

Муниципальное бюджетное общеобразовательное учреждение
«Международный образовательный комплекс «Гармония» г. Ижевска

РАССМОТРЕНО
на заседании
предметной лаборатории
Протокол № 1 от 28.08. 2024

ПРИНЯТА
на заседании Педагогического совета
Протокол № 11 от 28.08.2024

СОСТАВЛЕНА в соответствии с
требованиями к результатам
освоения основной образовательной
программы основного общего
образования
УТВЕРЖДЕНА
Приказом директора № 209 от
29.08.2024

РАБОЧАЯ ПРОГРАММА

по программированию на C++

8 классы

Составитель: Ившин Александр Николаевич
ФИО педагога, должность, категория

учитель информатики, высшей категории

2024\2025 учебный год

Пояснительная записка.

Рабочая программа учебного курса «Программирование на С++, 2 год обучения» составлена на основе следующих документов:

- закон 273 «Об образовании в РФ» от 29.12.2012
- приказ Министерства образования РФ №1897 от 17.12.2010 года «Об утверждении федерального государственного стандартов основного общего и образования».
- приказ Министерства образования и науки Российской Федерации от 31.12.2015 № 1577 «О внесении изменений в федеральный государственный образовательный стандарт основного общего образования, утвержденный приказом Министерства образования и науки Российской Федерации от 17 декабря 2010 г. № 1897» (Зарегистрирован в Минюсте России 02.02.2016 №40937)
- образовательная программа школы Гармония
- учебный план школы Гармония
- локальный акт о рабочей программе педагога школы Гармония в соответствии с требованиями ФГОС ООО
- МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Ижевский государственный технический университет имени М.Т. Калашникова» Рабочая программа для учащихся 8-х классов «Программирование на С++»

Программа предназначена для 8 классов и предусматривает 2 часа в неделю.

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.

Промежуточный и итоговый контроль качества знаний при освоении данного курса осуществляется автоматизированной системой проверки, разработанной специалистами Ижевского государственного технического университета имени М.Т. Калашникова. Система расположена на сайте университета по адресу: new.moodle.cs.istu.ru

Содержание курса

Тема 1. Типы данных в C++

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 CodeBlocks, выполнение нескольких заданий в рамках одного проекта.

Тема 2. Обработка строк

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 CodeBlocks.

Тема 3. Библиотека STL

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

Обобщенные алгоритмы STL. Неизменяющиеся 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. Способы сортировки массива

Реализация и сравнение сортировок: пузырьковая, поразрядным группированием и быстрая сортировка (quicksort).

Практическая часть:

Применение обобщенных алгоритмов сортировки из библиотеки STL.

Тема 5. Обработка двумерных массивов

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. Организация поиска элемента

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. Списки и деки

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.

Планируемые результаты изучения учебного предмета

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:

Ученик научится:

- использовать технологии дистанционного обучения программированию на языках высокого уровня с применением систем автоматической проверки решений;
- разрабатывать и реализовывать приложение консольного типа в интегрированной среде разработки программ Code::Blocks;
- находить и устранять логические, синтаксические и алгоритмические ошибки в программе в режиме пошаговой отладки;
- использовать 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..

Ученик получит возможность:

- познакомиться со способами решения олимпиадных задач;
- освоить математические знания в опережении учебной программы по математике 7 класса.

Ученик овладеет:

- интегрированной средой разработки программ Code::Blocks

Итоговой аттестацией является выполнение итоговых работ по основным разделам программы.

Тематический план

Раз-дел	Тема	Кол. часов	Основные виды учебной деятельности	Контрольные и практические работы
1	Потоковый ввод и вывод в языке C++	4	<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> • Знать, что объектно-ориентированное программирование – подход для проектирования больших программных систем. • Понимать взаимосвязь понятий объекта и класса на примере объектов cin и cout, являющихся представителями классов ostream для потоков входной и выходной символьной информации. • Понимать структуру программы на языке C++. • Выделять диапазон изменения целых чисел. • Понимать особенности представления и обработки символов в C++. • Знать правила записи операторов присваивания, ввода и вывода. <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> • Составлять операторы ввода и вывода числовой и символьной информации для решения простых линейных алгоритмов • 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. • 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>Практическая работа №1. Практическая работа №2. Практическая работа №3.</p>
2	Циклы и тестовый ввод	4	<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> • Использовать правило «пропуска символов-разделителей» при чтении символов с помощью объекта cin. • Использовать средства обнаружения завершения данных в cin: биты eofbit и failbit, функции eof() и fail(). <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> • Составлять алгоритмы, завершающие работу при завершении данных во входном потоке • Написать программу решения задачи обработки числовых и символьных последовательностей с применением циклов с пред- и постусловием 	<p>Практическая работа №4. Практическая работа №5. Практическая работа №6.</p>

3	Системы счисления	4	<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> • Понимать основы двоичной и шестнадцатеричной систем счисления. • Понимать и использовать алгоритмы перевода чисел между двоичной, десятичной и шестнадцатеричной системами счисления. • Понимать внутреннее представление целых чисел и символов в памяти компьютера. • знать типы данных для работы с целыми числами, правила записи и операции с целыми в языке C++. <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> • Выполнять преобразования целых чисел из внешнего представления (текстовой десятичной записи) во внутреннее и наоборот • 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 	<p>Практическая работа №7. Практическая работа №8. Практическая работа №9. Практическая работа №10.</p>
4	Строки в стиле С	4	<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> • Определять массив символов как хранилище (контейнер) строк в стиле С. • Выполнять ввод и вывод строк. • Понимать понятие указателя и работа с ним. • Понимать назначение и параметры основных функций для обработки строк: длина строки, копирование, сравнение строк, поиск подстрок <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> • Разработать и отладить программу обработки строк с применением «индуктивного» способа построения алгоритма • Разработать и отладить программу обработки строк с применением функций для строк 	<p>Практическая работа №11. Практическая работа №12. Практическая работа №13.</p>
5	Класс String	4	<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> • Понимать правила инициализации объектов типа string, операции присваивания, сцепления, сравнения, ввода и вывода. • Определять возможность использования синтаксических правил для вызова функций-членов класса string. • Понимать назначение и параметры функций-членов класса string: вставка и замена строк. • Определять возможность использования правил применения обобщенных алгоритмов (reverse, swap) для объектов string. 	<p>Практическая работа №14. Практическая работа №15. Практическая работа №16.</p>

			<p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> • Разработать и отладить программу обработки строк с применением функций для строк. • Разработать и отладить программу обработки строк с применением функций для объектов string. 	
6	Массивы String	4	<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> • Понимать правила объявления и инициализации массивов string, применения функций к элементам массивов. • Понимать правила объявления и инициализации структур, работы с полями структуры; работы с массивами структур <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> • Разработать и отладить программу обработки строк с применением функций для массивов string. • Разработать и отладить программу обработки символьной информации с применением структур и массивов структур. 	<p>Практическая работа №17. Практическая работа №18. Практическая работа №19. Контрольная работа №1.</p>
7	Векторы почти как массивы. Создание, методы.	4	<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> • Понимать правила объявления и инициализации векторов, операции над ячейками вектора и векторами в целом. • Определять возможность использования функции-члены класса вектор push_back(), capacity(). • Понимать назначение и параметры функций-членов класса вектор clear(), swap(). <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> • Разработать и отладить программу обработки последовательности чисел с хранением ее в контейнере вектор. • Разработать и отладить программу обработки последовательности чисел с хранением ее в контейнере вектор, применяя функции-члены класса вектор. 	<p>Практическая работа №20. Практическая работа №21. Практическая работа №22. Практическая работа №23.</p>
8	Простые итераторы, алгоритмы.	4	<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> • Понимать назначение и классификацию итераторов, правила их объявления, диапазоны и допустимые операции. • Определять возможность использования функций класса вектор, возвращающие значения итераторов: begin(), end(). <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> • Разработать и отладить программу обработки последовательности чисел с хранением ее в контейнере вектор, выполняя доступ к элементам вектора с 	<p>Практическая работа №24. Практическая работа №25. Практическая работа №26.</p>

			<p>использованием итераторов.</p> <ul style="list-style-type: none"> • 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. 	
9	Входные, выходные, потоковые итераторы	4	<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> • Понимать назначение, объявление и операции с выходными итераторами. • Понимать особенности выходных потоковых итераторов. • Понимать назначение, объявление и операции со входными итераторами. • Понимать особенности входных потоковых итераторов и итераторов вставки. <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> • 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 (сору()). • 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 (сору(), find()). 	<p>Практическая работа №27. Практическая работа №28. Практическая работа №29.</p>
10	Использование сортировки	4	<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> • Понимать способ оценки вычислительной сложности алгоритмов с помощью O-обозначений, оценки сложности распространенных алгоритмов сортировки. Понимать обозначения параметров и правила использования обобщенного алгоритма sort() из библиотеки STL. • Понимать правила применения функции sort() для векторов встроенных типов данных, а также векторов из структур. <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> • Оценивать вычислительную сложность алгоритма решения задачи, применять функцию sort(). • Применять функцию sort() для упорядочивания значений как в порядке возрастания, так и в порядке убывания. • Написать программу решения задачи. 	<p>Практическая работа №30. Практическая работа №31. Практическая работа №32.</p>
11	Двумерные и многомерные структуры	8	<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> • Понимать правила объявления и инициализации двумерных векторов, доступа к отдельным элементам, особенности применения обобщенных алгоритмов. 	<p>Практическая работа №33. Практическая работа №34. Практическая работа №35 Практическая работа №36.</p>

			<ul style="list-style-type: none"> Понимать правила объявления и инициализации двумерных векторов для хранения и обработки простых изображений, доступа к отдельным элементам, особенности применения обобщенных алгоритмов. <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> Разработать и отладить программу обработки простого изображения с хранением его в двумерном контейнере вектор. Разработать и отладить программу обработки простого изображения с хранением его в двумерном контейнере вектор. 	Практическая работа №37. Практическая работа №38. Практическая работа №39. Контрольная работа №2.
12	Двоичный поиск в массиве	4	<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> Понимать алгоритмы и параметры вычислительной сложности основных алгоритмов поиска в массиве (векторе), области применимости алгоритмов поиска. Понимать правила применения функций двоичного поиска <code>binary_search()</code>, <code>upper_bound()</code> и <code>lower_bound()</code> для векторов. <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> Разработать и отладить программу с применением двоичного поиска в контейнере вектор. 	Практическая работа №40. Практическая работа №41. Практическая работа №42. Практическая работа №43.
13	Двоичный поиск в по ответу	4	<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> Понимать способ двоичного поиска по ответу как средство уменьшения вычислительной сложности алгоритма решения задач; условия его применения. <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> Разработать и отладить программу с применением двоичного поиска по ответу. 	Практическая работа №44. Практическая работа №45. Практическая работа №46. Практическая работа №47.
14	Контейнер список	4	<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> Понимать правила объявления и инициализации списков, операции над ячейками списка и списками в целом. Определять возможность использования функций-членов класса список: <code>push_back()</code>, <code>erase()</code>, <code>front()</code>, <code>insert()</code>. Понимать назначение и параметры функций-членов класса список: <code>splice()</code>. <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> Разработать и отладить программу обработки последовательности чисел с хранением ее в контейнере список. 	Практическая работа №48. Практическая работа №49. Практическая работа №50. Практическая работа №51.
15	Контейнер стек	4	<ul style="list-style-type: none"> Понимать правила объявления и инициализации стеков. Определять возможность использования функций-членов класса стек: <code>push()</code>, <code>pop()</code>, <code>top()</code>, <code>size()</code>. Понимать модель памяти LIFO «последний пришел – первым обслужен», 	Практическая работа №52. Практическая работа №53. Практическая работа №54. Практическая работа №55.

			<p>условия ее применения.</p> <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> • Разработать и отладить программу обработки последовательности чисел с хранением ее в контейнере стек. 	
16	Контейнеры очередь, дек	4	<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> • Понимать правила объявления и инициализации очередей (деков). • Определять возможность использования функций-членов класса очередь: push(), pop(), front(), size(). <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> • Разработать и отладить программу обработки последовательности чисел с хранением ее в контейнере очередь (дек). 	<p>Практическая работа №57. Практическая работа №58. Практическая работа №59. Контрольная работа №3.</p>

Поурочный план

Раз-дел	Тема	Номер урока	Урок	Основные понятия	Планируемые результаты
1	Потоковый ввод и вывод в языке C++	1	Объектно-ориентированный подход к разработке программ	Объектно-ориентированный подход, класс, объект, система программирования, тип данных, проект	<p><i>Понимать, что</i> Объектно-ориентированное программирование – подход для проектирования больших программных систем. Понимать взаимосвязь понятий объекта и класса на примере объектов cin и cout, являющихся представителями классов ostream для потоков входной и выходной символьной информации. Понимать 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>Составлять операторы ввода и вывода числовой и символьной информации для решения простых линейных алгоритмов. 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. Создать проект в среде программирования, включить в него модуль с текстом программы, запустить программу и проанализировать результат.</p>
		2	Стиль написания исходного кода и операторы в языке C++		
		3	Набор и запуск программ		
		4	Решение задач		
2	Циклы и тестовый ввод	5	Функции get(), eof() и fail() объекта cin для более точной работы с символами	Функция, символ, символ-разделитель, специальные символы C++, цикл с предусловием, цикл с постусловием	<p>Понимать 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(). Составлять алгоритмы, завершающие работу при завершении данных во входном потоке. Написать программу решения задачи обработки числовых и символьных последовательностей с применением циклов с пред- и постусловием</p>
		6	Функция put() объекта cout для более точной работы с символами		
		7	Циклы с пред- и постусловием при вводе символьной информации.		
		8	Решение задач		

3	Системы счисления	9	Новые элементы в C++ в представлении целых чисел и в работе с ними	Система счисления, основание системы счисления, алгоритмы преобразования чисел	Понимать 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. Встроенные типы данных для работы с целыми числами, правила записи и операции с целыми в языке C++. Выполнять преобразования целых чисел из внешнего представления (текстовой десятичной записи) во внутреннее и наоборот. 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.
		10	Решение задач		
		11	Решение задач		
		12	Решение задач		
4	Строки в стиле C	13	Строки с завершающим нулем. Представление в памяти, инициализация в C++	Строка, инициализация, указатель, функция	Массив символов как хранилище (контейнер) строк в стиле C. Ввод и вывод строк. Понятие указателя и работа с ним. Назначение и параметры основных функций для обработки строк: длина строки, копирование, сравнение строк, поиск подстрок Разработать и отладить программу обработки строк с применением «индуктивного» способа построения алгоритма Разработать и отладить программу обработки строк с применением функций для строк
		14	Ввод и вывод строк. Работа с указателями. Функции обработки строк		
		15	Решение задач		
		16	Решение задач		
5	Класс String	17	Введение в класс string. Инициализация строк в C++, ввод и вывод.	Класс string, инициализация строк, функция	Понимать правила инициализации объектов типа string, операции присваивания, сцепления, сравнения, ввода и вывода. Синтаксические правила для вызова функций-членов класса string. Назначение и параметры функций-членов класса string: вставка и замена строк. Правила применения обобщенных алгоритмов (reverse, swap) для объектов string. Разработать и отладить программу обработки строк с применением функций для строк. Разработать и
		18	Присваивание, сравнение и вероятные ошибки. Основные функции.		
		19	Функции вставки и замены для строк		
		20	Применение обобщенных алгоритмов к объектам string		

					отладить программу обработки строк с применением функций для объектов string
6	Массивы String	21	Массивы String	Массив, структура, строка	Понимать правила объявления и инициализации массивов string, применения функций к элементам массивов. Правила объявления и инициализации структур, работы с полями структуры; работы с массивами структур Разработать и отладить программу обработки строк с применением функций для массивов string. Разработать и отладить программу обработки символьной информации с применением структур и массивов структур.
		22	Понятие структуры. Массивы структур		
		23	Решение задач		
		24	Контрольная работа 1		
7	Векторы почти как массивы. Создание, методы.	25	Отличие STL от других библиотек. Контейнеры последовательностей: вектор. Инициализация, ввод и вывод.	Библиотека, библиотека STL, контейнер, вектор, функция	Понимать правила объявления и инициализации векторов, операции над ячейками вектора и векторами в целом. Функции-члены класса вектор <code>push_back()</code> , <code>capacity()</code> . Назначение и параметры функций-членов класса вектор <code>clear()</code> , <code>swap()</code> Разработать и отладить программу обработки последовательности чисел с хранением ее в контейнере вектор. Разработать и отладить программу обработки последовательности чисел с хранением ее в контейнере вектор, применяя функции-члены класса вектор.
		26	Решение задач		
		27	Функции-члены класса вектор <code>clear()</code> , <code>swap()</code>		
		28	Решение задач		
8	Простые итераторы, алгоритмы	29	Итераторы – связующий элемент между контейнерами и алгоритмами: определения, классификация.	Итератор, контейнер	Понимать назначение и классификацию итераторов, правила их объявления, диапазоны и допустимые операции. Функции класса вектор, возвращающие значения итераторов: <code>begin()</code> , <code>end()</code> . Классификацию алгоритмов. Назначение и основные параметры обобщенных алгоритмов, правила использования в C++ (<code>adjacent_difference()</code> , <code>copy()</code> , <code>transform()</code>). Разработать и отладить программу обработки последовательности чисел с хранением ее в контейнере вектор, выполняя доступ к элементам вектора с использованием итераторов. Разработать
		30	Операции над итераторами		
		31	Обобщенные алгоритмы: определения, классификация, примеры		
		32	Решение задач		

					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
9	Входные, выходные, потоковые итераторы	33	Обратный итератор. Выходные и потоковые итераторы	Обратный итератор, входной итератор, выходной итератор, потоковый итератор, итератор вставки	Понимать назначение, объявление и операции с выходными итераторами. Особенности выходных потоковых итераторов. Назначение, объявление и операции со входными итераторами. Особенности входных потоковых итераторов и итераторов вставки. 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>). 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>).
		34	Решение задач		
		35	Входные и потоковые итераторы. Итераторы вставки		
		36	Решение задач		
10	Использование сортировки	37	Модели вычислительной сложности алгоритмов – O-обозначения. Алгоритмы сортировки и их сложность	Сложность алгоритма, сортировка	Понимать 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. Правила применения функции <code>sort()</code> для векторов встроенных типов данных, а также векторов из структур Оценивать вычислительную сложность алгоритма решения задачи, применять функцию <code>sort()</code> . Применять функцию <code>sort()</code> для упорядочивания значений как в порядке
		38	Сортировки векторов структур по разным полям		
		39	Решение задач		
		40	Решение задач		

					возрастания, так и в порядке убывания. Написать программу решения задачи.
11	Двумерные и многомерные структуры	41	Особенности применения контейнеров STL для хранения и обработки двумерных данных	Двумерные данные, контейнер STL, способ хранения изображения в двумерном контейнере	<p>Понимать правила объявления и инициализации двумерных векторов, доступа к отдельным элементам, особенности применения обобщенных алгоритмов. Правила объявления и инициализации двумерных векторов для хранения и обработки простых изображений, доступа к отдельным элементам, особенности применения обобщенных алгоритмов</p> <p>Разработать и отладить программу обработки последовательности чисел с хранением ее в двумерном контейнере вектор. Разработать и отладить программу обработки простого изображения с хранением его в двумерном контейнере вектор</p>
		42	Решение задач с обработкой двумерных числовых и символьных данных.		
		43	Решение задач		
		44	Решение задач		
		45	Особенности применения контейнеров STL для хранения и обработки простых изображений		
		46	Решение задач с обработкой простых изображений		
		47	Решение задач		
		48	Контрольная работа 2		
12	Двоичный поиск в массиве	49	Задача поиска информации; линейный и логарифмический (двоичный) поиск в массиве (векторе): алгоритмы и сравнение	Линейный поиск, двоичный поиск	<p>Понимать алгоритмы и параметры вычислительной сложности основных алгоритмов поиска в массиве (векторе), области применимости алгоритмов поиска. Правила применения функций двоичного поиска <code>binary_search()</code>, <code>upper_bound()</code> и <code>lower_bound()</code> для векторов.</p> <p>Разработать и отладить программу с применением двоичного поиска в контейнере вектор.</p>
		50	Решение задач		
		51	Обобщенные алгоритмы STL, связанные с двоичным поиском		
		52	Решение задач		
13	Двоичный поиск по ответу	53	Задачи двоичного поиска по ответу	Двоичный поиск	<p>Понимать структуру алгоритма двоичного поиска по ответу. Способ двоичного поиска по ответу как средство уменьшения вычислительной сложности алгоритма решения задач; условия его применения</p> <p>Разработать и отладить программу с применением двоичного поиска по ответу.</p>
		54	Решение задач		
		55	Решение задач методом двоичного поиска по ответу		
		56	Решение задач		

14	Контейнер список	57	Контейнеры последовательностей: список. Инициализация, ввод и вывод.	Контейнер, список, итератор	<p>Понимать 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>. Назначение и параметры функций-членов класса список: <code>splice()</code>.</p> <p>Разработать и отладить программу обработки последовательности чисел с хранением ее в контейнере список.</p>
		58	Решение задач		
		59	Итераторы входные, выходные и однонаправленные		
		60	Решение задач		
15	Контейнер стек	61	Контейнеры последовательностей: стек. Инициализация, ввод и вывод.	Контейнер, список, итератор	<p>Понимать правила объявления и инициализации стеков. Функции-члены класса стек: <code>push()</code>, <code>pop()</code>, <code>top()</code>, <code>size()</code>. Модель памяти LIFO «последний пришел – первым обслужен», условия ее применения</p> <p>Разработать и отладить программу обработки последовательности чисел с хранением ее в контейнере стек</p>
		62	Решение задач		
		63	Вычислительные задачи с обработкой данных в обратном порядке «последний пришел – первым обслужен».		
		64	Решение задач		
16	Контейнеры очередь, дек	65	Контейнеры последовательностей: очередь и дек. Инициализация, ввод и вывод	Контейнер, список, итератор	<p>Понимать правила объявления и инициализации стеков. Функции-члены класса стек: <code>push()</code>, <code>pop()</code>, <code>top()</code>, <code>size()</code>. Модель памяти LIFO «последний пришел – первым обслужен», условия ее применения</p> <p>Разработать и отладить программу обработки последовательности чисел с хранением ее в контейнере очередь, дек</p>
		66	Вычислительные задачи с обработкой данных в порядке поступления «первый пришел – первым обслужен»		
		67	Решение задач		
		68	Контрольная работа 3		

Дистанционное обучение

Данный учебный курс использует ресурсы Федерального государственного бюджетного образовательного учреждения высшего образования «Ижевский государственный технический университет имени М.Т. Калашникова». В частности сайт new.moodle.cs.istu.ru, с помощью которого осуществляется контроль правильности решения учащимися заданий на уроке, домашних заданий, практических работ и контрольных работ. Эта проверка осуществляется автоматически, разработанной ВУЗом тестовой системой

.

Учебно-методическое обеспечение и материально-техническое обеспечение учебного процесса

Для отражения количественных показателей в требованиях используется следующая система символических обозначений:

- **Д** – демонстрационный экземпляр (1 экз., кроме специально оговоренных случаев), буквой **Д** также обозначается все оборудование, необходимое в единственном экземпляре;
- **К** – полный комплект (исходя из реальной наполняемости класса), для школ с наполняемостью классов свыше 25 человек при комплектовании кабинета средствами ИКТ рекомендуется исходить из 15 рабочих мест учащихся;
- **Ф** – комплект для фронтальной работы (примерно в два раза меньше, чем полный комплект, то есть не менее 1 экз. на двух учащихся),
- **П** – комплект, необходимый для практической работы в группах, насчитывающих по несколько учащихся (5-7 экз.).

№	Наименования объектов и средств материально-технического обеспечения	Необходимое количество
1.	БИБЛИОТЕЧНЫЙ ФОНД (КНИГОПЕЧАТНАЯ ПРОДУКЦИЯ)	
1.1	Стандарт основного общего образования по информатике	Д
1.2	Примерная программа основного общего образования по информатике	Д
1.3	Методические пособия для учителя (рекомендации к проведению уроков)	Д
1.4	Учебник по информатике для основной школы	К
1.5	Рабочая тетрадь по информатике	К
1.6	Научная, научно-популярная литература, периодические издания	П
1.7	Справочные пособия (энциклопедии и т.п.)	П
1.8	Дидактические материалы по всем курсам	Ф
2.	ИНФОРМАЦИОННО-КОММУНИКАТИВНЫЕ СРЕДСТВА	
	<i>Программные средства</i>	
2.1	Операционная система	К
2.2	Файловый менеджер (в составе операционной системы или др.).	К
2.3	Почтовый клиент (входит в состав операционных систем или др.).	К
2.4	Программа для организации общения и групповой работы с использованием компьютерных сетей.	К
2.5	Программная оболочка для организации единого информационного пространства школы, включая возможность размещения работ учащихся и работу с цифровыми ресурсами	
2.6	Программное обеспечение для организации управляемого коллективного и безопасного доступа в Internet. Брандмауэр и НТТР-прокси сервер.	Д
2.7	Антивирусная программа	К
2.8	Программа-архиватор	К
2.9	Система оптического распознавания текста для русского, национального и изучаемых иностранных языков	К
2.10	Программа для записи CD и DVD дисков	К
2.11	Комплект общеупотребимых программ, включающий: текстовый редактор, программу разработки презентаций, электронные таблицы.	К
2.12	Звуковой редактор.	К
2.13	Программа для организации аудиоархивов.	К
2.14	Редакторы векторной и растровой графики.	К
2.15	Программа для просмотра статических изображений.	К
2.16	Мультимедиа проигрыватель	К

№	Наименования объектов и средств материально-технического обеспечения	Необходимое количество
2.17	Программа для проведения видеомонтажа и сжатия видеофайлов	П
2.18	Редактор Web-страниц.	К
2.19	Браузер	К
2.20	Система управления базами данных, обеспечивающая необходимые требования.	К
2.21	Геоинформационная система, позволяющая реализовать требования стандарта по предметам, использующим картографический материал.	К
2.22	Система автоматизированного проектирования.	К
2.23	Виртуальные компьютерные лаборатории по основным разделам курсов математики и естественных наук.	К
2.24	Интегрированные творческие среды.	К
2.25	Программа-переводчик, многоязычный электронный словарь.	К
2.26	Система программирования.	К
2.27	Клавиатурный тренажер.	К
2.28	Программное обеспечение для работы цифровой измерительной лаборатории, статистической обработки и визуализации данных	К
2.29	Программное обеспечение для работы цифровой лаборатории конструирования и робототехники	К
2.30	Программное обеспечение для работы цифрового микроскопа	К
2.31	Коллекции цифровых образовательных ресурсов по различным учебным предметам	К
3.	ЭКРАННО-ЗВУКОВЫЕ ПОСОБИЯ	
3.1	Комплекты презентационных слайдов по всем разделам курсов	Д
4.	ТЕХНИЧЕСКИЕ СРЕДСТВА ОБУЧЕНИЯ (СРЕДСТВА ИКТ)	
4.1	Экран (на штативе или настенный)	Д
4.2	Мультимедиа проектор	Д
4.3	Персональный компьютер – рабочее место учителя	Д
4.4	Персональный компьютер – рабочее место ученика	К
4.5	Сервер	Д
4.6	Комплект сетевого оборудования	Д
4.7	Комплект оборудования для подключения к сети Интернет	Д
4.8	Дискеты	
4.9	Диск для записи (CD-R или CD-RW)	
5.	МЕБЕЛЬ	
5.1	Компьютерный стол	Д/Ф
5.2	Аудиторная доска для письма фломастером с магнитной поверхностью	Д
5.3	Шкафы для хранения оборудования	Д

Программно-методическое обеспечение.

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.

Практические и контрольные работы

Практические и контрольные работы представлены на сайте new.moodle.cs.istu.ru