

ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ  
"ЦИФРА САРАТОВ"

Утверждаю



**Дополнительная общеобразовательная -  
общеразвивающая программа.  
«Разработка пользовательских программ на С#»**

Саратов  
2024г.

# **1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

## **Основные характеристики ДООП**

### **Направленность программы**

Данная программа является технологической направленности.

### **Актуальность программы.**

Актуальность программы обусловлена высоким интересом молодежи к IT-сфере и высоким уровнем дефицита кадров на данный момент. Дополнительная общеобразовательная общеразвивающей программа «Разработка пользовательских программ на C#» направлена на формирование компьютерных компетенций, получения практических навыков и теоретических знаний, необходимых для тех, кто хочет стать в будущем профессионалом в программировании.

### **Отличительные особенности**

Образовательная программа реализуется в соответствии с современными педагогическими тенденциями, поддерживает инновационные формы и методы обучения и использует наиболее актуальные цифровые технологии для организации учебного процесса. Представляемая программа имеет существенный ряд отличий от существующих аналогичных программ.

Содержание программы уникально, формированием учебного материала занимались не столько педагоги, сколько технические специалисты в области программирования, алгоритмики и математики под руководством и консультацией специалистов в области педагогики. Задачи, которые ставятся образовательной программой, максимально адаптированы для обучающегося. Отличительная особенность данной программы состоит в одновременном изучении как основных теоретических, так и практических аспектов программирования, что обеспечивает глубокое понимание процесса программирования в целом. Данная программа предполагает логическую связь изучения и практического применения, что обеспечивает хороший уровень освоения образовательной программы, возможность применения полученных знаний в других областях деятельности, таких как робототехника, связь, авиация, космос, медицина, IT. Обучающийся настроен на конечный результат, т.е. на создание своего первого программного продукта для ПК или мобильных устройств. При освоении программы обучающийся узнает, вспоминает и закрепляет принципы работы тех или иных устройств и оборудования.

Спрос на специалистов в области разработки программного обеспечения ежедневно растет. Банки, интернет-провайдеры, предприятия малого, среднего и крупного бизнеса, фирмы, занимающиеся разработкой и тестированием программного обеспечения, государственные предприятия заинтересованы в приёме на работу разработчиков ПО. Выпускники программы смогут работать программистом, тестировщиком программного обеспечения, инженером по

контролю качества, техническим писателем, web-программистом, разработчиком и администратором баз данных.

Программа регламентирует цели, планируемые результаты, содержание, условия и технологии реализации образовательного процесса, оценку качества подготовки обучающихся и включает в себя: учебный план, фонды оценочных средств, программу итоговой аттестации, календарный учебный график и другие материалы, обеспечивающие качество подготовки обучающихся.

## **Цель и задачи**

### **Цель:**

Сформировать у обучающихся практические знания по разработке программного обеспечения. Выработка практических навыков по применению компьютерного оборудования в качестве инструмента творческого самовыражения и интеллектуального развития. Создание условий для развития личности обучающихся посредством приобретения опыта экспериментальной, проектной и учебно-исследовательской деятельности.

### **Задачи Обучающие:**

- научить соблюдать требования безопасности труда;
- научить основным приёмам работы с языком программирования C#;
- научить работать с компилятором C#;
- научить разработки и исследованию алгоритмов, протоколов, программных решений, вычислительных моделей и моделей данных для реализации функций и сервисов систем информационных технологий;
- научить разработки архитектуры, алгоритмических и программных решений системного и прикладного программного обеспечения;
- научить разработки и исследованию математических, информационных и имитационных моделей по тематике выполняемых опытно-конструкторских и прикладных работ;
- научить разработки и выполнению процессов, работ и процедур жизненного цикла информационных систем, программного обеспечения, сервисов систем информационных технологий;
- научить разработки методов и средств тестирования информационных технологий на соответствие стандартам и исходным требованиям;
- научить разработки проектной и программной документации;
- научить создавать работающую программу.

### **Развивающие:**

- способствовать формированию навыков поиска нестандартного подхода к творческим задачам;
- развивать внимательность, аккуратность и мышление программиста;

- формировать умения самостоятельного решения задач по программированию;
- способствовать формированию компетенций познания окружающего мира, изучения постоянно обновляющейся информации;
- формировать навыки правильно строить защиту самостоятельных творческих проектов;
- способствовать формированию мотивации на получение результатов в конкурсах и состязаниях различного уровня.

### **Воспитательные**

- способствовать формированию у учащихся стремления к получению качественного законченного результата;
- формировать навыки самостоятельной работы и работе в коллективе;
- сформировать навыки культуры межличностных отношений на занятиях, соревнованиях.

### **Планируемые результаты освоения программы:**

#### **Обучающиеся будут знать:**

- требования безопасности труда;
- основные языки программирования
- методы разработки программного обеспечения
- способы выполнения простейших программ;
- основные приёмы работы с языком программирования С#;
- способы выполнения сложных (углубленных) программ;
- продвинутые приёмы работы с языком программирования С#.

#### **Обучающиеся будут уметь:**

- Создавать Windows-приложения с использованием клиент-серверных технологий, графики, баз данных и мультимедиа.
- Правильно и в короткие сроки просчитывать алгоритмы решения поставленных задач и стоимость их внедрения.
- Проектировать базы данных различной степени сложности и величины, управлять ими.
- Разрабатывать Интернет-приложения, реагирующие на действия пользователей и представляющие информацию, хранимую в базах данных.
- Автоматизировать работу с базами данных (как с одной, так и с несколькими базами одновременно), в зависимости от постановки задачи или требований заказчика.
- Модифицировать Windows-программы под конкретные заказы. Разрабатывать сетевые приложения.
- Разрабатывать приложения для мобильных устройств на базе Android.
- Создавать утилиты, решающие повседневные проблемы пользователя.
- Создавать приложения и динамические web-узлы на основе платформы .NET.

- Обеспечивать безопасность приложений, используя алгоритмы шифрования, аутентификации и безопасные соединения.
- Тестировать программное обеспечение, используя современные методики и технологии.
- Проектировать программные проекты, независимо от конечного языка реализации. Всеми аспектами и тонкостями программных проектов.
- Документировать создаваемые приложения.
- Создавать облачные решения, используя возможности Microsoft Azure.
- Применять современные методологии для успешной работы команды.
- Использовать паттерны проектирования в повседневной рабочей деятельности.
- Программировать кроссплатформенные решения.

## **Метапредметные**

### **У обучающихся будут развиты (сформированы):**

- навыки поиска нестандартного подхода к творческим задачам;
- внимательность, аккуратность и мышление программиста;
- умение самостоятельно решать задачи по программированию;
- компетенции познания окружающего мира и изучения постоянно обновляющейся информации;
- мотивация на получение результатов в конкурсах и состязаниях различного уровня;
- умение правильно строить защиту самостоятельных творческих проектов.

## **Личностные**

### **У учащихся будут развиты (сформированы):**

- стремление к получению качественного законченного результата;
- навыки самостоятельной работы и работы в коллективе;
- навыки культуры межличностных отношений на занятиях, соревнованиях.

## **2. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ**

Язык реализации: русский язык.

Форма обучения: очная

Особенности реализации.

Особенностью реализации дополнительной общеобразовательной общеразвивающей программы «Разработка пользовательских программ на C#» для обучающихся в возрасте 18 - 59 лет являются тенденции:

1. Простота и доступность, содержание структурировано и несложно для понимания. Используются понятные и простые примеры, основные концепции программирования объясняются на примерах из реальной жизни.

2. **Интерактивность:** В ДОП включены интерактивные задания, чтобы обучающиеся могли практиковаться и получать мгновенную обратную связь. Это помогает им усвоить материал более эффективно и делает обучение более интересным.
3. **Проектная работа:** Обучающимся предлагаются реальные проекты, которые они могут создать с использованием C#. Например, разработка простой игры, создание веб-страницы или автоматизация повседневных задач. Это дает возможность применить свои знания на практике и развить навыки проблемного мышления.
4. **Графический интерфейс:** В процессе обучения используются библиотеки, которые позволяют создавать графический интерфейс для программ. Это помогает обучающимся создавать более интерактивные и визуально привлекательные проекты.
5. **Сотрудничество и обмен опытом:** Много времени уделяется на занятия формированию навыка работать в команде и обмениваться опытом. Это реализуется через парные программирование, групповые проекты или форумы, где обучающиеся делятся своими достижениями и задают вопросы.
6. **Постепенное усложнение:** Обучение начинается с простых задач и постепенно усложняется материал. Построение на уже изученных концепциях помогает обучающимся укрепить свои знания и почувствовать прогресс в своем обучении.
7. **Мотивация и поощрение:** Достижения обучающихся в обучении публично признаются, их работы демонстрируются в группе.
8. **Ресурсы и поддержка:** Обучающимся предоставляется доступ к дополнительным ресурсам, таким как онлайн-курсы, учебникам или видеоурокам.

Образовательный процесс имеет развивающий характер, т. е. направлен на развитие природных задатков учащихся, на реализацию их интересов и способностей. Широко применяются личностно-ориентированные технологии обучения, в центре внимания которых неповторимая личность, стремящаяся к реализации своих возможностей. Выбор методов (способов) обучения зависит от психофизиологических, возрастных особенностей учащихся, темы и формы занятий. При этом в процессе обучения все методы реализуются в теснейшей взаимосвязи. На занятиях создается атмосфера, когда ребята свободно советуются, комментируют, помогают друг другу. Развитие в учащихся самостоятельности, творчества и изобретательности - один из основных приоритетов программы.

Методика проведения занятий предполагает постоянное создание ситуаций успешности, радости от преодоления трудностей в освоении изучаемого материала и при выполнении самостоятельной работы. Этому способствуют совместные обсуждения технологии выполнения заданий, а также поощрение,

создание положительной мотивации, актуализация интереса, олимпиады и конкурсы.

Для профилактики утомляемости на каждом занятии применяются элементы здоровьесберегающих технологий (Комплексы упражнений физкультурных минуток, Комплексы упражнений физкультурных пауз – СанПиН 2.4.3648-20, СанПиН СП 3.1/2.4.3598-20). Учебные занятия по программе организуются очно, а также в виде онлайн-курсов, обеспечивающих для независимо от их места нахождения достижение и оценку результатов обучения путем организации образовательной деятельности в электронной информационно-образовательной среде, к которой предоставляется открытый доступ через информационно-телекоммуникационную сеть Интернет. При дистанционном обучении теоретические занятия проводятся при обязательном онлайн-включении. На практических дистанционных занятиях могут проводиться индивидуальные консультации. Педагог дополнительного образования подключается к платформе, учащиеся могут заходить/выходить в течение всего занятия по мере необходимости. Педагог проводит работу с результатами тестов, проверяет задания, корректирует, комментирует ход работы, выполненные работы учащихся по электронной почте.

### **Условия набора и формирования групп.**

Занятия проводятся в разновозрастных группах. На обучение принимаются все желающие в возрасте 18 - 59 лет. Добор учащихся с целью сохранности наполнения учебной группы допускается при условии наличия у вновь зачисленных учащихся соответствующих знаний, умений, навыков, компетенций, необходимых для продолжения с обучающимися в группе освоения ДОП «Разработка пользовательских программ на C#» с даты их зачисления. Численный состав учащихся от 10 до 16 человек в группе. Рекомендуется количество учащихся в группе – 15 человек.

### **Формы организации и проведения занятий.**

В ходе образовательного процесса применяются различные формы организации деятельности обучающихся и методы обучения (индивидуальные, групповые и т.д.).

Виды занятий по дополнительной общеобразовательной общеразвивающей программе определяются содержанием программы. Предусмотрены лекции, практические и семинарские занятия, лабораторные работы, мастер-классы, выполнение самостоятельной работы, творческие отчеты, другие виды учебных занятий и учебных работ. Кроме того, учебные занятия по программе или ее части могут быть проведены удаленно в форме онлайн-уроков, видеоконференций, вебинаров, онлайн-тестирования. Тематическое и поурочное планирование осуществляется по принципу от простого к сложному. Кроме выполнения работ под руководством педагога обучающимся предлагаются творческие проекты (метод проектов), а также проблемные задания для

самостоятельного выполнения. Удачные авторские находки учащихся при выполнении практических работ выносятся на коллективный сравнительный анализ для мотивации творческой составляющей в процессе обучения.

### **Материально-техническое оснащение**

Для проведения учебного процесса необходимы:

- Рабочее место преподавателя.
- Рабочие места обучающихся с персональными компьютерами и доступом к сети Интернет.
- Сетевое оборудование.
- Доска учебная.
- Плазменный телевизор.

Программное обеспечение:

- Текстовый редактор и другие офисные программы.
- Компилятор С#.

Кадровое обеспечение.

Педагог дополнительного образования

Нормативные документы, регламентирующие разработку дополнительной профессиональной образовательной программы повышения квалификации

- Федеральный закон "Об образовании в Российской Федерации" от 29.12.2012 N 273-ФЗ.
- "Трудовой кодекс Российской Федерации" от 30.12.2001 N 197-ФЗ (ред. от 30.01.2024).
- Приказ Министерства просвещения РФ от 27 июля 2022 г. N 629 "Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам".

Срок освоения программы: 2 года

Объем программы составляет: 1152 академических часов.

Из них: 576 – теоретическая часть, 576 – практическая часть.

### **3. Учебный план**

Учебный план дополнительная общеобразовательная общеразвивающей программа «Разработка пользовательских программ на С#»

№	Название разделов (тем)	Всего часов	В том числе:		Форма итогового контроля
			Т	П	



<b>Первый модуль</b>					
1.	Основы информационных технологий	56	28	28	Экзамен
2.	Программирование на языке С#	176	88	88	
3.	Теория баз данных	36	18	18	
4.	<b>Промежуточная аттестация</b>	<b>4</b>		<b>4</b>	<b>Зачет</b>
<b>Второй модуль</b>					
5.	Работа с базами данных	48	24	24	
6.	Объектно-ориентированное программирование на С#	224	112	112	
7.	Паттерны проектирования	36	18	18	
8.	<b>Промежуточная аттестация</b>	<b>4</b>		<b>4</b>	<b>Зачет</b>
<b>Третий модуль</b>					
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.	<b>Промежуточная аттестация</b>	<b>4</b>		<b>4</b>	<b>Зачет</b>
<b>Четвертый модуль</b>					
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,	36	18	18	Практическое задание

	источников данных XML и технологии AJAX				
17	Создание облачных решений с использованием Microsoft Azure Программирование с использованием технологии Java	56	28	30	Практическое задание
18	<b>Промежуточная аттестация</b>	<b>4</b>		<b>4</b>	<b>Зачет</b>
19	<b>Итоговая аттестация: защита проекта</b>	<b>20</b>		<b>4</b>	<b>Зачет</b>
	<b>Итого:</b>	<b>1152</b>	<b>576</b>	<b>576</b>	

## Содержание

### Первый модуль

#### Основы информационных технологий.

Тема №1. Инструктаж по ТБ. Вводный контроль

Теория: Техника безопасности и правила пожарной безопасности при работе в компьютерном классе и в лаборатории. Представление плана работы.

Организационные вопросы.

Практика: Правила безопасной работы с оборудованием и материалами.

Тема №2. Базовое аппаратное обеспечение ПК

Теория: Базовая конфигурация персонального компьютера. Минимальный комплект аппаратных средств, которых достаточно для работы с компьютером.

Практика: Сбор и соединение основных аппаратных средств ПК.

Тема №3 IP Адреса, шлюз, DHCP

Теория: Системное администрирование, основы настройки локальных и интернет-сетей.

Практика: Сборка и наладка локальной сети.

Тема №4: Фаерволы.

Теория: Фаерволы: какие бывают и как они могут быть реализованы.

Практика: Практическая работа настройка фаерволов.

Тема №5. BIOS, RAID.

Теория: Как создать RAID в настройках BIOS.

Практика: Настройки RAID.

Тема №6: Модель OSI.

Теория: Что такое модель OSI и зачем она нужна: препарлируем слоёный пирог интернета.

Практика: Решение задач повышенной сложности.

### **Программирование на языке C#**

Тема №7. Основы программирования.

Теория: Введение в программирование.

Практика: Создание творческих работ.

Тема №8. Алгоритмы

Теория: Алгоритмы в нашей жизни. Интуитивное понятие алгоритма, характерные черты. Виды алгоритмов. Способы описания алгоритмов. Формализация понятия алгоритма. Современное состояние теории алгоритмов.

Практика: Создание творческих работ

Тема №9. Типы данных.

Теория: Какие типы данных существуют в программировании.

Практика: Решение задач, написание программного кода.

Тема №10. Строка.

Теория: Методы и функции строк.

Практика: Решение задач, написание программного кода.

Тема №11. Массив.

Теория: Что такое массив и как он устроен.

Практика: Решение задач, написание программного кода

Тема №12. Циклы.

Теория: Что такое циклы и как он устроен.

Практика: Решение задач, написание программного кода.

### **Теория баз данных.**

Тема №13. Язык структурированных запросов SQL.

Теория: Что такое язык структурированных запросов SQL и как он устроен.

Практика: Создание запросов на SQL.

Тема №14. Проектирование баз данных.

Теория: Основы проектирования баз данных.

Практика: Создание баз данных.

Тема №15. Нормализация баз данных.

Теория: Как привести данные в форму: что такое нормализация и зачем она нужна.

Практика: Нормализация баз данных.

Тема №16. Нормальные формы.

Теория: Нормальные формы базы данных что это такое и какие они бывают.

Практика: Создание форм в СУБД.

Тема №17. Многотабличные и вложенные запросы, агрегатные функции.  
ASP.NET MVC.

Теория: Что такое многотабличные и вложенные запросы.

Практика: Создание форм в СУБД.

## **Объектно-ориентированное программирование с использованием языка C#.**

Тема №18. Принципы ООП.

Теория: Как ООП помогает разработчикам писать код быстрее и проще.

Практика: Создание творческих работ.

Тема №19. Конструкторы.

Теория: Что такое конструктор в объектно-ориентированном программировании.

Практика: Создание творческих работ.

Тема №20. Деструктор.

Теория: Деструкторы – что нужно знать программисту.

Практика: Создание творческих работ.

Тема №21. Виртуальные функции.

Теория: Виртуальные функции и их переопределение.

Практика: Создание творческих работ.

Тема №22. Абстрактные классы.

Теория: Абстрактные классы и интерфейсы: гайда по ООП.

Практика: Создание СУБД.

Тема №23. Абстрактные классы.

Теория: Виртуальные функции и их переопределение.

Практика: Создание баз данных.

Тема №24. Проектирование баз данных.

Теория: Основы проектирования баз данных.

Практика: Проектирование баз данных.

Тема №25. Проектирование баз данных.

Теория: Основы проектирования баз данных.

Практика: Проектирование баз данных.

### **Программирование и администрирование Microsoft SQL Server 2014.**

Тема №26. Особенности диалекта T-SQL.

Теория: Основы T-SQL и примеры — функции (UDF), триггеры, процедуры, курсоры, циклы.

Практика: Создание творческих работ.

Тема №27. Представления.

Теория: Представление (VIEW) в T-SQL – описание и примеры использования.

Практика: Создание примеров использования T-SQL.

Тема №28. Триггеры.

Теория: Логические триггеры: схемы, классификация, устройство, назначение, применение.

Практика: Создание триггеров.

Тема №29. Хранимые процедуры.

Теория: Функции и хранимые процедуры в SQL: зачем нужны и как применять в реальных примерах.

Практика: Создание процедур.

Тема №30. Пользовательские функции.

Теория: Для чего нужны пользовательские функции в T-SQL (Microsoft SQL Server).

Практика: Создание функций.

Тема №31. Резервное копирование и восстановление.

Теория: Что такое бэкап и зачем он нужен?

Практика: Создание резервной копии.

Тема №32. Репликация, система прав.

Теория: Репликация БД. Управление правами доступа и безопасность приложений?

Практика: Создание системы прав доступа.

Тема №33. Системы управления базами данных: принципы программирования СУБД.

Теория: Найти, сохранить и защитить: как СУБД помогают аналитикам и маркетологам

Практика: Создание примеров взаимодействия СУБД.

Тема №34. Системы управления базами данных: принципы программирования СУБД.

Теория: Найти, сохранить и защитить: как СУБД помогают аналитикам и маркетологам

Практика: Создание примеров взаимодействия СУБД.

### **Системное программирование.**

Тема №35. Потоки.

Теория: Процессы и потоки in-depth. Обзор различных потоковых моделей.

Практика: Создание примеров организации потоков.

Тема №36. Процессы.

Теория: Понятия «процесс», «поток», «задача» являются общесистемными и контекстозависимыми. Они могут описываться как некоторый программный объект, так и выполняемые им действия.

Практика: Создание примеров процессов и задач.

Тема №37. Примитивы синхронизации.

Теория: Общие сведения о примитивах синхронизации.

Практика: Создание примеров синхронизации.

Тема №38. Примитивы синхронизации.

Теория: Общие сведения о примитивах синхронизации.

Практика: Создание примеров синхронизации.

Тема №39. Межпроцессное взаимодействие.

Теория: Синхронизация процессов и потоков. Межпроцессное взаимодействие.

Практика: Создание примеров синхронизации.

Тема №40. Межпроцессное взаимодействие.

Теория: Синхронизация процессов и потоков. Межпроцессное взаимодействие.

Практика: Создание примеров синхронизации.

## **Сетевое программирование.**

Тема №41. Сокеты.

Теория: Что такое сокет и зачем он нужен.

Практика: Работа с сокетами.

Тема №42. Синхронные и асинхронные механизмы обмена данными.

Теория: Разница между синхронной и асинхронной передачей.

Практика: Работа с механизмами обмена данными.

Тема №43. Протоколы.

Теория: Сетевые протоколы: базовые понятия и описание самых востребованных правил.

Практика: Работа с протоколами.

Тема №44. Протоколы.

Теория: Сетевые протоколы: базовые понятия и описание самых востребованных правил.

Практика: Работа с протоколами.

## **Разработка web-страниц на языке разметки HTML5 с использованием каскадных таблиц стилей CSS3:**

Тема №45. Форматирование текста при помощи HTML.

Теория: Основы редактирования текста в HTML.

Практика: Работа в HTML.

Тема №46. Форматирование при помощи CSS.

Теория: Использование CSS для форматирования текста. Основы работы со стилями CSS.

Практика: Работа с CSS.

Тема №47. Списки.

Теория: Списки и другие способы группировки контента.

Практика: Работа со Списками.

Тема №48. Использование таблиц.

Теория: HTML-таблицы: все, что нужно знать о них.

Практика: Работа с таблицами.

Тема №49. Работа с формами.

Теория: Формы - основы верстки контента.

Практика: Работа с формами.

Тема №50. HTML5.

Теория: Как писать на HTML5 и какие у него возможности.

Практика: Работа с HTML5.

Тема №51. CSS3.

Теория: Определение и подключение стилей на веб-страницу, важность и валидация кода CSS, каскадные таблицы стилей, определение селекторов.

Практика: Работа с CSS.

### **Язык сценариев JavaScript. ООП в JavaScript, jQuery, AngularJS.**

Тема №52. ООП в JavaScript.

Теория: JavaScript: ООП на простых примерах. Формы работы и основные принципы построения программ

Практика: Работа с JavaScript.

Тема №53. jQuery.

Теория: jQuery: обзор и основные функции библиотеки.

Практика: Работа с jQuery.

Тема №54. AngularJS.

Теория: AngularJS: введение и основные концепции.

Практика: Работа с AngularJS.

Тема №55. Расширяемый язык разметки для создания Web-страниц — XML.

Теория: Расширяемый язык разметки XML. Синтаксис XML. Структура XML документа. Применение XML.

Практика: Работа с Web-страниц — XML.

Тема №56. Создание web-приложений, исполняемых на стороне сервера с помощью языка программирования PHP и технологии AJAX.

Теория: AJAX + PHP: применение, взаимодействие, пример.

Практика: Работа с AJAX + PHP.



## **Создание приложений для платформы Microsoft .NET:**

Тема №57. Базовый язык реализации .NET решений — C#.

Теория: Язык C#: краткое описание и введение в технологии программирования.

Практика: Работа в C#.

Тема №58. Библиотека базовых классов BCL.

Теория: Что нового в библиотеке базовых классов в .NET 4.5.

Практика: Работа в библиотеках BCL.

Тема №59. Разработка приложений с использованием Windows Forms.

Теория: Введение в приложения C # Windows Forms.

Практика: Работа с Windows Forms.

Тема №60. Взаимодействие с источниками данных с помощью ADO.NET.

Теория: ADO.NET (ActiveX Data Object.NET) – набор классов, используемый для доступа к источникам данных в платформе .NET.

Практика: Работа с ADO.NET.

Тема №61. Разработка приложений с использованием WPF.

Теория: Платформа Windows Presentation Foundation (WPF). Основы создания клиентских приложений для настольных систем Windows с привлекательным пользовательским интерфейсом.

Практика: Создание клиентских приложений.

Тема №62. LINQ.

Теория: 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.

Тема №63. Entity Framework.

Теория: Entity Framework: как быстрее написать код для работы с базой данных.

Практика: Работа с Entity Framework.

## **Разработка web-приложений ASP.NET на языке C# с применением СУБД MS SQL Server, источников данных XML и технологии AJAX:**

Тема №64. ASP.NET MVC.

Теория: ASP.NET MVC основы работы с многофункциональной платформой для создания веб-приложений и API-интерфейсов с помощью структуры проектирования Model-View-Controller.

Практика: Работа с ASP.NET MVC.

Тема №65. ASP.NET и AJAX.

Теория: 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

Практика: Создание веб-служб AJAX-Enabled.

Тема №66. Взаимодействие с MS SQL Server.

Теория: База данных MS SQL Server: что это, зачем нужна, как появилась и чем хороша.

Практика: Работа с базой данных MS SQL.

### **Создание облачных решений с использованием Microsoft Azure. Программирование с использованием технологии Java:**

Тема №67. ООП.

Теория: Принципы ООП на примере языка программирования Java

Знакомство с основами объектно-ориентированного программирования в Java.

Практика: Работа с базой данных MS SQL.

Тема №68. Многопоточность.

Теория: Поток в Java: что это такое и как они работают. Выжимаем максимум из процессора и заставляем программы на Java выполнять несколько задач одновременно.

Практика: Работа с потоками Java.

Тема №69. Spring.

Теория: Spring для ленивых. Основы, базовые концепции и примеры с кодом.

Практика: Работа с возможностями Spring.

Тема №70. Hibernate.

Теория: Как фреймворк Hibernate помогает Java-разработчику облегчить работу с базами данных.

Практика: Работа с возможностями Hibernate.

Тема №71. JSP.

Теория: Что такое JSP? Java Server Pages (JSP) — основы технологии, которая используется для разработки веб-страниц путем вставки Java код в HTML-страницы, создавая специальные теги JSP. Теги JSP, которые позволяют включать в него код Java: `<% —java code—%>`. Разбираемся с возможностями на практике.







## 5. Рабочая программа учебных предметов, курсов, дисциплин (модулей)

Рабочая программа содержит перечень разделов и тем, а также рассматриваемых в них вопросов с учетом их трудоемкости.

Срок освоения программы – 2 года. Начало обучения – по мере набора группы. Примерный режим занятий: 3 дня в неделю по 4 академических часа. В неделю 12 академических часов, соответственно.

№ п/п	Наименование и краткое содержание раздела(модуля)	Объем, часов
1.	<b>Раздел 1. Основы информационных технологий</b> <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.	56
2.	<b>Раздел 2. Программирование на языке C#</b> <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.	176

3	<p>Раздел 3. Теория баз данных</p> <p>Язык структурированных запросов SQL</p> <p>Создание запросов на SQL</p> <p>Проектирование баз данных</p> <p>Основы проектирования баз данных</p> <p>Создание баз данных</p> <p>Нормализация баз данных</p> <p>Нормальные формы</p> <p>Создание форм в СУБД</p> <p>Многотабличные и вложенные запросы, агрегатные функции.</p> <p>ASP.NET MVC</p> <p>Создание форм в СУБД</p> <p><i>Краткое содержание</i></p> <p>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 форм в системах управления базами данных (СУБД). Кроме того, раздел включает обучение многотабличным запросам, агрегатным функциям и основам платформы ASP.NET MVC.</p>	36
---	---	----

	<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>          ООП в JavaScript.          Работа с JavaScript.          jQuery.          jQuery: обзор и основные функции библиотеки.          Работа с jQuery.</p> <p>AngularJS.          AngularJS: введение и основные концепции.          Практика: Работа с AngularJS.</p> <p><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>	88
	<p>Раздел 8. Расширяемый язык разметки для создания Web-страниц — XML</p> <p><i>Основные темы:</i>          Расширяемый язык разметки для создания Web-страниц — XML.          Структура XML документа. Применение XML.          Практика: Работа с Web-страниц — XML.          Создание веб-приложений, исполняемых на стороне сервера с помощью языка программирования PHP и технологии AJAX.          AJAX + PHP: применение, взаимодействие, пример.          Практика: Работа с AJAX + PHP.</p> <p><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 веб-приложений, исполняемых на стороне сервера с помощью языка программирования PHP и технологии AJAX</p> <p><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.</p> <p><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.</p> <p><i>Основные темы:</i></p> <p>Введение в технологию “Клиент-Сервер”.</p> <p>Протоколы и стандарты технологии “Клиент-Сервер”.</p> <p>Основы серверных технологий: Apache, Nginx, IIS.</p> <p>Основы клиентских технологий: HTML, CSS, JavaScript.</p> <p>Разработка клиентской части приложения с использованием фреймворка Bootstrap.</p> <p>Разработка серверной части приложения на языке программирования PHP с использованием баз данных MySQL.</p> <p>Создание RESTful API для взаимодействия клиента и сервера.</p> <p>Работа с формами на клиентской стороне с использованием JavaScript и отправка данных на сервер.</p> <p>Обработка ошибок и исключений на сервере и клиенте.</p> <p>Безопасность и защита данных в приложениях “Клиент-Сервер”.</p> <p><i>Краткое содержание:</i></p> <p>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>	90
	<p>Раздел 11. Создание приложений для платформы Microsoft .NET</p> <p><i>Основные темы:</i></p> <p>Базовый язык реализации .NET решений — C#.</p> <p>Библиотека базовых классов BCL.</p> <p>Разработка приложений с использованием Windows Forms.</p> <p>Взаимодействие с источниками данных с помощью ADO.NET.</p> <p>ADO.NET (ActiveX Data Object.NET) – набор классов, используемый для доступа к источникам данных в платформе .NET.</p> <p>Практика: Работа с ADO.NET.</p> <p>Платформа Windows Presentation Foundation (WPF). Основы создания клиентских приложений для настольных систем Windows с привлекательным пользовательским интерфейсом.</p> <p>Практика: Создание клиентских приложений.</p> <p>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.</p> <p>Практика: Создание запросов с использованием LINQ.</p> <p>Entity Framework.</p> <p>Практика: Работа с Entity Framework.</p> <p><i>Краткое содержание:</i></p> <p>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</p>	88

	<p>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>	
	<p><b>Раздел 12.</b> 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></p> <p>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.          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.          Безопасность в web-приложениях: аутентификация, авторизация и шифрование данных.          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>	36
	<p><b>Раздел 13.</b> 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.          Практика: Работа с потоками Java.          Spring.          Практика: Работа с возможностями Spring.          Hibernate.          Работа с возможностями Hibernate.          JSP.          Практика: Работа с возможностями JSP.          MVC.          Архитектура MVC на Java с примером разработки приложения JSP.          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

<b>7</b>	<p><b>Основные темы: Итоговая аттестация</b></p> <p>Краткое содержание:  В рамках итоговой аттестации слушатели выполняют проект, имеющий практическую направленность. Результаты выполнения проекта представляются к защите и позволяют оценить степень развития профессиональных компетенций.</p>	<b>20</b>
----------	---	-----------

## **6. 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 обучающимися.

## 5.2. Критерии оценки знаний обучающихся

Знания	Умения	Навыки ( трудовые функции)
<b>Разработка и отладка программного кода (А)</b>		
<ul style="list-style-type: none"> <li>• Методы и приемы формализации задач</li> <li>• Языки формализации функциональных спецификаций</li> <li>• Методы и приемы алгоритмизации поставленных задач</li> <li>• Нотации и программные продукты для графического отображения алгоритмов</li> <li>• Алгоритмы решения типовых задач, области и способы их применения</li> </ul>	<ul style="list-style-type: none"> <li>• Использовать методы и приемы формализации задач</li> <li>• Использовать методы и приемы алгоритмизации поставленных задач</li> <li>• Использовать программные продукты для графического отображения алгоритмов</li> <li>• Применять стандартные алгоритмы в соответствующих областях</li> </ul>	<p><b>Формализация и алгоритмизация поставленных задач А/01.3</b></p> <ul style="list-style-type: none"> <li>• Составление формализованных описаний решений поставленных задач в соответствии с требованиями технического задания или других принятых в организации нормативных документов</li> <li>• Разработка алгоритмов решения поставленных задач в соответствии с требованиями технического задания или других принятых в организации нормативных документов</li> <li>• Оценка и согласование сроков выполнения поставленных задач</li> </ul>
<ul style="list-style-type: none"> <li>• Синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования</li> <li>• Методологии разработки программного обеспечения</li> <li>• Методологии и технологии проектирования и использования баз данных</li> <li>• Технологии программирования</li> <li>• Особенности выбранной среды программирования и системы управления базами данных</li> <li>• Компоненты программно-технических архитектур, существующие приложения и интерфейсы взаимодействия с ними</li> </ul>	<ul style="list-style-type: none"> <li>• Применять выбранные языки программирования для написания программного кода</li> <li>• Использовать выбранную среду программирования и средства системы управления базами данных</li> <li>• Использовать возможности имеющейся технической и/или программной архитектуры</li> </ul>	<p><b>Написание программного кода с использованием языков программирования, определения и манипулирования данными А/02.3</b></p> <ul style="list-style-type: none"> <li>• Создание программного кода в соответствии с техническим заданием (готовыми спецификациями)</li> <li>• Оптимизация программного кода с использованием специализированных программных средств</li> <li>• Оценка и согласование сроков выполнения поставленных задач</li> </ul>
<ul style="list-style-type: none"> <li>• Инструментарий для создания и актуализации исходных текстов программ</li> <li>• Методы повышения читаемости программного кода</li> <li>• Системы кодировки символов, форматы хранения исходных текстов программ</li> <li>• Нормативные документы, определяющие требования к оформлению программного кода</li> </ul>	<ul style="list-style-type: none"> <li>• Применять нормативные документы, определяющие требования к оформлению программного кода</li> <li>• Применять инструментарий для создания и актуализации исходных текстов программ</li> <li>• Применять имеющиеся шаблоны для составления технической документации</li> </ul>	<p><b>Оформление программного кода в соответствии с установленными требованиями А/03.3</b></p> <ul style="list-style-type: none"> <li>• Приведение наименований переменных, функций, классов, структур данных и файлов в соответствие с установленными в организации требованиями</li> <li>• Структурирование исходного программного кода в соответствии с установленными в организации требованиями</li> <li>• Комментирование и разметка программного кода в соответствии с установленными в организации требованиями</li> <li>• Форматирование исходного программного кода в соответствии с установленными в организации требованиями</li> </ul>

<ul style="list-style-type: none"> <li>• Возможности используемой системы контроля версий и вспомогательных инструментальных программных средств</li> <li>• Установленный регламент использования системы контроля версий</li> </ul>	<ul style="list-style-type: none"> <li>• Использовать выбранную систему контроля версий</li> <li>• Использовать вспомогательные инструментальные программные средства для обработки исходного текста программного кода</li> <li>• Выполнять действия, соответствующие установленному регламенту используемой системы контроля версий</li> </ul>	<p><b>Работа с системой контроля версий А/04.3</b></p> <ul style="list-style-type: none"> <li>• Регистрация изменений исходного текста программного кода в системе контроля версий</li> <li>• Слияние, разделение и сравнение исходных текстов программного кода</li> <li>• Сохранение сделанных изменений программного кода в соответствии с регламентом контроля версий</li> </ul>
<ul style="list-style-type: none"> <li>• Методы и приемы отладки программного кода</li> <li>• Типы и форматы сообщений об ошибках, предупреждений</li> <li>• Способы использования технологических журналов, форматы и типы записей журналов</li> <li>• Современные компиляторы, отладчики и оптимизаторы программного кода</li> <li>• Сообщения о состоянии аппаратных средств</li> </ul>	<ul style="list-style-type: none"> <li>• Выявлять ошибки в программном коде</li> <li>• Применять методы и приемы отладки программного кода</li> <li>• Интерпретировать сообщения об ошибках, предупреждения, записи технологических журналов</li> <li>• Применять современные компиляторы, отладчики и оптимизаторы программного кода</li> </ul>	<p><b>Проверка и отладка программного кода А/05.3</b></p> <ul style="list-style-type: none"> <li>• Анализ и проверка исходного программного кода</li> <li>• программного кода на уровне программных модулей</li> <li>• Отладка программного кода на уровне межмодульных взаимодействий и взаимодействий с окружением</li> <li>• Оценка и согласование сроков выполнения поставленных задач</li> </ul>
<p><b>Проверка работоспособности и рефакторинг кода программного обеспечения (В)</b></p>		
<ul style="list-style-type: none"> <li>• Методы автоматической и автоматизированной проверки работоспособности программного обеспечения</li> <li>• Основные виды диагностических данных и способы их представления</li> <li>• Языки, утилиты и среды программирования, и средства пакетного выполнения процедур</li> <li>• Типовые метрики программного обеспечения</li> <li>• Основные методы измерения и оценки характеристик программного обеспечения</li> </ul>	<ul style="list-style-type: none"> <li>• Писать программный код процедур проверки работоспособности программного обеспечения на выбранном языке программирования</li> <li>• Использовать выбранную среду программирования для разработки процедур проверки работоспособности программного обеспечения на выбранном языке программирования</li> </ul>	<p><b>Разработка процедур проверки работоспособности и измерения характеристик программного обеспечения В/01.4</b></p> <ul style="list-style-type: none"> <li>• Разработка процедуры проверки работоспособности программного обеспечения</li> <li>• Разработка процедуры сбора диагностических данных</li> <li>• Разработка процедуры измерения требуемых характеристик программного обеспечения</li> <li>• Оценка и согласование сроков выполнения поставленных задач</li> </ul>
<ul style="list-style-type: none"> <li>• Методы создания и документирования контрольных примеров и тестовых наборов данных</li> <li>• Правила, алгоритмы и технологии создания тестовых наборов данных</li> <li>• Требования к структуре и форматам хранения тестовых наборов данных</li> </ul>	<ul style="list-style-type: none"> <li>• Разработка и оформление контрольных примеров для проверки работоспособности программного обеспечения</li> <li>• Разработка процедур генерации тестовых наборов данных с заданными характеристиками</li> <li>• Подготовка наборов данных, используемых в процессе проверки работоспособности программного обеспечения</li> </ul>	<p><b>Разработка тестовых наборов данных В/02.4</b></p> <ul style="list-style-type: none"> <li>• Подготовка тестовых наборов данных в соответствии с выбранной методикой</li> <li>• Оценка и согласование сроков выполнения поставленных задач</li> </ul>
<ul style="list-style-type: none"> <li>• Методы и средства проверки работоспособности программного обеспечения</li> <li>• Среда проверки работоспособности и отладки программного обеспечения</li> <li>• Внутренние нормативные документы, регламентирующие порядок документирования результатов проверки работоспособности программного обеспечения</li> </ul>	<ul style="list-style-type: none"> <li>• Применять методы и средства проверки работоспособности программного обеспечения</li> <li>• Интерпретировать диагностические данные (журналы, протоколы и др.)</li> <li>• Анализировать значения полученных характеристик программного обеспечения</li> <li>• Документировать результаты проверки работоспособности программного обеспечения</li> </ul>	<p><b>Проверка работоспособности программного обеспечения В/03.4</b></p> <ul style="list-style-type: none"> <li>• Проверка работоспособности программного обеспечения на основе разработанных тестовых наборов данных</li> <li>• Оценка соответствия программного обеспечения требуемым характеристикам</li> <li>• Сбор и анализ полученных результатов проверки работоспособности программного обеспечения</li> <li>• Оценка и согласование сроков выполнения поставленных задач</li> </ul>

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

### 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.

### 5.3. 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.

## **7. Материально-техническое и учебно-методическое обеспечение Программы**

Для проведения занятий и самостоятельной работы обучающихся в организации имеются помещения, оборудованные компьютерами, проектором, экраном, доской.

## **7. 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.

## 8.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.
- Саммерфилд М. Программирование на Python 3. Подробное руководство. – М.: