

Муниципальное общеобразовательное учреждение гимназия села Малая Пурга

РАССМОТРЕНА
на заседании Методического совета
Протокол № 1 от 28.08.2024 г.

УТВЕРЖДЕНА
Приказ № 177 от 29.08.2024г.

СОГЛАСОВАНА

Зам.директора по ВР

С.Н.Зайцева

Дополнительная общеобразовательная общеразвивающая программа
естественнонаучной направленности
«Программирование на языке С++»
**(с использованием цифрового и аналогового оборудования
естественнонаучного и технологического профилей центра «Точки роста»)**

Возраст детей 12-14 лет
Срок реализации программы 1 года

Автор-составитель:
Медведева Лидия Васильевна,
учитель информатики

2024 г

Пояснительная записка

Программа составлена в соответствии с федеральными нормативными правовыми актами в области дополнительного образования, государственными требованиями к образовательным программам системы дополнительного образования детей, а также локальными нормативными правовыми актами организации:

Дополнительная общеобразовательная общеразвивающая программа «Программирование на языке С» разработана на основании:

- Федеральным законом РФ от 29.12.2012 г. №273-ФЗ «Об образовании в Российской Федерации»;
- Федеральным законом РФ от 27.07.2006 г. № 152-ФЗ «О персональных данных»;
- Постановлением Главного государственного санитарного врача РФ от 28.09.2020 г. № 28 «Об утверждении Санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;
- Приказом Министерства просвещения РФ от 27 июля 2022 года № 629 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».
- Уставом Муниципального автономного вечернего (сменного) общеобразовательного учреждения «Малопургинского Центра образования»
- Положением о дополнительной общеобразовательной общеразвивающей программе Муниципального автономного вечернего (сменного) общеобразовательного учреждения «Малопургинский Центр образования»
- Локальными актами МАВОУ «Малопургинского ЦО».

Направленность (профиль) программы: техническая

Уровень программы: вводный, продвинутый.

Новизна программы состоит в том, что реализуется возможность обучению навыкам работы в группе, создания коллективных проектов, чего практически невозможно достичь при изучении традиционных языков Бейсик, Паскаль. Возможность увидеть результаты своего труда в Интернет также стимулирует интерес детей получить представление об алгоритмах и исполнителях, основных алгоритмических конструкциях языков программирования.

Актуальность программы:

В настоящее время процесс информатизации проявляется во всех сферах человеческой деятельности. Использование современных информационных технологий является необходимым условием успешного развития как отдельных отраслей, так и государства в целом.

Педагогическая целесообразность. Программа отвечает потребностям детей в созидательной и продуктивной деятельности, направлена на развитие цифровой грамотности.

Так, в соответствии с Законом Удмуртской Республики «О стратегии социально-экономического развития Удмуртской Республики до 2025 года» одними из наиболее значимых факторов и тенденций, оказывающих влияние на развитие отрасли радиоэлектроники и электроники станут автоматизация и цифровизация производственной области, а необходимость создавать компетенции в сфере ИТ решений и вести собственные разработки в области промышленного интернета, искусственного интеллекта, автоматизированного транспорта и телемедицины являются актуальными задачами развития отрасли в Удмуртии.

В связи с этим актуальной задачей является подготовка специалистов сферы информационных технологий, программирования, в соответствии с профессиональными требованиями динамично развивающихся отраслей.

Изучение основных принципов программирования невозможно без регулярной практики написания программ на каком-либо языке. Язык С++ является очень востребованным языком; он отлично подходит для знакомства с различными современными парадигмами программирования и активно применяется в самых разных областях от разработки веб-приложений до машинного обучения.

Научившись программировать на языке С++, обучающиеся получают мощный и удобный инструмент для решения как учебных, так и прикладных задач. Вместе с тем чистота и ясность его конструкций позволит обучающимся потом с лёгкостью выучить любой другой язык программирования.

Знания и умения, приобретённые в результате освоения программы, могут быть использованы обучающимися при участии в олимпиадах по программированию, при решении задач по физике, химии, биологии, лингвистике и другим наукам, а также они являются фундаментом для дальнейшего совершенствования мастерства программирования.

Цель:

Развитие интереса у обучающихся к информационным и телекоммуникационным технологиям; реализация творческих идей в области программирования и электроники в виде проектов различного уровня сложности.

Задачи программы:

Образовательные:

- сформировать представление об основах программирования в среде С++;
- сформировать навыки грамотной работы в системе программирования С++;
- ознакомить с базовыми понятиями теории алгоритмов при решении математических задач;
- обучить методам решения задач, реализуемым на языке С++;
- сформировать практические навыки решения прикладных задач;
- сформировать навыки поиска информации, работы с технической литературой.

Развивающие:

- развить навыки самостоятельного и творческого подхода к решению задач с помощью средств современной вычислительной техники;
- развить алгоритмическое и логическое мышление учащихся;
- развить творческие способности обучающихся, их потребность в самореализации;
- развить интеллектуальные и практические умения, самостоятельно приобретать и применять на практике полученные знания;

Воспитательные:

- воспитать уважительное отношение к преподавателям и сверстникам, культуру поведения во время занятий и совместной продуктивной деятельности;
- содействовать воспитанию устойчивого интереса к изучению программирования;
- содействовать воспитанию информационной культуры;
- формировать потребность в творческой деятельности, стремление к самовыражению через техническое творчество;
- содействовать воспитанию интереса профессиям, связанным с программированием.

Отличительные особенности

Занятия проводятся в группах учащихся одного возраста, являющихся основным составом объединения, а также индивидуально. Система работы включает в себя теоретические и практические занятия, ориентирована на большой объем практических творческих работ с использованием компьютера. Все образовательные модули предусматривают не только усвоение теоретических знаний, но и формирование деятельностно-практического опыта. Освоение материала в основном происходит в процессе практической творческой деятельности.

Методологической основой изучения является системно-деятельностный подход, который обеспечивает:

- формирование готовности обучающихся к саморазвитию и непрерывному образованию;
- проектирование и конструирование развивающей образовательной среды организации, осуществляющей образовательную деятельность;
- активную учебно-познавательную деятельность обучающихся;
- построение образовательной деятельности с учетом индивидуальных, возрастных, психологических, физиологических особенностей и здоровья обучающихся.

Условия реализации

Изучение программирования на языках высокого уровня в образовательных учреждениях среднего образования учащимися 12-14 лет и старше (начиная с 8 класса образовательных школ).

Занятия проводятся в группах, а также индивидуально.

Группы могут быть одно- или разновозрастными, смешанными или однополыми. Для более успешного усвоения программы количество обучающихся в группе от 10 до 15 человек.

Сроки реализации программы

Базисный учебный план отводит на изучение информатики 3 учебных часа в неделю (102 часа в год). В данной программе добавлены разделы, необходимые для успешного изучения алгоритмизации как начального этапа автоматизации производственных и информатизационных процессов, а также программирования на языке высокого уровня.

Режим занятий

Занятия проводятся 2 раза в неделю (по 1 часу и 2 часа). Продолжительность одного учебного часа 40 мин и перерыв между занятиями 10 мин.

Формы занятий

Система работы включает в себя теоретические и практические занятия, ориентирована на большой объем практических творческих работ с использованием компьютера. Все образовательные модули предусматривают не только усвоение теоретических знаний, но и формирование деятельностно - практического опыта. Освоение материала в основном происходит в процессе практической творческой деятельности. На занятиях применяются занимательные и доступные для понимания задания, и упражнения, задачи, вопросы и т.д.

Основная форма организации образовательного процесса дополнительного образования – учебное занятие. Учебные занятия могут быть в зависимости от поставленных задач иметь следующие формы:

1. Беседа, лекция
2. Деловая игра

3. Задание по образцу (с использованием инструкции)
4. Практическая (лабораторная) работа «Решение задач»
5. Комбинированные занятия
6. Проект.

Формы проведения занятий разнообразны - вводные, традиционные, практические, занятия ознакомления, повторения, обобщения и контроля полученных знаний, коллективные работы. Традиционны на занятиях рассказ и беседа не только познавательного характера, но и с воспитательной целью. Стимулируют интерес к обучению занятия в виде игр, викторин, кроссвордов, конкурсов, экскурсий

Для адаптации в современном информационном обществе важным фактором является формирование математического и алгоритмического стиля мышления, включающего индукцию и дедукцию, обобщение и конкретизацию, анализ и синтез, классификацию и систематизацию. Использование формальных языков позволяет развивать у учащихся грамотную устную и письменную речь.

Особенностью курса является его практическая направленность, которая служит успешному усвоению курса информатики.

Практическая значимость школьного курса программирования 8 класса состоит в том, что предметом его изучения являются количественные отношения и процессы реального мира, описанные математическими моделями в виде алгоритмов и программ на языке программирования высокого уровня. Основной целью является формирование абстрактного, логического и алгоритмического мышления.

Алгоритмические знания и умения необходимы для изучения других школьных предметов: математики, физики, химии и даже отдельных аспектов биологии.

УЧЕБНЫЙ ПЛАН

Наименование разделов (модулей)	Обязательная аудиторная учебная нагрузка, (часов)			Дистанционная/самостоятельная работа, часов	Использование оборудования «Точка роста»	Формы, виды контроля
	всего	в т.ч. практические	в т.ч. лабораторные			
Тема 1. История языка С++ и существующие стандарты. Поточковый ввод и вывод в языке С++. Новые элементы в С++ в представлении базовых типов данных и массивов и в работе с ними.	18	6	6	6	ноутбук	
Тема 2. Строки с завершающим нулем. Класс String: свойства и методы. Массивы строк.	18	6	6	6	ноутбук	Итоговая работа
Тема 3. Библиотека STL: принципы проектирования и функционирования. Контейнеры, итераторы, обобщенные алгоритмы. Последовательный контейнер вектор : свойства и методы.	18	6	6	6	ноутбук	
Тема 4. Быстрые методы и алгоритмы сортировки последовательностей с применением последовательных контейнеров.	6	2	2	2	ноутбук	
Тема 5. Двумерные и многомерные структуры. Решение задач.	12	4	4	4	ноутбук	Итоговая работа
Тема 6. Последовательные контейнеры список, стек и очередь: свойства и методы. Решение задач.	18	6	6	6	ноутбук	Итоговая работа
Тема 7. Алгоритмы двоичного поиска и их применение к решению задач.	12	4	4	4	ноутбук	
Всего:	102	34	34	34		

СОДЕРЖАНИЕ ЗАНЯТИЙ

Тема 1. История языка C++. Обеспечение совместимости языков C и C++ на основе стандартов C++ ISO (C++98/03 и C++11), поддержка идеи переносимого языка программирования. Препроцессор C++, файл iostream; имена заголовочных файлов, пространства имен. Вывод в C++ с помощью cout. Ввод информации с использованием cin.

Целочисленные типы short, int, long и long long. Типы без знаков. Тип char: символы и короткие целые числа. Тип bool. Квалификатор const. Правила инициализации массивов.

Двоичная система счисления и представление целых чисел в памяти ЭВМ.

Альтернативы массивам - шаблонный класс array (C++11). Сравнение массивов и объектов array.

Практическая часть:

Изучение основных приемов работы учителя и учащегося с системой автоматической проверки решений – сайт moodle.cs.istu.ru.

Создание проекта в среде программирования Visual Studio, выполнение нескольких заданий в рамках одного проекта.

Тема 2. Строки с завершающим нулем. Представление в памяти, инициализация, функции для строк. Введение в класс string. Инициализация строк в C++11, ввод и вывод. Присваивание, сравнение и вероятные ошибки. Выражения отношений. Сравнение строк в стиле C и строк класса string.

Функции для строк (string): вставка, замена, удаление, поиск. Массивы строк.

Практическая часть:

Объявления строк, ввод и вывод. Работа с библиотекой функций для обработки строк. Пошаговая отладка программ с функциями в среде программирования Visual Studio.

Тема 3. Отличие STL от других библиотек. Последовательные контейнеры: вектор. Инициализация, ввод и вывод. Итераторы прямого доступа.

Обобщенные алгоритмы STL. Неизменяющие и изменяющие алгоритмы над последовательностями (на примере вектора). Алгоритмы, связанные с сортировкой, и обобщенные числовые алгоритмы.

Практическая часть:

Объявление векторов, ввод и вывод, операции над элементами и с векторами в целом. Работа с библиотекой обобщенных алгоритмов STL.

Тема 4. Реализация и сравнение сортировок: пузырьковая, поразрядным группированием и быстросорт (quicksort).

Практическая часть:

Применение обобщенных алгоритмов сортировки из библиотеки STL.

Тема 5. Особенности применения контейнеров STL для хранения и обработки двумерных данных. Решение задач с обработкой двумерных числовых и символьных данных.

Практическая часть:

Объявление двумерных векторов, ввод и вывод, операции над элементами и с векторами в целом. Работа с библиотекой обобщенных алгоритмов STL.

Тема 6. Контейнеры последовательностей: список. Инициализация, ввод и вывод. Итераторы входные, выходные и однонаправленные. Контейнеры последовательностей: стек и очередь. Инициализация, ввод и вывод. Вычислительные задачи с обработкой данных в порядке поступления «первый пришел – первым обслужен» и в обратном порядке «последний пришел – первым обслужен».

Практическая часть:

Объявление очереди и стека, ввод и вывод, операции над элементами и с очередями и стеками в целом.

Тема 7. Задача поиска информации; линейный и логарифмический (двоичный) поиск: алгоритмы и сравнение. Задачи двоичного поиска по ответу.

Практическая часть:

Применение обобщенных алгоритмов двоичного поиска для обработки массивов и векторов из библиотеки STL. Решение задач.

Итоговый контроль. Количество часов – 1 час.

УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

Наименование раздела. Тема уроков	Кол. часов	№ урока	Характеристика основных видов деятельности ученика
I. История языка C++ и существующие стандарты. Потоковый ввод и вывод в языке C++	6		
Отличие объектно-ориентированного подхода разработки программ от процедурного.		1-2	<i>Знать.</i> Объектно-ориентированное программирование – подход для проектирования больших программных систем. Взаимосвязь понятий объекта и класса на примере объектов <code>cin</code> и <code>cout</code> , являющихся представителями классов <code>iostream</code> для потоков входной и выходной символьной информации. <i>Уметь.</i> Составлять операторы ввода и вывода числовой и символьной информации для решения простых линейных алгоритмов.
Стиль написания исходного кода и операторы в языке C++		3-4	<i>Знать.</i> Структуру программы на языке C++. Диапазон изменения и основные операции для целых чисел. Особенности представления и обработки символов в C++. Правила записи операторов присваивания, ввода и вывода. <i>Уметь.</i> Записать операторы ввода числовых и символьных значений с клавиатуры и сохранения их в переменных, операторы вывода символьных строк и значений переменных на экран монитора. Написать программу решения задачи.
Знакомство со средой программирования.		5-6	<i>Знать.</i> Последовательность запуска среды программирования, создания проекта, подключения и

Наименование раздела. Тема уроков	Кол. часов	№ урока	Характеристика основных видов деятельности ученика
Набор и запуск программ.			<p>исключения из проекта текстовых модулей с программами на C, запуска программ на выполнение.</p> <p><i>Уметь.</i> Создать проект в среде программирования, включить в него модуль с текстом программы, запустить программу и проанализировать результат. Зарегистрироваться в системе удаленной проверки задач, получить условие задачи, отправить решение на проверку, просмотреть результат.</p>
II. Циклы и тестовый ввод.	6		
Функции get(), eof() и fail() объекта cin и put() объекта cout для более точной работы с символами.		7-8	<p><i>Знать.</i> Правило «пропуска символов-разделителей» при чтении символов с помощью объекта cin. Перегрузка функций в C++ на примере cin.get(). Средства обнаружения завершения данных в cin: биты eofbit и failbit, функции eof() и fail().</p> <p><i>Уметь.</i> Составлять алгоритмы, завершающие работу при завершении данных во входном потоке.</p>
Циклы с пред- и постусловием при вводе символьной информации. Решение задач.		9-12	<p><i>Знать.</i> Новые возможности инициализации массивов в C++.</p> <p><i>Уметь.</i> Написать программу решения задачи обработки числовых и символьных последовательностей с применением циклов с пред- и постусловием.</p>
III. Системы счисления.	6		
Позиционная система счисления: алфавит, основание, представление целых чисел и преобразования		13-14	<p><i>Знать.</i> Основы двоичной и шестнадцатеричной систем счисления, алгоритмы перевода чисел между двоичной, десятичной и шестнадцатеричной системами счисления. Внутреннее представление целых чисел и символов в памяти компьютера.</p> <p><i>Уметь.</i> Выполнять преобразования целых чисел из внешнего представления (текстовой десятичной записи) во внутреннее и наоборот.</p>
Новые элементы в C++ в представлении целых чисел и в		15-18	<p><i>Знать.</i> Встроенные типы данных для работы с целыми числами, правила записи и операции с целыми в языке C++.</p>

Наименование раздела. Тема уроков	Кол. часов	№ урока	Характеристика основных видов деятельности ученика
работе с ними. Решение задач.			<i>Уметь.</i> Разработать и отладить алгоритм и программу преобразования целого числа из одной системы счисления в другую. Написать программу решения задачи, опирающейся на свойства целого и значения отдельных разрядов числа.
IV. Строки в стиле С.	6		
Строки с завершающим нулем. Представление в памяти, инициализация в С++11, ввод и вывод. Работа с указателями.		19- 20	<i>Знать.</i> Массив символов как хранилище (контейнер) строк в стиле С. Ввод и вывод строк. Понятие указателя и работа с ним. <i>Уметь.</i> Разработать и отладить программу обработки строк с применением «индуктивного» способа построения алгоритма.
Функции для строк. Решение задач.		21- 24	<i>Знать.</i> Назначение и параметры основных функций для обработки строк: длина строки, копирование, сравнение строк, поиск подстрок. <i>Уметь.</i> Разработать и отладить программу обработки строк с применением функций для строк.
V. Класс String.	6		
Введение в класс string. Инициализация строк в С++11, ввод и вывод. Присваивание, сравнение и вероятные ошибки. Основные функции.		25- 26	<i>Знать.</i> Правила инициализации объектов типа string, операции присваивания, сцепления, сравнения, ввода и вывода. Синтаксические правила для вызова функций-членов класса string. <i>Уметь.</i> Разработать и отладить программу обработки строк с применением функций для строк.
Функции вставки и замены для строк. Применение обобщенных алгоритмов к объектам string. Решение задач.		27- 30	<i>Знать.</i> Назначение и параметры функций-членов класса string: вставка и замена строк. Правила применения обобщенных алгоритмов (reverse, swap) для объектов string. <i>Уметь.</i> Разработать и отладить программу обработки строк с применением функций для объектов string.
VI. Массивы String.	6		

Наименование раздела. Тема уроков	Кол. часов	№ урока	Характеристика основных видов деятельности ученика
Массивы String.		31- 32	<i>Знать.</i> Правила объявления и инициализации массивов string, применения функций к элементам массивов. <i>Уметь.</i> Разработать и отладить программу обработки строк с применением функций для массивов string.
Понятие структуры. Массивы структур. Решение задач.		33- 35	<i>Знать.</i> Правила объявления и инициализации структур, работы с полями структуры; работы с массивами структур. <i>Уметь.</i> Разработать и отладить программу обработки символьной информации с применением структур и массивов структур.
Контрольная работа 1		36	Применять полученные знания и умения при решении примеров и задач.
VII. Векторы почти как массивы. Создание, методы.	6		
Отличие STL от других библиотек. Контейнеры последовательностей: вектор. Инициализация, ввод и вывод.		37- 38	<i>Знать.</i> Правила объявления и инициализации векторов, операции над ячейками вектора и векторами в целом. Функции-члены класса вектор push_back(), capacity(). <i>Уметь.</i> Разработать и отладить программу обработки последовательности чисел с хранением ее в контейнере вектор.
Функции-члены класса вектор clear(), swap(). Решение задач.		39- 42	<i>Знать.</i> Назначение и параметры функций-членов класса вектор clear(), swap(). <i>Уметь.</i> Разработать и отладить программу обработки последовательности чисел с хранением ее в контейнере вектор, применяя функции-члены класса вектор.
VIII. Простые итераторы, алгоритмы.	6		
Итераторы – связующий элемент между контейнерами и алгоритмами:		43- 44	<i>Знать.</i> Назначение и классификацию итераторов, правила их объявления, диапазоны и допустимые

Наименование раздела. Тема уроков	Кол. часов	№ урока	Характеристика основных видов деятельности ученика
определения, классификация. Операции над итераторами.			<p>операции. Функции класса вектор, возвращающие значения итераторов: <code>begin()</code>, <code>end()</code>.</p> <p><i>Уметь.</i> Разработать и отладить программу обработки последовательности чисел с хранением ее в контейнере вектор, выполняя доступ к элементам вектора с использованием итераторов.</p>
Обобщенные алгоритмы: определения, классификация, примеры. Решение задач.		45-48	<p><i>Знать.</i> Классификацию алгоритмов. Назначение и основные параметры обобщенных алгоритмов, правила использования в C++ (<code>adjacent_difference()</code>, <code>copy()</code>, <code>transform()</code>).</p> <p><i>Уметь.</i> Разработать и отладить программу обработки последовательности чисел с хранением ее в контейнере вектор, выполняя доступ к элементам вектора с использованием итераторов и обработку с использованием обобщенных алгоритмов.</p>
IX. Входные, выходные, потоковые итераторы.	6		
Обратный итератор. Выходные и потоковые итераторы.		49-50	<p><i>Знать.</i> Назначение, объявление и операции с выходными итераторами. Особенности выходных потоковых итераторов.</p> <p><i>Уметь.</i> Разработать и отладить программу обработки последовательности чисел с хранением ее в контейнере вектор, выполняя доступ к элементам вектора с использованием выходных итераторов и обработку с использованием обобщенных алгоритмов (<code>copy()</code>).</p>
Входные и потоковые итераторы. Итераторы вставки. Решение задач.		51-54	<p><i>Знать.</i> Назначение, объявление и операции со входными итераторами. Особенности входных потоковых итераторов и итераторов вставки.</p> <p><i>Уметь.</i> Разработать и отладить программу обработки последовательности чисел с хранением ее в контейнере вектор, выполняя доступ к элементам вектора с использованием входных итераторов и обработку с использованием обобщенных алгоритмов (<code>copy()</code>, <code>find()</code>).</p>

Наименование раздела. Тема уроков	Кол. часов	№ урока	Характеристика основных видов деятельности ученика
X. Использование сортировки.	6		
<p>Модели вычислительной сложности алгоритмов – O-обозначения.</p> <p>Алгоритмы сортировки и их сложность.</p>		55-56	<p><i>Знать.</i> Способ оценки вычислительной сложности алгоритмов с помощью O-обозначений, оценки сложности распространенных алгоритмов сортировки. Обозначения параметров и правила использования обобщенного алгоритма <code>sort()</code> из библиотеки STL.</p> <p><i>Уметь.</i> Оценивать вычислительную сложность алгоритма решения задачи, применять функцию <code>sort()</code>.</p>
<p>Сортировки векторов структур по разным полям. Решение задач.</p>		57-60	<p><i>Знать.</i> Правила применения функции <code>sort()</code> для векторов встроенных типов данных, а также векторов из структур.</p> <p><i>Уметь.</i> Применять функцию <code>sort()</code> для упорядочивания значений как в порядке возрастания, так и в порядке убывания. Написать программу решения задачи.</p>
XI. Двумерные и многомерные структуры.	12		
<p>Особенности применения контейнеров STL для хранения и обработки двумерных данных.</p>		61-62	<p><i>Знать.</i> Правила объявления и инициализации двумерных векторов, доступа к отдельным элементам.</p> <p><i>Уметь.</i> Разработать и отладить программу обработки последовательности чисел с хранением ее в двумерном контейнере вектор.</p>
<p>Решение задач с обработкой двумерных числовых и символьных данных.</p>		63-66	<p><i>Знать.</i> Правила объявления и инициализации двумерных векторов, доступа к отдельным элементам, особенности применения обобщенных алгоритмов.</p> <p><i>Уметь.</i> Разработать и отладить программу обработки последовательности чисел с хранением ее в двумерном контейнере вектор.</p>

Наименование раздела. Тема уроков	Кол. часов	№ урока	Характеристика основных видов деятельности ученика
Особенности применения контейнеров STL для хранения и обработки простых изображений.		67-68	<i>Знать.</i> Правила объявления и инициализации двумерных векторов для хранения и обработки простых изображений, доступа к отдельным элементам. <i>Уметь.</i> Разработать и отладить программу обработки простого изображения с хранением его в двумерном контейнере вектор.
Решение задач с обработкой простых изображений.		69-71	<i>Знать.</i> Правила объявления и инициализации двумерных векторов для хранения и обработки простых изображений, доступа к отдельным элементам, особенности применения обобщенных алгоритмов. <i>Уметь.</i> Разработать и отладить программу обработки простого изображения с хранением его в двумерном контейнере вектор.
Контрольная работа 2		72	Применять полученные знания и умения при решении примеров и задач.
ХII. Контейнер список.	6		
Контейнеры последовательностей: список. Инициализация, ввод и вывод.		73-74	<i>Знать.</i> Правила объявления и инициализации списков, операции над ячейками списка и списками в целом. Функции-члены класса список: <code>push_back()</code> , <code>erase()</code> , <code>front()</code> , <code>insert()</code> . <i>Уметь.</i> Разработать и отладить программу обработки последовательности чисел с хранением ее в контейнере список.
Итераторы входные, выходные и однонаправленные. Решение задач.		75-78	<i>Знать.</i> Назначение и параметры функций-членов класса список: <code>splice()</code> . <i>Уметь.</i> Разработать и отладить программу обработки последовательности чисел с хранением ее в контейнере список.
ХIII. Контейнер стек.	6		
Контейнеры последовательностей:		79-80	<i>Знать.</i> Правила объявления и инициализации стеков. Функции-члены класса стек: <code>push()</code> , <code>pop()</code> , <code>top()</code> , <code>size()</code> .

Наименование раздела. Тема уроков	Кол. часов	№ урока	Характеристика основных видов деятельности ученика
стек. Инициализация, ввод и вывод.			<i>Уметь.</i> Разработать и отладить программу обработки последовательности чисел с хранением ее в контейнере стек.
Вычислительные задачи с обработкой данных в обратном порядке «последний пришел – первым обслужен».		81-84	<i>Знать.</i> Модель памяти LIFO «последний пришел – первым обслужен», условия ее применения. <i>Уметь.</i> Разработать и отладить программу обработки последовательности чисел с хранением ее в контейнере стек.
XIV. Контейнеры очередь, дек.	6		
Контейнеры последовательностей: очередь и дек. Инициализация, ввод и вывод.		85-86	<i>Знать.</i> Правила объявления и инициализации очередей (деков). Функции-члены класса очередь: push(), pop(), front(), size(). <i>Уметь.</i> Разработать и отладить программу обработки последовательности чисел с хранением ее в контейнере очередь (дек).
Вычислительные задачи с обработкой данных в порядке поступления «первый пришел – первым обслужен»		87-90	<i>Знать.</i> Модель памяти FIFO «первый пришел – первым обслужен», условия ее применения. <i>Уметь.</i> Разработать и отладить программу обработки последовательности чисел с хранением ее в контейнере очередь (дек).
XV. Двоичный поиск в массиве.	6		
Задача поиска информации; линейный и логарифмический (двоичный) поиск в массиве (векторе): алгоритмы и сравнение.		91-92	<i>Знать.</i> Алгоритмы и параметры вычислительной сложности основных алгоритмов поиска в массиве (векторе), области применимости алгоритмов поиска. <i>Уметь.</i> Разработать и отладить программу с применением двоичного поиска в контейнере вектор.
Обобщенные алгоритмы STL, связанные с двоичным		93-96	<i>Знать.</i> Правила применения функций двоичного поиска binary_search(), upper_bound() и lower_bound() для векторов.

Наименование раздела. Тема уроков	Кол. часов	№ урока	Характеристика основных видов деятельности ученика
поиском. Решение задач.			<i>Уметь.</i> Разработать и отладить программу с применением функций двоичного поиска в контейнере вектор.
XVI. Двоичный поиск в по ответу.	6		
Задачи двоичного поиска по ответу.		97-98	<i>Знать.</i> Способ двоичного поиска по ответу как средство уменьшения вычислительной сложности алгоритма решения задач; условия его применения. <i>Уметь.</i> Разработать и отладить программу с применением двоичного поиска по ответу.
Решение задач методом двоичного поиска по ответу.		99-101	<i>Знать.</i> Структуру алгоритма двоичного поиска по ответу. <i>Уметь.</i> Разработать и отладить программу с применением двоичного поиска по ответу.
Контрольная работа 3		102	Применять полученные знания и умения при решении примеров и задач.

Ожидаемые результаты освоения программы

Личностные, метапредметные и предметные результаты освоения курса основ программирования 8 класса

Личностные результаты:

- воспитание российской гражданской идентичности: патриотизма, уважения к Отечеству, осознание вклада отечественных ученых в развитие мировой науки;
- ответственное отношение к учению, готовность к саморазвитию и самообразованию;
- осознанный выбор и построение дальнейшей индивидуальной траектории образования;
- умение контролировать процесс и результат учебной деятельности;
- критичность мышления, инициатива, активность при решении алгоритмических задач.

Метапредметные результаты:

- умение самостоятельно определять цели своего обучения, развивать мотивы и интересы своей познавательной деятельности;
- умение соотносить свои действия с планируемыми результатами;
- умение определять понятия, обобщать, устанавливать аналогии, классифицировать;
- развивать компетенции в области использования информационно-коммуникационных технологий;
- умение находить информацию в различных источниках;
- умение выдвигать гипотезы;
- понимать сущности алгоритмических предписаний;
- устанавливать причинно-следственные связи, проводить доказательные рассуждения;
- умение иллюстрировать изученные понятия и свойства алгоритмов и программ.

Предметные результаты:

- осознание значения алгоритмизации и программирования для повседневной жизни;
- развитие умений работать с математическим текстом;
- выражать свои мысли с применением терминологии компьютерной математики и теоретических основ информатики и программирования;
- владение базовым понятийным аппаратом по основным разделам содержания;
- практически значимые умения и навыки алгоритмизации и программирования, их применение к решению математических и алгоритмических задач.

В результате освоения программы учащиеся должны:

приобрести следующие профессиональные компетенции:

владеть: технологиями дистанционного обучения программированию на языках высокого уровня с применением систем автоматической проверки решений;

уметь: разработать и реализовать приложение консольного типа в интегрированной среде разработки программ Visual Studio (или подобной); найти и устранить логические ошибки в программе в режиме пошаговой отладки;

знать: структуру программы и основные типы данных, управляющие конструкции языка C++, способы создания иерархических программных систем и элементы технологии разработки программного обеспечения, приемы работы с библиотекой STL.

Результативность программы определяется по срезам знаний после каждого года обучения, творческих заданий, проведению конкурсов и викторин, по практическому изготовлению изделий и участию в выставках и конкурсах.

Программа предполагает входную, промежуточную и итоговую аттестацию обучающихся.

Входной контроль осуществляется в форме собеседования и направлен на диагностику начального уровня обучающихся.

Промежуточный контроль проводится в течение года и позволяет выявлять уровень освоения разделов и тем программы в форме педагогического наблюдения, бесед, творческих заданий.

Итоговый контроль осуществляется в конце обучения и направлен на выявление уровня освоения программы за весь срок обучения.

Итоговый контроль - выполнение итоговых работ по основным разделам программы.

Правила выбора темы итогового проекта

Итоговым результатом освоения обучающимися полученных в процессе обучения навыков и компетенций в рамках представленной программы является итоговая защита проекта. Так как от выбора темы проекта зависит качество проделанной самостоятельной работы, а также итоговая защита проекта, зачастую у обучающихся возникает проблема выбора темы итогового проекта.

Поэтому необходимо помочь обучающимся найти все пути, ведущие к достижению цели, выделить общепринятые, общеизвестные и нестандартные, альтернативные; сделать выбор, оценив эффективность каждого способа.

Правило 1. Тема должна быть интересна ребенку, должна увлекать его. Исследовательская работа эффективна только на добровольной основе. Тема, навязанная ученику, какой бы важной она ни казалась взрослым, не даст должного эффекта.

Правило 2. Тема должна быть выполнима, решение ее должно быть полезно участникам исследования. Натолкнуть ребенка на ту идею, в которой он максимально реализуется как исследователь, раскроет лучшие стороны своего интеллекта, получит новые полезные знания, умения и навыки, – сложная, но необходимая задача для педагога.

Правило 3. Тема должна быть оригинальной с элементами неожиданности, необычности. Оригинальность следует понимать, как способность нестандартно смотреть на традиционные предметы и явления.

Правило 4. Тема должна быть такой, чтобы работа могла быть выполнена относительно быстро. Способность долго концентрировать собственное внимание на одном объекте, т. е. долговременно, целеустремленно работать в одном направлении, у школьника ограничена.

Правило 5. Тема должна быть доступной. Она должна соответствовать возрастным особенностям детей. Это касается не только выбора темы исследования, но и формулировки и отбора материала для ее решения. Одна и та же проблема может решаться разными возрастными группами на различных этапах обучения.

Правило 6. Сочетание желаний и возможностей. Выбирая тему, педагог должен учесть наличие требуемых средств и материалов – исследовательской базы. Ее отсутствие, невозможность собрать необходимые данные обычно приводят к поверхностному решению,

порождают "пустословие". Это мешает развитию критического мышления, основанного на доказательном исследовании и надежных знаниях.

Правило 7. С выбором темы не стоит затягивать. Большинство обучающихся не имеют постоянных пристрастий. Поэтому, выбирая тему, действовать следует быстро, пока интерес не угас.

УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

Кадровое обеспечение

Уровень образования педагога: высшее образование (в том числе по направлениям, соответствующим направлениям дополнительных общеобразовательных программ, реализуемых организацией, осуществляющей образовательную деятельность) и отвечающими квалификационным требованиям, указанным в квалификационных справочниках, и (или) профессиональным стандарта Педагогом дополнительного образования пройдено повышение квалификации по направлению программы.

Информационное обеспечение

Сценарии, фотоматериалы, презентации, необходимые для проведения занятий и различных тематических мероприятий. Интернет источники.

Материально-техническое обеспечение

Для успешного выполнения кейсов потребуется следующее оборудование, материалы, программное обеспечение и условия. Количество единиц оборудования и материалов указано из расчета количественного состава группы обучающихся.

Учебная лекционная аудитория, компьютерный класс (12 – 15 рабочих мест). Компьютеры объединены в локальную сеть и имеют выход в Интернет, установлена операционная система Windows, Web-browser, MS Visual Studio, текстовый процессор Word.

Необходим высокоскоростной канал для подключения к сети Интернет.

Распределение комплектов оборудования и материалов – 1 комплект на обучающегося:

- работа над кейсом должна производиться в хорошо освещенном, просторном, проветриваемом помещении;
- компьютер (ноутбук) с монитором, клавиатурой и мышкой, на котором установлено следующее программное обеспечение: операционная система Windows (версия не ниже 10) / Linux, Python (версия не ниже 3.7.4), пакет Anaconda (версия не ниже 2020.02), среда Wing 101 (версия не ниже 7.2.2), среда разработки PyCharm (версия не ниже 2019.3.4), пакет офисных программ MS Office / LibreOffice ;
- компьютеры (ноутбуки) должны быть подключены к единой сети с доступом в Интернет;
- презентационное оборудование (проектор с экраном/телевизор с большим экраном) с возможностью подключения к компьютеру (ноутбуку) – 1 комплект;
- флипчарт с комплектом листов / маркерная доска, соответствующий набор письменных принадлежностей – 1 шт.;
- каждый стол для работы должен предоставлять достаточно места для работы.

Имеется электронная версия учебного пособия «Начальный курс программирования» на сайте moodle.cs.istu.ru, тем не менее желательно издание печатной версии пособия.

Учебно-методические материалы

Необходимые теоретические сведения и наборы задач к темам курса готовятся к загрузке на электронные ресурсы moodle.cs.istu.ru и bacs.cs.istu.ru. Необходимы финансовые средства на программно-техническую и организационно-методическую поддержку этих ресурсов.

Методическое обеспечение программы

Отличительной особенностью данной программы является ориентация на компетентностный подход, позволяющий обучающимся развивать и наращивать предметные и межпредметные компетенции.

Программой предусматриваются следующий методический инструментарий:

1. Набор наглядных материалов (папок по отдельным разделам и темам);
2. Презентации, слайд-фильмы, мастер-классы (по темам и разделам программы);
3. Технологические карты;
4. Различные дидактические материалы;
5. Папки: «Основные технологические приемы» (по основным разделам программы);
6. Методическая продукция: методические рекомендации, методические пособия, разработки диагностических материалов, разработки мероприятий, конспекты занятий;
7. Информационный стенд;
8. Банк творческих проектов обучающихся;
9. Педагогическая копилка: творческие игры, психологические тесты, игры на внимание, кроссворды, дидактические и познавательные игры, сценарии проведения массовых мероприятий.
10. Специальная литература;
11. Интернет-ресурсы.

Формы организации учебной деятельности:

- Групповая
- Индивидуальная/Самостоятельная
- Парная
- В малых группах

Формы занятий:

- Практическое занятие
- Консультация
- Беседа
- Используемые методы в рамках занятий:
- Проектный метод
- Проблемное обучение

Виды учебной деятельности в рамках занятий:

- Поиск и анализ информации
- Анализ и решение проблемных ситуаций
- Просмотр презентаций и видеороликов

Проведение исследовательских экспериментов
Публичное выступление и защита

Календарный учебный график

Полугодие	Месяц	Недел и обучен ия	Даты учебных недель	Год обучения	
1 полугодие	Сентябрь	1	02-07	У, ВА	
		2	09-14	У	
		3	16-21	У	
		4	23-28	У	
			5	30.09-5.10	У
	Октябрь	6	7-12	У	
		7	14-19	У	
		8	21-26	У	
		9		Каникулы с 27.10 П, каникулы по 4.11	
			10	5-9.11	У
	ноябрь	11	11-16	У	
		12	18-23	У	
		13	25-30	У	
		14	2-7.12	У	
	декабрь	15	9-14	У	
		16	16-21	У	
		17	23-28	У	
		18		П, каникулы с 30.12 П, каникулы по 08.01У	
2 полугодие	Январь	19	9.01- 11.01.2025	У	
		20	13-18	У	
		21	20-25	У	
		22	27-1.02	У	
	Февраль	23	3-8	У	
		24	10-15	У	
		25	17-22	У	
		26	24-1.03	У	
	Март	27	3-7	У	
		28	10-15	У	
		29	17-22	У	
		30		Каникулы с 24.03 по 29.03	
			31	31.03-5.04	У
	Апрель	32	7-12	У	
		33	14-19	У	
		34	21-26	У	
		35	28-3.05	У	
	Май	36	5-10	ПА	
		37	12-17	ИА	
38		19-24	ИА		
				Каникулы с 26.05 по 31.08	
	Всего учебных недель			34	
	Всего часов по программе			34	

	Дата учебного года	02.09.2024 г.
	Дата окончания учебного года	24.05.2025 г.

Условные обозначения:

У – учебная неделя

П – праздничная неделя

ВА – входная аттестация

ПА – промежуточная аттестация

ИА – итоговая аттестация

Формы и методы контроля

- наблюдение;
- устный опрос;
- устный анализ самостоятельных работ;
- тестирование, самостоятельные и контрольные работы;
- участие в проектной деятельности;
- промежуточная аттестация - диагностика по параметрам – в сентябре, декабре и мае;
- обучение заканчивается написанием программы для решения одной из задач;

Наблюдение – необходимый педагогу метод для осуществления промежуточной и текущей аттестации, применяется педагогом постоянно.

Устный опрос – метод при котором педагог может оценить теоретически знания учащихся. Проводится в конце пройденной темы.

Устный анализ самостоятельных работ – дает возможность учащимся научиться логически мыслить и уметь высказать собственное суждение, поможет педагогу оценить логическое мышление учащихся. Проводится в конце пройденной темы.

Тестирование, самостоятельные и контрольные работы – помогут учащимся проверить свои знания по пройденным темам и оценить приобретенные умения, проводятся в конце каждого раздела.

Участие в проектной деятельности – это условие, необходимое для становления самостоятельности учащихся, которые сформируют установку на самостоятельность, индивидуальные цели и возможности их реализации;

Промежуточная аттестация проводится три раза за учебный год: в начале учебного года – в сентябре, в середине учебного года – в декабре и в конце учебного года – в мае. Задачей промежуточной аттестации является определение эффективности педагогического воздействия и полноты освоения образовательной программы.

Параметры и критерии освоения программы в области специальных знаний языка программирования C++.

Оценочный материал Проверка теоретических знаний

Оценка качества освоения образовательной программы проводится по результатам собеседования по одному-двум вопросам из нижеследующего списка. Также могут быть предложены задачи для самостоятельного решения.

1. Препроцессор C++ и файл `iostream`. Имена заголовочных файлов.
2. Вывод в C++ с помощью `cout`. Манипулятор `endl`. Символ новой строки. Конкатенация с помощью `cout`.
3. Операторы объявления и переменные. Операторы присваивания.
4. Использование `cin`. Построчное чтение ввода.
5. Смешивание строкового и числового ввода.
6. `cin` и `cout`: признак класса.
7. Пространства имен. Местоположение директивы `using` в программах с множеством функций.
8. Циклы и текстовый ввод. Применение для ввода простого `cin`.
9. Выбор используемой версии `cin.get()`. Условие конца файла.
10. Целочисленные типы `short`, `int`, `long` и `long long`. Типы без знаков.
11. Выбор целочисленного типа. Целочисленные литералы. Определение компилятором C++ типа константы.
12. Тип `char`: символы и короткие целые числа.
13. Тип `bool`. Квалификатор `const`. Объявления `auto` в C++11.
14. Правила инициализации массивов.
15. Альтернативы массивам - шаблонный класс `array` (C++11).
16. Сравнение массивов и объектов `array`.
17. Строковый тип с нулевым символом-завершителем. Выделение подстроки. Нахождение первого вхождения буквы.
18. Функции для работы со строками.
19. Введение в класс `string`. Инициализация строк в C++11.
20. Выражения отношений. Формы строковых литералов.
21. Присваивание, сравнение и вероятные ошибки.
22. Сравнение строк в стиле C. Сравнение строк класса `string`.
23. STL – обобщенное программирование: связь контейнеров с итераторами.
24. Компоненты STL. Обобщенный алгоритм STL `reverse` со строкой и массивом.
25. Компоненты STL. Обобщенный алгоритм STL `find` с массивом и вектором.
26. Компоненты STL. Обобщенный алгоритм STL `find` со списком.
27. Компоненты STL. Обобщенный алгоритм STL `merge`.
28. Классификация итераторов STL.
29. Диапазоны итераторов. Входные и выходные итераторы.
30. Одно- и двунаправленные итераторы.
31. Итераторы с произвольным доступом.
32. Двоичный поиск заданного элемента в массиве.
33. Интегрированная среда разработки. Техпроцесс создания программы.

Оценки *«отлично»* заслуживает учащийся, обнаруживший всестороннее, систематическое и глубокое знание учебно-программного материала, умение свободно выполнять задания, предусмотренные программой, усвоивший основную и знакомый с дополнительной литературой, рекомендованной программой. Как правило, оценка «отлично» выставляется учащимся, усвоившим взаимосвязь основных понятий дисциплины в их значении для приобретаемой профессии, проявившим творческие способности в понимании, изложении и использовании учебно-программного материала.

Оценки *«хорошо»* заслуживает учащийся, обнаруживший полное знание учебно-программного материала, успешно выполняющий предусмотренные в программе задания, усвоивший основную литературу, рекомендованную в программе. Как правило, оценка «хорошо» выставляется учащимся, показавшим систематический характер знаний по дисциплине и способным к их самостоятельному пополнению и обновлению в ходе дальнейшей учебной работы и профессиональной деятельности.

Оценки *«удовлетворительно»* заслуживает учащийся, обнаруживший знания основного

учебно-программного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по специальности, справляющийся с выполнением заданий, предусмотренных программой, знакомый с основной литературой, рекомендованной программой. Как правило, оценка «удовлетворительно» выставляется учащимся, допустившим погрешности в ответе на экзамене и при выполнении экзаменационных заданий, но обладающим необходимыми знаниями для их устранения под руководством учителя.

Оценка *«неудовлетворительно»* выставляется учащемуся, обнаружившему пробелы в знаниях основного учебно-программного материала, допустившему принципиальные ошибки в выполнении предусмотренных программой заданий. Как правило, оценка «неудовлетворительно» ставится учащимся, которые не могут продолжить обучение или приступить к профессиональной деятельности по окончании вуза без дополнительных занятий по соответствующей дисциплине.

Контрольно-измерительные материалы

Таблица 1 - Критерии оценки проектов

Критерий	Показатель	Уровень	Балл	Диагностический инструментари й
Разнообразие источников информации, целесообразность их использования	Большая часть представленной информации не относится к теме работы	Низкий	1	Наблюдение
	Работа содержит незначительный объем подходящей информации из ограниченного числа однотипных источников	Средний	2	
	Работа содержит достаточно полную информацию из разнообразных источников	Высокий	3	
Заинтересованность группы, творческий подход к работе	Группа проявила незначительный интерес к теме проекта, но не продемонстрировала самостоятельность в работе, не использовала возможности творческого подхода	Низкий	1	Наблюдение
	Работа самостоятельная, демонстрирующая серьезную заинтересованность группы; предпринята попытка представить личный взгляд на тему проекта, применены элементы творчества	Средний	2	
	Работа отличается творческим подходом, собственным оригинальным отношением группы к идее проекта	Высокий	3	
Соответствие требованиям оформления письменной части	В письменной части отсутствуют установленные правилами порядок и четкая структура, допущены серьезные ошибки в оформлении	Низкий	1	Наблюдение
	Предприняты попытки оформить работу в соответствии с установленными правилами, придать ей соответствующую структуру	Средний	2	
	Работа отличается четким и грамотным оформлением в точном соответствии с установленными правилами	Высокий	3	
Качество защиты проекта	Материал изложен с учетом регламента, однако автору не удалось заинтересовать аудиторию	Низкий	1	Наблюдение

Критерий	Показатель	Уровень	Балл	Диагностический инструментари й
	Автору удалось вызвать интерес аудитории, но он вышел за рамки регламента	Средний	2	
	Автору удалось вызвать интерес аудитории и уложиться в регламент	Высокий	3	
Качество проектного продукта	Проектный продукт не соответствует требованиям качества (эстетика, удобство использования, соответствие заявленным целям)	Низкий	1	Тестирование
	Продукт не полностью соответствует требованиям качества	Средний	2	
	Продукт полностью соответствует требованиям качества (эстетичен, удобен в использовании, соответствует заявленным целям)	Высокий	3	

Таблица - Основные показатели оценки результата, формы и методы контроля и оценки по определению сформированности компетенций

Результаты (освоенные компетенции)	Основные показатели оценки результата	Формы и методы контроля и оценки
Soft-компетенции		
владение умениями самостоятельно планировать пути достижения целей; соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий, корректировать свои действия в соответствии с изменяющейся ситуацией; оценивать правильность выполнения учебной задачи;	- своевременность и качество выполнения учебных заданий; - обоснованность постановки цели, выбора и применения способа решения профессиональной задачи из известных в соответствии с реальными и заданными условиями и имеющимися ресурсами;	- наблюдение, оценка педагогом выполнения итогового проекта; - оценка педагогом обоснования собственной деятельности обучающегося;
владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;		- анализ и оценка педагогом рефлексии,

Результаты (освоенные компетенции)	Основные показатели оценки результата	Формы и методы контроля и оценки
<p>владение информационным моделированием как основным методом приобретения знаний: умение преобразовывать объект из чувственной формы в пространственно-графическую или знаково-символическую модель;</p>	<p>- рациональное распределение времени на все этапы работы;</p> <p>- самостоятельность обнаружения допущенных ошибок, своевременность коррекции деятельности на основе результатов самооценки продукта.</p>	<p>самооценки учебной деятельности обучающегося</p>
<p>формулирование вопросов, ответы на которые требуются для создания продукта, и другие навыки исследовательской деятельности;</p>		
<p>планирование и контроль процессов через проектную деятельность;</p>		
<p>разработка перечня ключевых показателей эффективности и их оценка;</p>		
<p>выступление с компьютерным сопровождением.</p>		
Hard-компетенции		
<p>– знание умение объяснять и использовать на практике как простые, так и сложные структуры данных и конструкции для работы с ними;</p> <p>– умение искать и обрабатывать ошибки в коде;</p> <p>– умение разбивать решение задачи на подзадачи;</p> <p>– способность писать грамотный, красивый код;</p> <p>– способность работать с информацией: находить, оценивать и использовать информацию из различных источников, необходимую для решения профессиональных задач (в том числе на основе системного подхода).</p>	<p>5 – 7 баллов: низкий уровень;</p> <p>8 – 11 баллов: средний уровень;</p> <p>12 – 15 баллов: высокий уровень.</p>	<p>- наблюдение, оценка преподавателем выполнения итогового проекта;</p> <p>- анализ и оценка преподавателем рефлексии, самооценки учебной деятельности обучающегося</p>

Примерные задачи для самостоятельного решения

1. Разные форматы вывода в языке С++: одно- и многострочный.
2. Напечатать заданное расположение наборов отрезков, перпендикулярных осям координат.
3. Напечатать изображение «игрового» поля с использованием от 1 до 5 символов.
4. Обработать числовую последовательность с формированием результатов в виде символьных кодов.
5. Обработать числовую последовательность с формированием результатов в виде искомым числовых наборов и текстовых пояснений.
6. Обработать числовую последовательность с преобразованием символьных последовательностей в числа и обратно.
7. Обработать числовую последовательность с преобразованием значений из одной системы счисления в другую.

Для каждого занятия подготовлен комплект задач в системе автоматической проверки решений – сайт moodle.cs.istu.ru. В системе организована регистрация участников, для каждого участника ведется учет его работы как в компьютерном классе, так и при выполнении самостоятельной работы дистанционно. Учителю доступны все решения учащихся: как ошибочные, так и прошедшие полную процедуру тестирования в автоматической системе.

РАБОЧАЯ ПРОГРАММА ВОСПИТАНИЯ, КАЛЕНДАРНЫЙ ПЛАН ВОСПИТАТЕЛЬНОЙ РАБОТЫ

Цель: личностное развитие учащихся средствами духовно-нравственных ценностей народов Российской Федерации, исторических и национально-культурных традиций; формирование культуры здорового и безопасного образа жизни и готовности к осознанному профессиональному выбору.

Направление. Формирование и развитие творческих способностей учащихся, выявление и поддержка талантливых учащихся.

Задачи:

создание условий для развития творческих способностей учащихся, оказание поддержки и сопровождение одаренных детей.

<i>Сроки</i>	<i>Мероприятие</i>
СЕНТЯБРЬ	
02-10.09.2024	Организация работы по реализации индивидуального образовательного маршрута по сопровождению проектной деятельности
сентябрь - май	Презентация программы на родительских собраниях
в течение года, согласно Положения о конференциях, конкурсах	Участие в школьных, районных, городских, республиканских, всероссийских научно-практических конференциях
ОКТАБРЬ	
в течение года, согласно Положения о конкурсах	Участие учащихся в конкурсах, выставках, олимпиадах различного уровня
НОЯБРЬ	
28.10 - 04.11.2024	Работа с мотивированными детьми в дни школьных каникул
23.10.2024	Школьная олимпиада
ДЕКАБРЬ	
1-31.12.2024	районная олимпиада
ЯНВАРЬ	
1-8.01.2025	Творческая работа с учащимися
	Презентация проектов
ФЕВРАЛЬ	
	Беседа «Открытия, которые потрясли мир»
	Конкурс компьютерных рисунков
МАРТ	
24-29.03.2025	Работа с мотивированными детьми в дни школьных каникул
АПРЕЛЬ	
	Участие в конференции технических и творческих проектов
	Экскурсии на выставки мастеров и умельцев города
МАЙ	
	Участие в дистанционных конкурсах
	Презентация проектов
22.05.2025	Подведение итогов

Список литературы

Перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы.

Основная литература

1. С. Прата. Язык программирования C++. Лекции и упражнения, 6-е изд. : Пер. с англ. - М. : ООО "И.Д. Вильямс", 2012. - 1248 с.
2. Д. Р. Мюссер, Ж. Дж. Дердж, А. Сейни. C++ и STL: справочное руководство, 2-е изд. (серия C++ in Depth): Пер. с англ. - М.: 000 "И.Д. Вильямс", 2010. — 432 с.
3. Электронный ресурс moodle.cs.istu.ru.

Дополнительная литература

1. Б. Керниган, Д. Ритчи. Язык программирования Си.\Пер. с англ., 3-е изд., испр. - СПб.: "Невский Диалект", 2001.
2. В. Давыдов. Visual C++. - СПб.: Изд-во «БХВ», 2008.

Разработчик программы: В.Г.Тарасов, профессор кафедры программного обеспечения ИЖГТУ имени М.Т.Калашникова