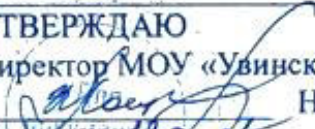


МУНИЦИПАЛЬНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
«УВИНСКАЯ СРЕДНЯЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА №2
С УГЛУБЛЕННЫМ ИЗУЧЕНИЕМ ОТДЕЛЬНЫХ ПРЕДМЕТОВ»

РАССМОТРЕНО На заседании ШМО Протокол № <u>1</u> « <u>30</u> » <u>августа</u> 20 <u>24</u> г	ПРИНЯТО Педагогическим советом школы Протокол № <u>1</u> « <u>30</u> » <u>августа</u> 20 <u>24</u> г	УТВЕРЖДАЮ Директор МОУ «Увинская СОШ №2»  Н.В.Лощилова Приказ № <u>130 от</u> « <u>30</u> » <u>августа</u> 20 <u>24</u> г.г
---	---	---

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
«Программирование на C++»
(второй год обучения)**

Разработчик: Пасынкова О.А.,
учитель информатики,
педагог дополнительного образования
МОУ «Увинская СОШ № 2»

2024-2025 учебный год

СОДЕРЖАНИЕ

I. КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ПРОГРАММЫ			
1.	Пояснительная записка		3-5
	1.1	Направленность программы	3
	1.2	Актуальность программы	3
	1.3	Отличительные особенности программы	4
	1.4	Нормативно-правовые основания	4
	1.5	Адресат программы	5
	1.6	Объем программы	5
	1.7	Организация образовательного процесса	5
2.	Цель и задачи программы		5-6
3.	Содержание программы		6-10
	3.1	Учебно-тематическое планирование	6-9
	3.2	Содержание учебно-тематического плана	9-13
4.	Планируемые результаты		13-14
II. КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ			
1.	Календарный учебный график		14-16
2.	Условия реализации программы		16-17
	2.1	Материально-техническое обеспечение	16
	2.2	Кадровое обеспечение	17
3.	Формы аттестации/контроля		17-18
4.	Календарный план воспитательных мероприятий		18
5.	Оценочные материалы		18-19
6.	Методические материалы		19-20
7.	Список литературы		20

I. КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ДОПОЛНИТЕЛЬНОЙ ОБЩЕОБРАЗОВАТЕЛЬНОЙ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ /далее – Программа, ДООП/

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1.1. Направленность программы

Настоящая программа имеет *техническую направленность*. Способствует формированию у школьников информационной и функциональной компетентности, развитию алгоритмического мышления и навыкам программирования.

1.2. Актуальность программы

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.

Программа составлена на основе

- фундаментального ядра содержания общего образования;
- требований к результатам освоения образовательной программы, представленных в федеральном государственном стандарте основного общего образования;

В программе доминируют идеи и положения программы развития и формирования универсальных учебных действий, обеспечивающие формирование

- российской гражданской идентичности;
- коммуникативных качеств личности;
- ключевой компетенции – умения учиться;
- алгоритмического мышления, необходимого для успешного освоения курса программирования.

1.3. Отличительные особенности программы

Базисный учебный план отводит на изучение информатики 1 учебный час в неделю (34 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.

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.

1.4. Нормативно-правовые основания

Программа разработана в соответствии с нормативными документами и не противоречит законодательству РФ.

1.5. Адресат программы

Программа рассчитана на обучающихся 8 классов.

1.6. Объем программы

Срок реализации программы – 1 год (102 часа).

1.7. Организация образовательного процесса

Режим занятий: 3 занятия в неделю, общее количество часов в год – 102.

Продолжительность занятия – 40 минут.

Распределение учебного времени занятия определяется в соответствии с возрастом детей и требованиями СанПиН.

При организации образовательного процесса используются фронтальные, групповые и индивидуальные формы работы.

2. ЦЕЛЬ И ЗАДАЧИ ПРОГРАММЫ

Цель:– обучение программированию на языке C++ учащихся 13-15 лет и старше образовательных школ.

Задачи:

Образовательные:

- развить умение самостоятельно определять цели своего обучения, мотивы и интересы своей познавательной деятельности; соотносить свои действия с планируемыми результатами;
- научить пользоваться базовым понятийным аппаратом по основным разделам содержания программы; обобщать, устанавливать аналогии,
- классифицировать, выдвигать гипотезы и устанавливать причинно-следственные связи; визуализировать изученные понятия и свойства алгоритмов и программ;
- привить практически значимые умения и навыки алгоритмизации и программирования, их применение к решению математических и алгоритмических задач

Развивающие:

- способствовать развитию критического, системного, алгоритмического и творческого мышления;
- развивать внимание, память, наблюдательность; познавательный интерес;
- развивать умение работать с компьютерными программами и дополнительными источниками информации;
- развивать навыки планирования проекта, умение работать в группе.

Воспитательные:

- формировать положительное отношение к информатике и ИКТ;
- развивать самостоятельность и формировать умение работать в паре, малой группе, коллективе;
- формировать умение демонстрировать результаты своей работы.

3. СОДЕРЖАНИЕ ПРОГРАММЫ

3.1. Учебный план

Наименование разделов (модулей)	Всего часов	в т.ч. практические занятия	в т.ч. лабораторные занятия	Формы, виды контроля
Тема 1. История языка C++ и существующие	18	17	17	

стандарты. Поточковый ввод и вывод в языке C++. Новые элементы в C++ в представлении базовых типов данных и массивов и в работе с ними.				
Тема 2. Строки с завершающим нулем. Класс String: свойства и методы. Массивы строк.	18	18	18	Итоговая работа
Тема 3. Библиотека STL: принципы проектирования и функционирования. Контейнеры, итераторы, обобщенные алгоритмы. Последовательный контейнер вектор : свойства и методы.	18	18	18	
Тема 4. Быстрые методы и алгоритмы сортировки последовательностей с применением последовательных контейнеров.	12	12	12	
Тема 5. Двумерные и многомерные структуры. Решение задач.	12	12	12	Итоговая работа
Тема 6. Алгоритмы двоичного поиска и их применение к решению задач.	12	12	12	
Тема 7. Последовательные контейнеры список, стек и очередь: свойства и методы. Решение задач.	12	12	12	Итоговая работа
Всего:	102	101	101	

3.2. Учебно-тематический план

Наименование раздела. Тема уроков	Кол. часов		Характеристика основных видов деятельности ученика
	Всего	Практические работы	
I. История языка C++ и существующие стандарты. Поточковый ввод и вывод в языке C++	6	5	

Отличие объектно-ориентированного подхода разработки программ от процедурного.	1		<p><i>Знать.</i> Объектно-ориентированное программирование – подход для проектирования больших программных систем. Взаимосвязь понятий объекта и класса на примере объектов cin и cout, являющихся представителями классов ostream для потоков входной и выходной символьной информации.</p> <p><i>Уметь.</i> Составлять операторы ввода и вывода числовой и символьной информации для решения простых линейных алгоритмов.</p>
Стиль написания исходного кода и операторы в языке C++	1	1	<p><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.</p> <p><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.</p>
Знакомство со средой программирования. Набор и запуск программ.	4	4	<p><i>Знать.</i> Последовательность запуска среды программирования, создания проекта, подключения и 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.</p>

			Зарегистрироваться в системе удаленной проверки задач, получить условие задачи, отправить решение на проверку, просмотреть результат.
II. Циклы и тестовый ввод.	6	6	
Функции get(), eof() и fail() объекта cin и put() объекта cout для более точной работы с символами.	3	3	<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(). <i>Уметь.</i> Составлять алгоритмы, завершающие работу при завершении данных во входном потоке.
Циклы с пред- и постусловием при вводе символьной информации. Решение задач.	3	3	<i>Знать.</i> Новые возможности инициализации массивов в C++. <i>Уметь.</i> Написать программу решения задачи обработки числовых и символьных последовательностей с применением циклов с пред- и постусловием.
III. Системы счисления.	6	6	
Позиционная система счисления: алфавит, основание, представление целых чисел и преобразования	3	3	<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. <i>Уметь.</i> Выполнять преобразования целых

			чисел из внешнего представления (текстовой десятичной записи) во внутреннее и наоборот.
Новые элементы в C++ в представлении целых чисел и в работе с ними. Решение задач.	3	3	<i>Знать.</i> Встроенные типы данных для работы с целыми числами, правила записи и операции с целыми в языке C++. <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.	6	6	
Строки с завершающим нулем. Представление в памяти, инициализация в C++11, ввод и вывод. Работа с указателями.	3	3	<i>Знать.</i> Массив символов как хранилище (контейнер) строк в стиле C. 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.	3	3	<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	6	
Введение в класс string. Инициализация строк в C++11, ввод и вывод. Присваивание, сравнение и вероятные ошибки. Основные функции.	3	3	<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.	3	3	<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	6	
Массивы String.	3	3	<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.	3	3	<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.
VII. Векторы почти как массивы. Создание, методы.	6	6	
Отличие STL от других библиотек. Контейнеры последовательностей: вектор. Инициализация, ввод и вывод.	3	3	<i>Знать.</i> Правила объявления и инициализации векторов, операции над ячейками вектора и векторами в целом. Функции-члены класса вектор <code>push_back()</code> , <code>capacity()</code> . <i>Уметь.</i> Разработать и отладить программу обработки последовательности чисел с хранением ее в контейнере вектор.
Функции-члены класса вектор <code>clear()</code> , <code>swap()</code> . Решение задач.	3	3	<i>Знать.</i> Назначение и параметры функций-членов класса вектор <code>clear()</code> , <code>swap()</code> . <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	6	
Итераторы – связующий элемент между контейнерами и алгоритмами: определения, классификация. Операции над итераторами.	3	3	<i>Знать.</i> Назначение и классификацию итераторов, правила их объявления, диапазоны и допустимые операции. Функции класса вектор, возвращающие значения итераторов: <code>begin()</code> , <code>end()</code> . <i>Уметь.</i> Разработать и отладить программу обработки последовательности чисел с хранением ее в контейнере вектор, выполняя доступ к элементам вектора с использованием итераторов.

Обобщенные алгоритмы: определения, классификация, примеры. Решение задач.	3	3	<p><i>Знать.</i> Классификацию алгоритмов. Назначение и основные параметры обобщенных алгоритмов, правила использования в C++ (adjacent_difference(), copy(), transform ()).</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	6	
Обратный итератор. Выходные и потоковые итераторы.	3	3	<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 (copy()).</p>
Входные и потоковые итераторы. Итераторы вставки. Решение задач.	3	3	<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</p>

			доступ к элементам вектора с использованием входных итераторов и обработку с использованием обобщенных алгоритмов (copy(), find()).
Х. Использование сортировки.	6	6	
Модели вычислительной сложности алгоритмов – O-обозначения. Алгоритмы сортировки и их сложность.	3	3	<i>Знать.</i> Способ оценки вычислительной сложности алгоритмов с помощью O-обозначений, оценки сложности распространенных алгоритмов сортировки. Обозначения параметров и правила использования обобщенного алгоритма sort() из библиотеки STL. <i>Уметь.</i> Оценивать вычислительную сложность алгоритма решения задачи, применять функцию sort().
Сортировки векторов структур по разным полям. Решение задач.	3	3	<i>Знать.</i> Правила применения функции sort() для векторов встроенных типов данных, а также векторов из структур. <i>Уметь.</i> Применять функцию sort() для упорядочивания значений как в порядке возрастания, так и в порядке убывания. Написать программу решения задачи.
XI. Двумерные и многомерные структуры.	8	8	
Особенности применения контейнеров STL для хранения и обработки двумерных данных.	2	2	<i>Знать.</i> Правила объявления и инициализации двумерных векторов, доступа к отдельным элементам. <i>Уметь.</i> Разработать и отладить программу обработки последовательности чисел с

			хранением ее в двумерном контейнере вектор.
Решение задач с обработкой двумерных числовых и символьных данных.	2	2	<i>Знать.</i> Правила объявления и инициализации двумерных векторов, доступа к отдельным элементам, особенности применения обобщенных алгоритмов. <i>Уметь.</i> Разработать и отладить программу обработки последовательности чисел с хранением ее в двумерном контейнере вектор.
Особенности применения контейнеров STL для хранения и обработки простых изображений.	2	2	<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.	2	2	<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.
ХII. Двоичный поиск в массиве.	6	6	

Задача поиска информации; линейный и логарифмический (двоичный) поиск в массиве (векторе): алгоритмы и сравнение.	3	3	<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 поиском. Решение задач.	3	3	<i>Знать.</i> Правила применения функций двоичного поиска <code>binary_search()</code> , <code>upper_bound()</code> и <code>lower_bound()</code> для векторов. <i>Уметь.</i> Разработать и отладить программу с применением функций двоичного поиска в контейнере вектор.
XIII. Двоичный поиск в по ответу.	7	7	
Задачи двоичного поиска по ответу.	3	3	<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.	4	4	<i>Знать.</i> Структуру алгоритма двоичного поиска по ответу. <i>Уметь.</i> Разработать и отладить программу с применением двоичного поиска по ответу.
XIV. Контейнер список.	7	7	

Контейнеры последовательностей: список. Инициализация, ввод и вывод.	3	3	<i>Знать.</i> Правила объявления и инициализации списков, операции над ячейками списка и списками в целом. Функции-члены класса список: push_back(), erase(), front(), insert(). <i>Уметь.</i> Разработать и отладить программу обработки последовательности чисел с хранением ее в контейнере список.
Итераторы входные, выходные и односторонние. Решение задач.	4	4	<i>Знать.</i> Назначение и параметры функций-членов класса список: splice(). <i>Уметь.</i> Разработать и отладить программу обработки последовательности чисел с хранением ее в контейнере список.
XV. Контейнер стек.	6	6	
Контейнеры последовательностей: стек. Инициализация, ввод и вывод.	3	3	<i>Знать.</i> Правила объявления и инициализации стеков. Функции-члены класса стек: push(), pop(), top(), size(). <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».	3	3	<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.
XVI. Контейнеры очередь, дек.	8	8	

Контейнеры последовательностей: очередь и дек. Инициализация, ввод и вывод.	3	3	<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»	3	3	<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).
Контрольная работа 3	2	2	Применять полученные знания и умения при решении примеров и задач.

3.3 Содержание

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 быстрая сортировка (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. 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.

Тема 7. 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.

Итоговый контроль. Количество часов – 1 час.

4. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДООП

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.

II. КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ

1. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК /с 02.09.2024г. по 25.05.2025г./

Месяц	Сроки изучения учебного материала/нед	№ группы/год обучения/			Аттестация обучающихся	Каникулярный период
		Количество часов				
		№ 1/1	-	-		
Сентябрь	1 неделя	3				
	2 неделя	3				
	3 неделя	3				
	4 неделя	3				
	5 неделя	3				
Октябрь	1 неделя	3				
	2 неделя	3				
	3 неделя	3				
	4 неделя	3				
Ноябрь	1 неделя	Самостоятельная работа				
	2 неделя	3				
	3 неделя	3				

	4 неделя	3				
Декабрь	1 неделя	3				
	2 неделя	3				
	3 неделя	3				
	4 неделя	3				
	5 неделя	3				
Январь	1 неделя	Самостоятельная работа				
	2 неделя	3				
	3 неделя	3				
	4 неделя	3				
Февраль	1 неделя	3				
	2 неделя	3				
	3 неделя	3				
	4 неделя	3				
Март	1 неделя	3				
	2 неделя	3				
	3 неделя	Самостоятельная работа				
	4 неделя	3				
Апрель	1 неделя	3				
	2 неделя	3				
	3 неделя	3				
	4 неделя	3				
Май	1 неделя	3				
	2 неделя	3				
	3 неделя	3			Итоговая аттестация	
	4 неделя					
Всего учебных	34					

неделя						
Всего часов по программе		68				

2. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

2.1. Материально-техническое обеспечение

Для реализации дополнительной общеобразовательной общеразвивающей программы «Программирование на С++» необходим специально оборудованный учебный кабинет для проведения занятий.

Для эффективности образовательного процесса необходимы:

Техническое оборудование:

- проектор, экран
- компьютеры объединенные в локальную сеть и имеющие выход в Интернет,
- установленная на компьютерах операционная система Windows, Web-browser, Code Blocks, текстовый процессор Word.

Методическое обеспечение:

- дидактические материалы (опорные конспекты, проекты примеры, раздаточный материал для практических работ).

- учебно-тематический план.

2.2. Кадровое обеспечение

Программа реализуется педагогом дополнительного образования, имеющим профессиональное образование по направленности программы.

3. ФОРМЫ АТТЕСТАЦИИ/КОНТРОЛЯ

Система мониторинга достижения детьми планируемых результатов освоения дополнительной общеобразовательной общеразвивающей программы обеспечивает комплексный подход к оценке текущих и промежуточных результатов программы и позволяет осуществить оценку динамики достижения обучающихся, а также

включает описание объекта, форм, периодичность и содержание мониторинга. Отслеживание результатов осуществляется через входной, текущий, промежуточный и итоговый контроль.

Время проведения	Цель проведения	Формы контроля
Входной контроль		
В начале учебного года	Определение уровня развития детей, их знания ИКТ.	беседа
Текущий контроль		
В течение учебного года	Определение степени усвоения обучающимися учебного материала. Определение готовности детей к восприятию нового материала. Выявление детей, отстающих и опережающих обучение. Подбор наиболее эффективных методов и средств обучения.	Задачи проверяются в системе дистанционного обучения с автоматической проверкой на сайте http://new.moodle.cs.istu.ru/
Промежуточный контроль		
В конце месяца, полугодия	Определение степени усвоения обучающимися учебного материала. Определение результатов обучения.	Проверочные работы
Итоговый контроль		

В конце учебного года /или курса обучения/	Определение изменения уровня развития детей, их творческих способностей. Определение результатов обучения.	Итоговая контрольная работа
--	--	-----------------------------

4. КАЛЕНДАРНЫЙ ПЛАН ВОСПИТАТЕЛЬНЫХ МЕРОПРИЯТИЙ

Название мероприятия (тема)	Срок (месяц)
Проведение классных часов и родительских собраний	Сентябрь 2023
Участие в конкурсах, олимпиадах, экскурсии в ИжГТУ	Февраль, март 2024
Итоговая контрольная работы	Май 2024

5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

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 С++ и файл iostream. Имена заголовочных файлов.
2. Вывод в С++ 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.

Контрольно-измерительные материалы размещены на <http://new.moodle.cs.istu.ru/>

– 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

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.

6. СПИСОК ЛИТЕРАТУРЫ

1. Программирование на C, C++ для детей, 8—11 класс. / С. Солнечная — «Издательские решения»,.
2. Пахомов Б. И. C/C++ и MS Visual C++ 2012 для начинающих. — 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2015.
3. Программирование. Python.C++. Учебное пособие/ К.Ю.Поляков. – М.: БИНОМ. Лаборатория знаний,2018.
4. С. Прата. Язык программирования C++. Лекции и упражнения, 6-е изд. : Пер. с англ. - М. : ООО "И.Д. Вильямс", 2012. - 1248 с.
5. Д. Р. Мюссер, Ж. Дж. Дердж, А. Сейни. C++ и STL: справочное руководство, 2-е изд. (серия C++ in Depth): Пер. с англ. - М.: ООО "И.Д. Вильямс", 2010. — 432 с.
6. Б. Керниган, Д. Ритчи. Язык программирования Си. Пер. с англ., 3-е изд., испр. - СПб.: "Невский Диалект", 2001.
7. В. Давыдов. Visual C++. - СПб.: Изд-во «БХВ», 2008.

ЭЛЕКТРОННЫЕ ОБРАЗОВАТЕЛЬНЫЕ РЕСУРСЫ

1. <http://new.moodle.cs.istu.ru>
2. <https://www.lektorium.tv/basics-of-programming>