

Областная государственная автономная нетиповая образовательная организация
«Центр выявления и поддержки одарённых детей в Ульяновской области
«Алые паруса»

СОГЛАСОВАНО

Протокол заседания
экспертного совета
«ОГАН ОО Центр «Алые паруса»
от _____ 2026 № ____

СОГЛАСОВАНО

Протокол заседания
педагогического совета
«ОГАН ОО Центр «Алые паруса»
от _____ 2026 № ____

УТВЕРЖДАЮ

Директор «ОГАН ОО Центр «Алые паруса»
_____ Т.А. Хмелевская

Приказ № ____ от «__» ____ 2026

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА**

«Программирование на C#»

направленность: техническая
уровень программы: базовый

Срок реализации программы: 5 месяцев
Возраст обучающихся: 13-17 лет

Автор-разработчик
педагог дополнительного образования
Мердеев Никита Сергеевич

Ульяновск, 2026 г.

Структура дополнительной общеразвивающей программы

1. Комплекс основных характеристик программы	3
1.1 Пояснительная записка	3
1.2 Цели и задачи программы	5
1.3 Содержание программы	6
1.3.1 Учебный план	6
1.4 Планируемые результаты	14
2 Комплекс организационно-педагогических условий	17
2.1 Календарный учебный график.....	17
2.2 Условия реализации программы	22
2.3 Формы аттестации	23
2.4 Критерии оценки	24
2.5 Воспитательный компонент	25
Список литературы	29
Приложение	31

1. Комплекс основных характеристик программы

1.1 Пояснительная записка

Нормативно-правовое обеспечение программы

- Федеральный закон от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации» (ст. 2, ст. 15, ст.16, ст.17, ст.75, ст. 79).
- Приказ Минпросвещения РФ от 27.07.2022 г. № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».
- Методические рекомендации по проектированию дополнительных общеразвивающих программ (Письмо Минобрнауки России от 18.11.2015 г. № 09-3242).
- Постановление Главного государственного санитарного врача Российской Федерации от 28.09.2020 № 28 «Об утверждении СанПиН 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи» (зарегистрировано в Минюсте России 18.12.2020 № 61573)
- Постановление Правительства Российской Федерации об утверждении Правил применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ от 11 октября 2023 г. № 1678.
- Распоряжение Правительства РФ от 31.03.2022 № 678-р «Об утверждении Концепции развития дополнительного образования детей и признании утратившим силу Распоряжения Правительства РФ от 04.09.2014 № 1726-р».
- Устав «ОГАН ОО Центр «Алые паруса».
- Локальные акты «ОГАН ОО Центр «Алые паруса».

Уровень программы: базовый.

Направленность программы: техническая.

Программа «Программирование на C#» направлена на формирование у детей интереса к информационным технологиям, а также выявление творческого потенциала и развитие личности ребенка.

Дополнительность программы по отношению к программам общего образования. Данный вид деятельности в общеобразовательных учреждениях не изучается.

Актуальность программы состоит в том, что она составлена с учётом современных потребностей рынка в специалистах в области информационных технологий, особенно в сфере разработки корпоративного программного обеспечения, веб-сервисов и игровой индустрии.

Научившись программировать на языке C#, обучающиеся получают мощный и универсальный инструмент для решения как учебных, так и прикладных задач в экосистеме .NET. Строгая типизация и объектно-ориентированная парадигма C# позволят учащимся глубже понять внутренние

механизмы работы программ, что с лёгкостью поможет им освоить любой другой компилируемый язык программирования.

Знания и умения, приобретенные в результате освоения курса, могут быть использованы обучающимися при сдаче ОГЭ/ЕГЭ, при участии в олимпиадах по программированию, хакатонах по разработке приложений и профильных компетенциях («Программные решения для бизнеса»), а также они являются фундаментом для дальнейшего совершенствования мастерства промышленной разработки ПО.

Данная программа дает возможность детям творчески мыслить, находить самостоятельные индивидуальные решения, а полученные умения и навыки применять в жизни. Развитие творческих способностей помогает также в профессиональной ориентации подростков.

Новизна программы состоит в том, что она учитывает новые технологические уклады, которые требуют новый способ мышления и тесного взаимодействия при постоянном повышении уровня междисциплинарности проектов, а также использует новые формы диагностики и подведения итогов реализации программы, выполняемые в формате защиты кроссплатформенных приложений и игровых прототипов.

В основу программы «Программирование на C#» заложены принципы модульности практической направленности, что обеспечит вариативность обучения. Содержание учебных модулей направлено на:

- детальное изучение алгоритмизации;
- освоение принципов объектно-ориентированного программирования (ООП);
- проектирование графических интерфейсов пользователя (GUI);
- работу с базовыми структурами данных и коллекциями .NET.

Отличительные особенности программы. Программа предназначена для обучающихся, проявляющих повышенный интерес к программированию. Программа имеет практическую направленность с ориентацией на реальные потребности, соответствующие возрасту ученика.

Ключевым элементом обучения является проектная деятельность, которая ориентирована на использование знаний, умений и навыков, полученных в ходе обучения, для постановки и решения практических задач, которые носят прикладной характер. Она позволяет обучающимся участвовать в создании конкретного результата и научиться работать в условиях ограниченного времени, под руководством заказчика, презентовать проект, а также обрести навыки профессиональной коммуникации с контрагентами.

Педагогическая целесообразность программы заключается в создании организационных и психолого-педагогических условий для привлечения детей к занятиям техническим творчеством, обеспечивающих развитие мотивации к познанию, творчеству и труду, логических способностей, формирование компетенций важных в программировании, как факторов успешного самоопределения и самореализации личности в современном мире.

Адресат программы: 13-17 лет.

Формы обучения: очная.

Формы занятий: чаще всего применяется комбинированные и практические занятия.

Виды занятий: лекции и практические работы. Основной формой является групповое обучение.

Объем программы: 72 часов

Срок освоения программы: 5 месяцев.

Режим занятий: 2 раза в неделю по 2 академических часа с перерывом 10 минут.

1.2 Цели и задачи программы

Цель программы - формирование у обучающихся базовых и продвинутых компетенций в области алгоритмизации и объектно-ориентированного программирования с использованием языка C# и платформы .NET для решения практико-ориентированных задач.

Задачи:

Образовательные:

- дать знания о синтаксисе языка C#, правилах построения выражений, базовых типах данных и структурах управления.
- обучить проектировать алгоритмы различной сложности и реализовывать их в среде разработки Visual Studio
- сформировать понимание принципов объектно-ориентированного программирования (инкапсуляция, наследование, полиморфизм).
- обучить методам отладки, обработки исключений и тестирования программного кода.
- обучить работе с коллекциями данных, файловой системой и внешними библиотеками.

Развивающие:

- развить навыки алгоритмического и логического мышления, грамотной разработки программ;
- развить познавательные способности ребенка, память, внимание, пространственное мышление, аккуратность и изобретательность;
- сформировать творческий подход к поставленной задаче;
- развить навыки инженерного мышления, умения работать как по предложенным инструкциям, так и находить свои собственные пути решения поставленных задач;
- развить навыки эффективной деятельности в проекте;
- развить стрессоустойчивость;
- развить способности к самоанализу, самопознанию;
- сформировать навыки рефлексивной деятельности.
- Развить аналитическое, логическое и алгоритмическое мышление учащихся.

Воспитательные:

- воспитать инженерную культуру, аккуратность при ведении документации проекта и написании чистого кода
- сформировать навыки командной работы над проектом;
- воспитать способности к самореализации и саморазвитию;
- сориентировать обучающихся на получение технической специальности;
- способствовать социализации обучающихся путем приобщения их к совместной работе, а также современным культурным тенденциям в сфере дизайна.

1.3 Содержание программы

1.3.1 Учебный план

№ п\п	Наименование раздела, темы	Теория	Практика	Всего	Форма аттестации/ контроля
1	Повторение стартового уровня и углубление в методы	2	4	6	Устный опрос, практическая работа, наблюдение
2	Динамические коллекции данных: Списки (List)	2	6	8	Устный опрос, практическая работа, наблюдение
3	Словари (Dictionary) и множества (HashSet)	2	4	6	Устный опрос, практическая работа, наблюдение
4	Введение в ООП: Классы, объекты и конструкторы	3	5	8	Устный опрос, практическая работа, наблюдение
5	Принципы ООП: Инкапсуляция и свойства	2	4	6	Устный опрос, практическая работа, наблюдение
6	Принципы ООП: Наследование и полиморфизм	3	5	8	Самостоятельная работа
7	Абстрактные классы и интерфейсы	2	4	6	Устный опрос, практическая работа, наблюдение
8	Работа с файловой системой и потоками данных	2	4	6	Устный опрос, практическая работа, наблюдение
9	Основы LINQ	2	4	6	Устный опрос, практическая работа, наблюдение
10	Разработка приложений с графическим интерфейсом	2	4	6	Устный опрос, практическая

					работа, наблюдение
11	Итоговое проектное программирование и защита	1	5	6	Самостоятельная работа
	Итого	23	49	72	

Содержание учебного плана.

Раздел 1 Повторение стартового уровня и углубление в методы.

Тема 1.1 – 1.3 Повторение стартового уровня и углубление в методы.

Теория. Экспресс-повторение синтаксиса (циклы, массивы). Углубленная работа с методами: передача параметров по значению и по ссылке (модификаторы ref, out). Перегрузка методов (Method Overloading). Опциональные параметры.

Повторение техники безопасности.

Практика. Входное тестирование. Написание методов с возвратом нескольких значений через out-параметры. Создание перегруженных функций для обработки разных типов данных.

Контроль. Устный опрос, практическая работа, наблюдение.

Раздел 2 Динамические коллекции данных: Списки (List).

Тема 2.1 – 2.4 Динамические коллекции данных: Списки (List).

Теория. Ограничения стандартных массивов. Понятие динамического списка List<T>. Обобщенные типы (Generics). Основные операции со списками: добавление (Add, AddRange), удаление (Remove, RemoveAt), вставка (Insert), поиск элементов, сортировка списка.

Практика. Создание динамического электронного журнала. Реализация программы для управления списком задач (добавление, удаление, редактирование, сортировка по приоритету) без ограничений по размеру.

Контроль. Устный опрос, практическая работа, наблюдение.

Раздел 3 Словари (Dictionary) и множества (HashSet).

Тема 3.1 – 3.3. Словари (Dictionary) и множества (HashSet).

Теория. Структура данных «Ключ-Значение». Класс Dictionary<TKey, TValue> и его эффективность. Проверка существования ключа (ContainsKey). Коллекция уникальных элементов HashSet<T>. Перебор ассоциативных массивов.

Практика. Создание телефонного справочника или англо-русского словаря. Решение задач на подсчет частоты встречаемости слов в тексте. Очистка списка от дубликатов с помощью множеств.

Контроль. Устный опрос, практическая работа, наблюдение.

Раздел 4 Введение в ООП: Классы, объекты и конструкторы.

Тема 4.1 – 4.4 Введение в ООП: Классы, объекты и конструкторы.

Теория. Философия объектно-ориентированного программирования. Различие между типом значения (Value Type) и ссылочным типом

(Reference Type). Анатомия класса. Модификаторы доступа public и private. Понятие конструктора класса. Конструктор по умолчанию и параметризованные конструкторы. Ключевое слово this.

Практика. Проектирование класса «Персонаж игры» или «Книга». Создание объектов с автоматической инициализацией через конструкторы. Управление базовым состоянием объектов.

Контроль. Устный опрос, практическая работа, наблюдение.

Раздел 5 Принципы ООП: Инкапсуляция и свойства.

Тема 5.1 – 5.3 Принципы ООП: Инкапсуляция и свойства.

Теория. Сущность инкапсуляции как сокрытия внутренней реализации. Защита данных от некорректного изменения. Механизм свойств (Properties) в C#: геттеры (get) и сеттеры (set). Автоматические свойства. Реализация логики валидации данных внутри свойств.

Практика. Модернизация классов: запрет на установку отрицательного возраста, некорректной цены или пустых строк. Реализация защищенного класса «Банковский счет».

Контроль. Устный опрос, практическая работа, наблюдение.

Раздел 6 Принципы ООП: Наследование и полиморфизм.

Тема 6.1 – 6.4 Принципы ООП: Наследование и полиморфизм.

Теория. Концепция наследования: базовые (родительские) и производные (дочерние) классы. Ключевое слово : для наследования. Передача параметров в базовый конструктор (base). Полиморфизм. Виртуальные методы (virtual) и их переопределение (override). Ключевое слово new для сокрытия методов.

Практика. Создание иерархии классов для компьютерной игры (базовый класс Воин → дочерние Маг, Лучник). Переопределение методов атаки и защиты для каждого подкласса, демонстрация работы полиморфного списка.

Контроль. Самостоятельная работа.

Раздел 7 Абстрактные классы и интерфейсы.

Тема 7. Абстрактные классы и интерфейсы.

Теория. Понятие абстракции. Абстрактные классы (abstract) и абстрактные методы без реализации. Понятие интерфейса (interface) как контракта поведения. Отличия интерфейсов от абстрактных классов. Множественное наследование интерфейсов в C#.

Практика. Проектирование системы классов «Геометрические фигуры» с абстрактным методом подсчета площади. Реализация интерфейсов для масштабирования или отрисовки объектов.

Контроль. Устный опрос, практическая работа, наблюдение.

Раздел 8 Работа с файловой системой и потоками данных.

Тема 8.1 – 8.3 Работа с файловой системой и потоками данных.

Теория. Пространство имен System.IO. Работа с файлами и папками (File, Directory). Чтение и запись текстовых файлов при помощи StreamReader и StreamWriter. Автоматическое управление ресурсами: конструкция using. Основы сериализации данных (сохранение объектов в текстовый формат).

Практика. Написание программы для сохранения истории игровых

сессий или базы данных пользователей в текстовый файл. Считывание конфигурационных файлов при запуске программы.

Контроль. Устный опрос, практическая работа, наблюдение.

Раздел 9 Основы LINQ.

Тема 9.1 – 9.3 Основы LINQ.

Теория. Что такое LINQ и зачем он нужен. Пространство имен System.Linq. Синтаксис запросов (Query Syntax) и синтаксис методов (Method Syntax / лямбда-выражения). Основные операторы фильтрации (Where), проекции (Select), сортировки (OrderBy, OrderByDescending), агрегации (Sum, Min, Max).

Практика. Быстрая фильтрация больших коллекций объектов (например, выборка из базы данных студентов только отличников старше 18 лет, отсортированных по алфавиту) в одну строчку кода.

Контроль. Устный опрос, практическая работа, наблюдение.

Раздел 10 Разработка приложений с графическим интерфейсом.

Тема 10.1 – 10.3 Разработка приложений с графическим интерфейсом.

Теория. Основы событийно-ориентированного программирования. Знакомство с фреймворком для создания оконных приложений (Windows Forms или кроссплатформенный Avalonia/WPF). Конструктор интерфейсов. Основные компоненты: формы, кнопки (Button), текстовые поля (TextBox), метки (Label), списки (ListBox). Обработчики событий (клик по кнопке, изменение текста).

Практика. Перенос старых консольных алгоритмов в графическое окно. Создание полноценного оконного калькулятора, телефонной книги с кнопками или мини-игры с графическим окном.

Контроль. Устный опрос, практическая работа, наблюдение.

Раздел 11 Итоговое проектное программирование и защита.

Тема 11.1 – 11.3 Итоговое проектное программирование и защита.

Теория. Паттерны проектирования программных систем. Архитектурный подход к созданию сложных приложений. Правила тестирования объектно-ориентированного кода. Подготовка проектной документации и презентации кроссплатформенного приложения.

Практика. Самостоятельная разработка сложного практического проекта с применением ООП, файлов и графического (или продвинутого текстового) интерфейса. Примеры тем: информационная система учета товаров на складе, настольная игра (шашки/крестики-нолики) с GUI, органайзер со сбором данных и сохранением в файлы. Публичная презентация и защита работы.

Контроль. Самостоятельная работа.

1.4 Планируемые результаты

Личностные результаты:

- ✓ формирование ответственного отношения к учению, готовности и способности, обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию;
- ✓ формирование целостного мировоззрения, соответствующего современному уровню развития науки и общественной практики;
- ✓ развитие осознанного и ответственного отношения к собственным поступкам при работе с графической информацией;
- ✓ развитие критического мышления при работе с интернет-ресурсами;
- ✓ формирование коммуникативной компетентности в процессе образовательной, учебно-исследовательской, творческой и других видов деятельности.

Метапредметные результаты:

- ✓ умение работать с информацией: находить с применением правил поиска в компьютерных сетях, оценивать и использовать информацию из различных источников при выполнении заданий и проектов по различным темам;
- ✓ умение определять результат выполнения алгоритма при заданных исходных данных, узнавать изученные алгоритмы обработки чисел и числовых последовательностей, создавать на их основе программы анализа данных, читать и понимать программы, написанные на языке программирования С#;
- ✓ умение выполнять пошагово алгоритмы управления исполнителями и анализа числовых и текстовых данных;
- ✓ разрабатывать и использовать компьютерно-математические модели, оценивать числовые параметры моделируемых объектов и процессов, интерпретировать результаты, получаемые в ходе моделирования реальных процессов, анализировать готовые модели на предмет соответствия реальному объекту или процессу;
- ✓ самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности;
- ✓ умение самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата;
- ✓ умение критически оценивать правильность решения учебно-исследовательской задачи;
- ✓ умение корректировать свои действия, вносить изменения в программу и отлаживать её в соответствии с изменяющимися условиями;
- ✓ умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками в процессе проектной и учебно-исследовательской деятельности;

Предметные результаты:

- ✓ обладать навыками исследовательской, проектной и социальной деятельности, строить логическое доказательство;

✓ разрабатывать эффективные алгоритмы и программы на основе изучения языка программирования C#;

✓ использовать специальные средства и библиотеки языка C#;

✓ работать в интегрированной среде разработки на языке программирования C#;

✓ умение определять понятия, создавать обобщения, устанавливать аналогии, классифицировать, самостоятельно выбирать основания и критерии для классификации, устанавливать причинно-следственные связи, строить логическое рассуждение, умозаключение (индуктивное, дедуктивное и по аналогии) и делать выводы;

Основным результатом обучения является достижение высокой информационно-коммуникационной компетентности обучающегося в области программирования на языке C# и проектирования программного обеспечения на платформе .NET.

2 Комплекс организационно-педагогических условий

2.1 Календарный учебный график

Количество учебных недель: 18

Количество учебных дней: 36

Даты начала и окончания учебного периода:

№ п/п	Тема занятия	Кол-во часов	Форма занятия	Форма контроля	Дата проведения занятия		Причина изменения даты
					планируемая	фактическая	
1.	Повторение стартового уровня и углубление в методы	2	Комбинированное занятие	Устный опрос, практическая работа, наблюдение.			
2.	Повторение стартового уровня и углубление в методы	2	Комбинированное занятие	Практическая работа, наблюдение.			
3.	Повторение стартового уровня и углубление в методы	2	Комбинированное занятие	Практическая работа, наблюдение.			
4.	Динамические коллекции данных: Списки (List)	2	Комбинированное занятие	Практическая работа, наблюдение.			
5.	Динамические коллекции данных: Списки (List)	2	Комбинированное занятие	Практическая работа, наблюдение.			
6.	Динамические коллекции данных: Списки (List)	2	Комбинированное занятие	Практическая работа, наблюдение.			
7.	Динамические коллекции данных: Списки (List)	2	Комбинированное занятие	Практическая работа, наблюдение.			
8.	Словари (Dictionary) и множества (HashSet)	2	Комбинированное занятие	Практическая работа, наблюдение.			
9.	Словари (Dictionary) и множества (HashSet)	2	Комбинированное занятие	Практическая работа, наблюдение.			
10.	Словари (Dictionary) и множества (HashSet)	2	Комбинированное занятие	Практическая работа, наблюдение.			

11.	Введение в ООП: Классы, объекты и конструкторы	2	Комбинированное занятие	Практическая работа, наблюдение.			
12.	Введение в ООП: Классы, объекты и конструкторы	2	Комбинированное занятие	Практическая работа, наблюдение.			
13.	Введение в ООП: Классы, объекты и конструкторы	2	Комбинированное занятие	Практическая работа, наблюдение.			
14.	Введение в ООП: Классы, объекты и конструкторы	2	Комбинированное занятие	Практическая работа, наблюдение.			
15.	Принципы ООП: Инкапсуляция и свойства	2	Комбинированное занятие	Практическая работа, наблюдение.			
16.	Принципы ООП: Инкапсуляция и свойства	2	Комбинированное занятие	Практическая работа, наблюдение.			
17.	Принципы ООП: Инкапсуляция и свойства	2	Комбинированное занятие	Практическая работа, наблюдение.			
18.	Принципы ООП: Наследование и полиморфизм	2	Комбинированное занятие	Практическая работа, наблюдение.			
19.	Принципы ООП: Наследование и полиморфизм	2	Комбинированное занятие	Практическая работа, наблюдение.			
20.	Принципы ООП: Наследование и полиморфизм	2	Комбинированное занятие	Практическая работа, наблюдение.			
21.	Принципы ООП: Наследование и полиморфизм	2	Комбинированное занятие	Практическая работа, наблюдение.			
22.	Абстрактные классы и интерфейсы	2	Комбинированное занятие	Практическая работа, наблюдение.			
23.	Абстрактные классы и интерфейсы	2	Комбинированное занятие	Практическая работа, наблюдение.			

24.	Абстрактные классы и интерфейсы	2	Комбинированное занятие	Практическая работа, наблюдение.			
25.	Работа с файловой системой и потоками данных	2	Комбинированное занятие	Практическая работа, наблюдение.			
26.	Работа с файловой системой и потоками данных	2	Комбинированное занятие	Практическая работа, наблюдение.			
27.	Работа с файловой системой и потоками данных	2	Комбинированное занятие	Практическая работа, наблюдение.			
28.	Основы LINQ	2	Комбинированное занятие	Практическая работа, наблюдение.			
29.	Основы LINQ	2	Комбинированное занятие	Практическая работа, наблюдение			
30.	Основы LINQ	2	Комбинированное занятие	Практическая работа, наблюдение			
31.	Разработка приложений с графическим интерфейсом	2	Комбинированное занятие	Практическая работа, наблюдение			
32.	Разработка приложений с графическим интерфейсом	2	Комбинированное занятие	Практическая работа, наблюдение			
33.	Разработка приложений с графическим интерфейсом	2	Комбинированное занятие	Практическая работа, наблюдение			
34.	Итоговое проектное программирование и защита	2	Комбинированное занятие	Самостоятельная работа			
35.	Итоговое проектное программирование и защита	2	Комбинированное занятие	Самостоятельная работа			
36.	Итоговое проектное программирование и защита	2	Комбинированное занятие	Самостоятельная работа			
	Итого	72					

2.2 Условия реализации программы

Материально-техническое обеспечение

Объединение «Программирование на C#» располагается в учебном кабинете. Кабинет обеспечен соответствующей мебелью: рабочими столами, стульями, компьютерами, программным обеспечением, выходом в интернет, мультимедийной доской, столом для руководителя.

Группа обучающихся состоит не более чем из 12 человек.

Рабочее место оснащено столом, стульями, персональным компьютером или ноутбуком, компьютерной мышью, программным обеспечением

Методическое обеспечение

Методическое обеспечение программы включает в себя: дидактический материал – презентации, видеозаписи, таблицы, схемы, плакаты, мультимедийные материалы.

Дидактический и раздаточный материал создается, подбирается и систематизируется в соответствии с учебно-тематическим планом (по каждой теме), возрастными и психологическими особенностями обучающихся, уровнем их развития и способностей.

К работе обучающиеся приступают после проведения соответствующего инструктажа по правилам техники безопасной работы, объявления темы занятия, плана работы. Новую тему педагог объясняет с применением технологий мультимедиа.

Методические материалы

Для организации занятия в образовательном процессе используются:

- словесный метод (устное изложение, опрос и др.);
- наглядный метод (показ видеоматериалов, иллюстраций, мастер классы);
- практический метод (решение поставленных задач).

Для диагностики успешности освоения учебной программы используются:

- метод наблюдения;
- метод анализа продуктов образовательной деятельности обучающегося.

Информационное обеспечение

Персональный компьютер со специальной программой с процессором не ниже 1,2 ГГц и 256 Мб оперативной памяти с установленной операционной системой Linux или Windows (11 шт.) и выходом в сеть Интернет.

- Инструкции по установке .NET, карточки-задания с математическими формулами, шаблоны Console App.

- Архитектурные карты классов, шаблоны ООП-проектов, чек-листы «Clean Code».

Кадровое обеспечение: В реализации программы занят педагог дополнительного образования Мердеев Никита Сергеевич.

2.3 Формы аттестации

Формы проведения аттестации: выполнение итогового контроля, участие в олимпиадах и соревнованиях различного уровня.

Виды контроля:

- Предварительный (входная диагностика) – проводится в виде тестирования при зачислении на программу.
- Текущий контроль – проводится на каждом занятии в виде тестирования, дискуссии, практической работы над заданиями в определенной тематике.
- Тематический контроль – проводится по мере освоения каждого раздела. Включает фронтальный устный опрос, а также различные виды деятельности при индивидуальном и групповом опросе.
- Итоговый контроль: заключительная проверка знаний, умений, навыков в конце учебного курса.

2.4 Критерии оценки

Формы проведения аттестации: выполнение итогового контроля, участие в соревнованиях различного уровня.

Виды контроля:

- Предварительный (входная диагностика) – проводится в виде тестирования при зачислении на программу.
- Текущий контроль – проводится на каждом занятии в виде тестирования, дискуссии, практической работы над заданиями в определенной тематике.
- Тематический контроль – проводится по мере освоения каждого раздела. Включает фронтальный устный опрос, а также различные виды деятельности при индивидуальном и групповом опросе.
- Итоговый контроль: заключительная проверка знаний, умений, навыков в конце учебного курса.

2.4 Критерии оценки

Оценочные материалы выражаются в успешной сдаче текущих и итоговых тестов по разделам программы, в участии в олимпиадах и конкурсах.

Формы проведения итогов:

- Итоговая контрольная работа.
- Соревнования различного уровня.

Способы определения результативности

В образовательном процессе для диагностики успешности освоения учебной программы используются:

- метод наблюдения;
- метод анализа продуктов образовательной деятельности обучающегося.

Формы диагностики:

1. Промежуточная диагностика, уровня освоения материала – решение поставленных задач по итогам изучения каждого раздела.
2. Итоговая диагностика. Предметом оценки служат умения и знания, направлены на формирование общих и профессиональных компетенций.

Оперативный контроль осуществляется по итогам каждого занятия в форме тестирования.

Задачи текущего контроля:

- повышение мотивации обучающихся к регулярной учебной работе;
- развитие навыков самостоятельной работы;
- обеспечение обратной связи между обучающимися и преподавателем, на основании которой устанавливается, как обучающиеся воспринимают и усваивают учебный материал;
- дифференциация итоговой оценки знаний

Формы проведения итогов:

Итоговая контрольная работа.

- Соревнования различного уровня.

Способы определения результативности:

В образовательном процессе для диагностики успешности освоения учебной программы используются:

- метод наблюдения;
- метод анализа продуктов образовательной деятельности обучающегося.

Формы диагностики:

1. Промежуточная диагностика, уровня освоения материала – решение поставленных задач по итогам изучения каждого раздела.

2. Итоговая диагностика.

Предметом оценки служат умения и знания, направлены на формирование общих и профессиональных компетенций.

Задачи текущего контроля:

- повышение мотивации обучающихся к регулярной учебной работе;
- развитие навыков самостоятельной работы;
- обеспечение обратной связи между обучающимися и преподавателем, на основании которой устанавливается, как обучающиеся воспринимают и усваивают учебный материал;
- дифференциация итоговой оценки знаний.

2.5 Воспитательный компонент

Цель воспитательной работы

Создать условия для формирования у обучающихся ответственного отношения к разработке программного продукта.

Задачи воспитательной работы

1. Познакомить обучающихся с основными принципами и методами создания интерактивных приложений, развить интерес к творческой и технической деятельности.

2. Познакомить обучающихся с основными направлениями в сфере создания интерактивных приложений, помочь принять решение в дальнейшем выборе профессии.

Приоритетные направления воспитательной деятельности:

Создание благоприятной образовательной среды, способствующей развитию интереса обучающихся к информатике, углублению знаний о разработке программных продуктов и для подготовки к дальнейшему изучению информатики на более продвинутом уровне.

Формы воспитательной работы:

– беседа, лекция, дискуссия.

Методы воспитательной работ:

– беседа, дискуссия, создание воспитывающих ситуаций, соревнование, игра, выставки, анализ результатов деятельности.

Планируемые результаты воспитательной работы:

1. Формирование интереса к разработке программного обеспечения:
 - интерес к изучению методов создания программных продуктов.
 - развивается любознательность и желание узнавать новое о сфере информационных технологий.
2. Развитие навыков самостоятельной работы и исследования:
 - умение самостоятельно реализовывать задуманные идеи в виде программного продукта.
 - формирование навыков систематизации информации и логического мышления.
3. Развитие навыков коммуникации и сотрудничества:
 - умение работать в группе, обмениваться информацией и идеями.
4. Поддержка развития творческого мышления и критического мышления:
 - умение предлагать альтернативные решения, анализировать информацию и высказывать свои собственные выводы.
 - способность к критическому мышлению и аргументации.

Эти результаты помогут учащимся не только углубить знания в области разработки программного обеспечения, но также развить навыки, которые будут полезны им в учебе, повседневной жизни и будущей профессиональной деятельности.

Календарный план воспитательной работы

№ п/п	Тема занятия	Воспитательный компонент
1.	Повторение стартового уровня и углубление в методы. Динамические коллекции данных: Списки (List). Словари (Dictionary) и множества (HashSet).	Цель: формирование ценностного отношения к науке, развитие познавательного интереса, личностных качеств и общекультурных компетенций через «Программирование на C#». Задачи: 1. Стимулирование познавательной активности обучающихся через интересные и понятные задачи и эксперименты. 2. Развитие критического мышления и способности анализировать информацию. 3. Поддержка творческого потенциала через решение нетривиальных задач и проектную деятельность. 4. Воспитание ответственности за результаты своей учебной деятельности и за окружающий мир.

Введение в ООП: Классы, объекты и конструкторы. Принципы ООП: Инкапсуляция и свойства. Принципы ООП: Наследование и полиморфизм. Абстрактные классы и интерфейсы Работа с файловой системой и потоками данных.	Цель: формирование ценностного отношения к науке, развитие познавательного интереса, личностных качеств и общекультурных компетенций через «Программирование на C#» Задачи: 1. Стимулирование познавательной активности обучающихся через интересные и понятные задачи и эксперименты. 2. Развитие критического мышления и способности анализировать информацию. 3. Поддержка творческого потенциала через решение нетривиальных задач и проектную деятельность. 4. Воспитание ответственности за результаты своей учебной деятельности и за окружающий мир.
Основы LINQ. Разработка приложений с графическим интерфейсом. Итоговое проектное программирование и защита.	Цель: формирование ценностного отношения к науке, развитие познавательного интереса, личностных качеств и общекультурных компетенций через «Программирование на C#». Задачи: 1. Стимулирование познавательной активности обучающихся через интересные и понятные задачи и эксперименты. 2. Развитие критического мышления и способности анализировать информацию. 3. Поддержка творческого потенциала через решение нетривиальных задач и проектную деятельность. 4. Воспитание ответственности за результаты своей учебной деятельности и за окружающий мир.

Воспитательный компонент программы направлен на развитие обучающихся не только в области программирования на C#, но также способствует формированию логического мышления, аналитического мышления, умения работать в команде, поощряет творческое мышление и интерес к программированию.

Список литературы

Для педагога:

1. Поляков, К.Ю. Информатика. Углублённый уровень. Учебник для 10 класса в 2 частях / К.Ю. Поляков, Е.А. Еремин. — Москва: БИНОМ. Лаборатория знаний, 2014.
2. Задачи по программированию / под ред. С. М. Окулова. — Москва: БИНОМ. Лаборатория знаний, 2006.

Для обучающихся:

1. Васильев, А.Н. Программирование на С# для начинающих. Основные навыки / А.Н. Васильев. — Москва: Эксмо, 2022.
2. Информатика и ИКТ. Задачник-практикум в 2 частях / под ред. И.Г. Семакина, Е.К. Хеннера. — Москва: БИНОМ. Лаборатория знаний, 2014.
3. Microsoft Learn: материалы, интерактивные задания и руководства. — Официальный портал. — URL: <https://learn.microsoft.com/ru-ru/> (дата обращения: 08.08.2026).
4. Metanit.com: руководство по языку программирования С#. — Веб-сайт. — URL: <https://metanit.com/sharp/> (дата обращения: 08.08.2026).

Для родителей:

1. С.М. Окулов. Основы программирования. М.: Бином. Лаборатория знаний, 2012.
2. А.Н. Васильев. Программирование на С# для начинающих. Основные навыки. М.: Эксмо, 2022.

Приложение

Методические рекомендации по проведению занятий с применением оборудования.

Использование интерактивной доски на занятиях

Интерактивная доска – сенсорный экран, подсоединенный к компьютеру, изображение с которого передает на доску проектор. Вместе они являются интерактивным комплексом. В программном обеспечении любой ИД имеются различные функции, с помощью которых можно продуктивно работать с любыми объектами на доске: перемещать, группировать, скрывать за шторкой, делать съемку экрана, видеозапись урока и многое другое.

Виды и типы интерактивных досок.

Сегодня в России в основном продаются модели досок следующих производителей:

1. Smart (доски SmartBoard) - Канада
2. Hitachi (доски StarBoard и FX-DUO) - Япония
3. Panasonic (доски Panaboard) - Япония
4. PolyVision (доски Walk-and-Talk и Webster) - США
5. Sahara Interactive (доски Communicator77 и Penbord) - Бельгия
6. Triumph (доски Triumphboard) – Чехия
7. QOMO HiteVision (доски QWB200 и QWB300) – США
8. InterWrite (доски InterWriteBoard) - США
9. Prometeam LTD (доски AktivBoard) – Англия

Интерактивные доски делятся на два класса в зависимости от расположения проектора: с фронтальной и обратной проекцией.

Доски с фронтальной проекцией распространены наиболее широко, хотя и обладают очевидным недостатком: докладчик может загораживать собой часть изображения. Чтобы этого не было, проектор подвешивают под потолком как можно ближе к доске, объектив наклоняют вниз, а возникающие трапециевидные искажения компенсируют с помощью системы цифровой коррекции.

Доски с обратной проекцией, где проектор находится позади экрана, существенно дороже и занимают в аудитории больше места, чем доски с прямой проекцией. Поскольку экран работает на просвет, возможны проблемы с видимостью изображения под большими углами.

Проектор и компьютер для работы с интерактивной доской могут быть практически любыми (например, те, что уже есть в школе) – специальных требований к ним для работы с доской не предъявляется.

Преимущества и недостатки интерактивных досок

Преимущества для педагогов:

1. Работа с доской в полной мере позволяет реализовать принцип наглядности;
2. Поощряет импровизацию и гибкость, позволяя рисовать и делать записи поверх любых приложений;

Продолжение приложения

3. Позволяет сохранять и распечатывать изображения на доске, включая любые записи, сделанные во время занятия, не затрачивая при этом много времени и сил и упрощая проверку усвоенного материала;

4. Разнообразие цветов, доступных на интерактивной доске, позволяет выделять важные области и привлекать внимание к ней, связывать общие идеи или показывать их отличие и демонстрировать ход размышления.

5. Возможность вынести ключи решений на доску. Их можно временно спрятать за каким-либо объектом на рабочем слайде или поместить на следующий слайд.

6. Значительно экономит время при подготовке к урокам за счет встроенных функций ПО;

7. Удобна при работе в большой аудитории;

8. Вдохновляет преподавателей на поиск новых подходов к обучению, стимулирует профессиональный рост.

Преимущества для учащихся:

1. Делает занятия интересными и является мощным инструментом для развития мотивации к обучению;

2. Предоставляет больше возможностей для активного участия в учебном процессе;

3. Облегчает понимание сложного материала в результате более ясного, эффективного и динамичного его представления;

4. Способствует развитию творчества и самостоятельности учащихся;

5. Возможность вырезать и стирать объекты с экрана, копировать и вставлять их, отменять или возвращать действия придает учащимся больше уверенности: они знают, что всегда могут вернуться на шаг назад, что-то изменить или исправить.

Среди недостатков или возможных трудностей, можно отметить:

1. Недостаточно приспособленное для педагогов программное обеспечение досок;

2. Невозможность обновления существующего ПО из-за отсутствия выхода в Интернет;

3. Наличие нескольких разных типов интерактивных досок, программы которых несовместимы;

4. Наличие чисто технических проблем: при скачках напряжения или отказе доски по неизвестной причине далеко не все учителя сумеют найти верное решение проблемы;

5. Интерактивные доски намного дороже, чем стандартные доски или же проектор с экраном.

6. Поверхность интерактивных досок может повредиться, замена поврежденной поверхности также очень дорогостоящая услуга (в России такой ремонт, возможно, будет равноценен покупке новой доски).

Окончание приложения

7. Если к интерактивной доске разрешен удаленный доступ, то некоторые пользователи могут передать на экран нежелательное сообщение или рисунок.

8. Необходимость временного ограничения работы с интерактивной доской на уроке из-за необходимости соблюдать санитарные нормы.

Сан Пин по использованию интерактивной доски

Допускается оборудование учебных помещений и кабинетов интерактивными досками, отвечающими гигиеническим требованиям. При использовании интерактивной доски и проекционного экрана необходимо обеспечить равномерное ее освещение и отсутствие световых пятен повышенной яркости.

(В данном разделе использованы материалы из интернет-ресурса: [Интерактивная доска для начинающих | Статья на тему: | Образовательная социальная сеть \(nsportal.ru\)](#))