Проверка работоспособности и рефакторинг кода программного обеспечения (В)</p>		
<ul style="list-style-type: none"> • Методы автоматической и автоматизированной проверки работоспособности программного обеспечения • Основные виды диагностических данных и способы их представления • Языки, утилиты и среды программирования, и средства пакетного выполнения процедур • Типовые метрики программного обеспечения • Основные методы измерения и оценки характеристик программного обеспечения 	<ul style="list-style-type: none"> • Писать программный код процедур проверки работоспособности программного обеспечения на выбранном языке программирования • Использовать выбранную среду программирования для разработки процедур проверки работоспособности программного обеспечения на выбранном языке программирования 	<p>Разработка процедур проверки работоспособности и измерения характеристик программного обеспечения В/01.4</p> <ul style="list-style-type: none"> • Разработка процедуры проверки работоспособности программного обеспечения • Разработка процедуры сбора диагностических данных • Разработка процедуры измерения требуемых характеристик программного обеспечения • Оценка и согласование сроков выполнения поставленных задач
<ul style="list-style-type: none"> • Методы создания и документирования контрольных примеров и тестовых наборов данных • Правила, алгоритмы и технологии создания тестовых наборов данных • Требования к структуре и форматам хранения тестовых наборов данных 	<ul style="list-style-type: none"> • Разработка и оформление контрольных примеров для проверки работоспособности программного обеспечения • Разработка процедур генерации тестовых наборов данных с заданными характеристиками • Подготовка наборов данных, используемых в процессе проверки работоспособности программного обеспечения 	<p>Разработка тестовых наборов данных В/02.4</p> <ul style="list-style-type: none"> • Подготовка тестовых наборов данных в соответствии с выбранной методикой • Оценка и согласование сроков выполнения поставленных задач
<ul style="list-style-type: none"> • Методы и средства проверки работоспособности программного обеспечения 	<ul style="list-style-type: none"> • Применять методы и средства проверки работоспособности программного обеспечения 	<p>Проверка работоспособности программного обеспечения В/03.4</p> <ul style="list-style-type: none"> • Проверка работоспособности программного обеспечения на основе

<ul style="list-style-type: none"> Среда проверки работоспособности и отладки программного обеспечения Внутренние нормативные документы, регламентирующие порядок документирования результатов проверки работоспособности программного обеспечения 	<ul style="list-style-type: none"> Интерпретировать диагностические данные (журналы, протоколы и др.) Анализировать значения полученных характеристик программного обеспечения Документировать результаты проверки работоспособности программного обеспечения 	<p>разработанных тестовых наборов данных</p> <ul style="list-style-type: none"> Оценка соответствия программного обеспечения требуемым характеристикам Сбор и анализ полученных результатов проверки работоспособности программного обеспечения Оценка и согласование сроков выполнения поставленных задач
<ul style="list-style-type: none"> Методы и средства рефакторинга и оптимизации программного кода Языки программирования и среды разработки Внутренние нормативные документы, регламентирующие требования к программному коду, порядок отражения изменений в системе контроля версий Внутренние нормативные документы, регламентирующие порядок отражения результатов рефакторинга и оптимизации в коллективной базе знаний 	<ul style="list-style-type: none"> Применять методы, средства для рефакторинга и оптимизации Применять инструментальные средства коллективной работы над программным кодом Публиковать результаты рефакторинга и оптимизации в коллективной базе знаний в виде лучших практик Использовать систему контроля версий для регистрации произведенных изменений 	<p>Рефакторинг и оптимизация программного кода В/04.4</p> <ul style="list-style-type: none"> Анализ программного кода на соответствие требованиям по читаемости и производительности Внесение изменений в программный код и проверка его работоспособности Оценка и согласование сроков выполнения поставленных задач
<ul style="list-style-type: none"> Методы и приемы отладки программного кода Типовые ошибки, возникающие при разработке программного обеспечения, и методы их диагностики и исправления 	<ul style="list-style-type: none"> Применять методы и приемы отладки дефектного программного кода Интерпретировать сообщения об ошибках, предупреждения, записи технологических журналов, возникающих при выполнении дефектного кода 	<p>Исправление дефектов, зафиксированных в базе данных дефектов В/05.4</p> <ul style="list-style-type: none"> Воспроизведение дефектов, зафиксированных в базе данных дефектов Установление причин возникновения дефектов Внесение изменений в программный код для устранения выявленных дефектов Оценка и согласование сроков выполнения поставленных задач

Критерии оценки промежуточной аттестации – зачета:

1. Оценка «зачтено» выставляется обучающемуся, который

- прочно усвоил предусмотренный программный материал;
- показал глубокие систематизированные знания, владеет навыками разработки информационных технологий, представил на аттестацию законченный проект, владеет навыками презентации и аргументации;

Дополнительным условием получения оценки «зачтено» могут стать активность на занятиях в процессе изучения материала, систематическая активная работа на практических занятиях.

2. Оценка «не зачтено» выставляется обучающемуся, который не справился с 60% заданий, при разработке проекта допустил существенные ошибки. Не способен создать целостный информационный продукт. Навыков разработки и представления проекта у обучающегося нет.

Критерии оценки итоговой аттестации:

Оценка «зачтено».

«Зачет» выставляется обучающемуся, если он правильно применяет теоретические положения при решении практических вопросов, владеет необходимыми навыками и приемами их выполнения. Обучающийся представил проект на должном уровне, аргументировал выбор решений, применил полученные знания и навыки в достаточной мере.

Оценка «не зачтено».

«Не зачтено» выставляется обучающемуся, который допускает существенные ошибки, неуверенно, с большими затруднениями решает практические вопросы или не справляется с ними

самостоятельно. Обучающийся не сумел реализовать проект на должном уровне, демонстрируемые знания и навыки недостаточны для подтверждения уровня знаний.

Фонд оценочных средств итоговой аттестации

Итоговая аттестация выпускников программы является заключительным этапом обучения, подтверждающего квалификацию.

К итоговой аттестации допускаются лица, выполнившие требования, предусмотренные курсом обучения по образовательной программе и успешно прошедшие все промежуточные аттестационные испытания по теоретическому и практическому этапам обучения, предусмотренные утвержденным учебным планом программы.

Цель ИА – установить соответствие уровня и качества подготовки выпускника.

Итоговая аттестация проводится в форме защиты проекта.

VII. Материально-техническое и учебно-методическое обеспечение программы

Для проведения занятий и самостоятельной работы обучающихся в организации имеются:

- помещения, оборудованные персональными компьютерами по числу студентов, с выходом в Интернет;
- жидкокристаллический телевизор;
- маркерная доска;
- электронные пособия;
- электронная система внутреннего контроля успеваемости студентов;
- сетевое оборудование;
- наглядные пособия для сетевого администрирования и архитектуры построения персонального компьютера.

VIII. Общие требования к организации образовательного процесса.

При реализации программы профессиональной переподготовки используется лекционно-семинарская система, практические занятия, индивидуальные и групповые консультации, мастер-классы и другие интерактивные формы обучения. Интегративно-дифференцированная организация занятий в процессе подготовки обучающихся позволяет, не только адаптировать образовательный процесс к индивидуальным особенностям и интересам обучающихся, но и обеспечить свободу выбора и вариативность образования, сформировать у обучающихся стремление к самообразованию, способствовать решению индивидуальных задач развития личности.

IX. Список литературы

Основная:

1. Аббасов И.Б. Основы трехмерного моделирования в 3DS MAX 2018. – М.: ДМК Пресс, 2017. – 186 с.
2. Белиовская Л.Г., Белиовский Н.А. Использование LEGO-роботов в инженерных проектах школьников. Отраслевой подход. Учебное пособие. – М.: ДМК Пресс, 2016. – 88 с.
3. Кнут Д.Э. Искусство программирования. Том 1. Выпуск 1. MMIX - RISC-компьютер для нового тысячелетия. – М.: Вильямс, 2017. – 160 с.

4. Кнут Д.Э. Искусство программирования. Том 1. Основные алгоритмы. – М.: Вильямс, 2017. – 720 с.
5. Кнут Д.Э. Искусство программирования. Том 2. Получисленные алгоритмы. – М.: Вильямс, 2017. – 832 с.
6. Кнут Д.Э. Искусство программирования. Том 3. Сортировка и поиск. – М.: Вильямс, 2017. – 824 с.
7. Кнут Д.Э. Искусство программирования. Том 4А. Комбинаторные алгоритмы. Часть 1. – М.: Вильямс, 2018. – 960 с.
8. Кормен Т.Х., Лейзерсон Ч.И., Ривест Р.Л., Штайн К. Алгоритмы. Построение и анализ. – М.: Вильямс, 2016. – 1328 с.
9. Корягин А.В. Образовательная робототехника Lego WeDo. Сборник методических рекомендаций и практикумов. – М.: ДМК Пресс, 2016. – 254 с.
10. Лаврентьев М. Дизайн в пространстве культуры. От арт-объекта до эклектики. – М.: Альпина Паблишер, 2018. – 152 с.
11. Макконнелл С. Совершенный код. – СПб.: БХВ-Петербург, 2017. – 896 с.
12. Митчелл Р. Скрапинг веб-сайтов с помощью Python. – М.: ДМК Пресс, 2016. – 280 с.
13. Огановская Е.Ю., Гайсина С.В., Князева И.В. Робототехника, 3D-моделирование и прототипирование на уроках и во внеурочной деятельности. 5-7, 8(9) классы. – М.: Каро, 2017. – 256 с.
14. Тарапата В.В., Самылкина Н.Н. Робототехника в школе. Методика, программы, проекты. – М.: Лаборатория знаний, 2017. – 112 с.
15. Уильямс Р. Аниматор: набор для выживания. Секреты и методы создания анимации, 3D-графики и компьютерных игр. – М.: Эксмо, 2018. – 392 с.
16. Филиппов С.А. Уроки робототехники. Конструкция. Движение. Управление. – М.: Лаборатория знаний, 2017. – 176 с.
17. Хеллман Д. Стандартная библиотека Python 3. Справочник с примерами. – М.: Вильямс, 2018. – 1376 с.

Дополнительная:

18. Гарднер Л., Григсби Д. Разработка веб-сайтов для мобильных устройств. – М.-СПб.: Питер, 2013. – 448 с.
19. Кнут Д.Э. Искусство программирования. Том 4. Выпуск 2. Генерация всех кортежей и перестановок. – М.: Вильямс, 2008. – 160 с.
20. Кнут Д.Э. Искусство программирования. Том 4. Выпуск 2. Генерация всех деревьев. История комбинаторной генерации. – М.: Вильямс, 2007. – 160 с.
21. Кроудер Д. Создание веб-сайта для чайников. – М.: Вильямс, 2017. – 336 с.
22. Мацеевский Н.С., Степанищев Е.В., Кондратенко Г.И. Реактивные веб-сайты. Клиентская оптимизация в алгоритмах и примерах. – М.: Лаборатория знаний, 2014. – 336 с.
23. Прахов А.А. Blender. 3D-моделирование и анимация. Руководство для начинающих (+ CD-ROM). – СПб.: БХВ-Петербург, 2009. – 266 с.
24. Розенсон И.А. Основы теории дизайна. – М.-СПб.: Питер, 2013. – 256 с.
25. Саммерфилд М. Программирование на Python 3. Подробное руководство. – М.: Символ-Плюс, 2009. – 608 с.