муниципальное бюджетное общеобразовательное учреждение

«Дегтярская средняя общеобразовательная школа»

Немецкий национальный район

Алтайский край

УТВЕРЖЕНО

директор школы

\_\_\_\_\_\_\_\_\_ Е.М. Бардаева

Приказ № 34

от "31" мая 2022 г.

**РАБОЧАЯ ПРОГРАММА**

«Программирование на Си»

внеурочной деятельности технической направленности

образовательного центра «Точка роста»

для 10-11 классов среднего общего образования

на 2022-2023 учебный год

**возраст учащихся – 15-17 лет**

**срок реализации программы - 2 года**

Составитель: Удовик Александр Николаевич

Дегтярка 2022

**1.Пояснительная записка**

Курс «Программирование на Си» для учащихся 10 - 11 классов обучающихся в центре образования «Точка роста» МБОУ «Дегтярская СОШ» разработан на основе учебного пособия для общеобразовательных организаций К.Ю. Полякова «Программирование. Python. C++» в 4-х частях. Включает 138 часов аудиторных занятий и (при возможности) самостоятельную работу учащихся, из них 70 часов в 10 классе и 68 часов в 11 классе.

Предполагается, что учащиеся предварительно изучили программирование на языке Паскаль, имеют навыки основ алгоритмирования.

Электронный материал авторских разработок доктора технических наук К.Ю. Полякова для сопровождения курса представлен в полном объёме на сайте автора <http://kpolyakov.spb.ru/school/c.htm> и включает презентации, решение задач и сборник программ в среде [Dev-C++](http://kpolyakov.spb.ru/school/c/devcpp.htm).

Предметом изучения являются принципы и методы программирования в среде бесплатно распространяемой оболочки [Dev-C++](http://kpolyakov.spb.ru/school/c/devcpp.htm) с открытым исходным кодом, включающая компилятор [GCC](http://gcc.gnu.org/).

Сотрудник компании Bell Labs Деннис Ритчи в начале 1970-х годов разработал один из самых востребованных языков программирования и остаётся на лидирующих позициях до настоящего времени - язык С.

Независимый от процессора язык С работает напрямую с аппаратурой, заменяя чрезвычайно сложное программирование в машинных кодах и на языке ассемблера.

Для моделирования сложных систем, в начале 1980-х годов Бьёрн Страуструп, сотрудник той же компании Bell Labs, дополнил язык С новыми возможностями и создал язык C++, который стал одним из самых популярных языков программирования. Все программы на «чистом» С будут работать и в C++.

Одно из достоинств современного языка C++ — обширная стандартная библиотека: набор готовых подпрограмм для решения часто встречающихся задач (например, для обработки символьных строк).

Целесообразность изучения данного курса определяется тем, что язык С++ – основной язык для создания игр и операционных систем. Программы на C++ работают значительно быстрее, чем на других языках, например Python, предназначенный более для решения небольших задач, в которых скорость выполнения не очень важна и применяемый для разработки сайтов, решения задач биоинформатики и обработки больших данных. Но программы, созданные на Си сложнее писать и отлаживать. «…Умение программировать на разных языках – это обязательное требование к разработчику программного обеспечения»[1]

**Цели курса:**

* ознакомление учащихся с принципами и методами современного объектно – ориентированного программирования высокого уровня;
* формирование знаний, умений и навыков в области программирования на языке С++;
* развитие алгоритмических способностей учащихся по программированию.

**Задачи курса:** научить учащихся

* создавать программы для решения прикладных задач разного уровня в среде Dev-C++ с открытым исходным кодом, включающая компилятор GCC*;*
* понимать основные принципы объектно-ориентированного программирования.

Данный курс имеет выраженную практическую направленность, которая и определяет логику построения материала учебных занятий. Основной формой обучения является практикум. Для работы необходим персональный компьютер (один на каждого ученика) и среда Dev-C++ с открытым исходным кодом, включающая компилятор GCC. Для выполнения большинства заданий достаточно использовать среду, бесплатно загруженную с Web-сайта **http://www.bloodshed.net/dev/devcpp.html**

Для поддержки курса К.Ю. Поляковым разработано электронное учебное пособие в формате PDF, которое содержит теоретический материал, задания для выполнения практических работ, презентации, которое используется во время уроков для самостоятельной работы и в качестве справочника. Это позволяет успешно организовывать занятия в группах, в которых есть ученики с разным темпом усвоения материала.

Знания, полученные при изучении элективного курса «Программирование на Си++», учащиеся могут применить для решения прикладных задач разного рода, повышения качества подготовки к успешной сдаче ЕГЭ. Полученные знания и умения являются основой для последующего изучения программирования в высших профессиональных образовательных учреждениях.

**2.Требования к результатам освоения обучающимися программы элективного курса «Программирование на СИ++»**

В рамках данного курса учащиеся получают следующие знания и умения:

### Уметь

* самостоятельно определять цели деятельности и составлять планы деятельности; самостоятельно осуществлять, контролировать и корректировать деятельность; использовать все возможные ресурсы для достижения поставленных целей и реализации планов деятельности; выбирать успешные стратегии в различных ситуациях;
* продуктивно общаться и взаимодействовать в процессе совместной деятельности, учитывать позиции других участников деятельности, эффективно разрешать конфликты;
* владеть навыками познавательной, учебно-исследовательской и проектной деятельности, навыками разрешения проблем; осуществлять самостоятельный поиск методов решения практических задач;
* вести самостоятельную информационно-познавательную деятельность, включая ориентирование в различных источниках информации, критически оценивать и интерпретировать информацию, получаемую из различных источников;
* применять навыки алгоритмического мышления и понимать необходимость применения формального описания алгоритмов;
* использовать понятие сложности алгоритма, применять основные алгоритмы обработки числовой и текстовой информации, алгоритмы поиска и сортировки;
* владеть приёмами написания программы в среде Dev-C++ для решения прикладных задач с использованием основных конструкций программирования; осуществлять отладку таких программ; тестировать и оптимизировать готовые прикладные компьютерные программы;
* применять базовые типы данных и структуры данных; использовать основные управляющие конструкции;
* осуществлять формализацию задач и производить документирование программ.

### Знать

* алгоритмические конструкции, лежащие в основе программирования в среде Dev-C++ с открытым исходным кодом, включающей компилятор GCC;
* синтаксис операторов и их применение в решении задач;
* основные понятия объектно-ориентированного программирования (объект, свойство, метод, наследование).

**Формы подведения итогов**

Текущий контроль уровня усвоения материала осуществляется по результатам выполнения учащимися практических заданий. В конце разделов курса каждый учащийся выполняет индивидуальный проект в качестве зачетной работы.

**3.Содержание программы курса**

***10 класс***

**Программирование на языке С++ (часть 1, глава 2).**

**Основные управляющие конструкции.– 35 ч.**

*Тема 1.1. Первые программы.*

Повторение: понятие алгоритма, свойства алгоритма, исполнитель алгоритма (виды, основные характеристики), способы записи алгоритма.

Структура программы. Операторы ввода-вывода. Переменные.

Ключевые слова: компилятор, препроцессор, раздельная компиляция, вывод данных, поток вывода, поток ввода, пространство имен, символьная строка.

*Тема 1.2. Диалоговые программы.*

Диалоговые программы. Переменные и их типы. Сумма чисел.

Ключевые слова: ввод данных, переменная, объявление переменной, входной поток.

*Тема 1.3. Компьютерная графика.*

Библиотека TX Library. Управляем пикселями. Линии и фигуры. Замкнутые фигуры.

Ключевые слова: окно, координаты, оси координат, пиксель, цвет контура, цвет заливки, код цвета, прозрачный цвет.

*Тема 1.4. Процедуры.*

Длинная программа. Рефакторинг. Процедуры с параметрами.

Ключевые слова: подпрограмма, процедура, рефакторинг, аргументы, параметры, базовая точка.

*Тема 1.5. Обработка целых чисел.*

Предельные значения чисел. Различие между вещественными и целыми числами. Дискретное представление чисел. Программное повышение точности вычислений.

Хранение в памяти целых чисел: со знаком, без знака. Повторение: получение дополнительного кода числа.

Арифметические операции с целыми числами: сложение и вычитание, умножение и деление. Операция сравнения чисел. Повторение: поразрядные логические операции, сдвиги.

Ключевые слова: арифметические выражения, частное, остаток, форматный вывод, случайные числа, зерно.

*Тема 1.6. Обработка вещественных чисел.*

Повторение: хранение в памяти вещественных чисел. Нормализованное представление вещественных чисел в компьютере.

Арифметические операции с вещественными числами: сложение и вычитание, умножение и деление.

Ключевые слова: вещественное число, научный формат, мантисса, форматный вывод, округление.

*Тема 1.7.* *Ветвления.*

Условный оператор. Полная и неполная формы записи условного оператора. Сложные условия в условном операторе и их применение в написании программ. Множественный выбор - переключатель switch.

Ключевые слова: условный оператор, полная форма, неполная форма, составной оператор, вложенный условный оператор, логические переменные.

*Тема 1.8. Циклы.*

Понятие циклического алгоритма. Циклы с предусловием. Циклы с постусловием. Циклы с переменными (счетчиком). Вложенные циклы.

Ключевые слова: цикл с предусловием, цикл с постусловием, цикл по переменной.

*Тема 1.9*. *Анимация*

Принципы анимации. Рисуем шарик. Начальное положение. Анимация движения. Обработка нажатия клавиши. Ключевые слова: анимация, процедура, пауза, нажатие клавиши.

*Учащиеся должны знать:*

- понятие и свойства алгоритма, способы записи, типы алгоритмических структур;

- структуру программы, операторы ввода-вывода, переменные;

- типы переменных, стандартные функции;

- условный оператор, циклы;

- принципы анимации.

*Учащиеся должны уметь:*

- применять полученные знания к решению задач;

- применять операторы ввода-вывода в программе при создании диалога с пользователем в режиме компиляции;

- соблюдать структуру программы;

- применять стандартные функции к решению задач;

- владеть приемами применения типов переменных;

- составлять программы с использованием конструкций ветвления и циклов;

-применять принципы анимации при создании анимированных объектов.

### *Предметные результаты*

1. *Владение универсальным языком программирования высокого уровня (по выбору), представлениями о базовых типах данных и структурах данных; умением использовать основные управляющие конструкции.*
2. *Владение умением понимать программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня; знанием основных конструкций программирования; умением анализировать алгоритмы с использованием таблиц.*
3. *Владение навыками алгоритмического мышления и понимание необходимости формального описания алгоритмов.*
4. *Овладение понятием сложности алгоритма, знание основных алгоритмов обработки числовой и текстовой информации.*

**Программирование на языке С++ (часть 2, глава 2)**

**Процедуры и функции — элементы структуризации программ.**

**Массив — фундаментальная структура данных. – 35 ч.**

*Тема 2.1. Процедуры.*

Вспомогательные алгоритмы. Процедуры. Запись процедуры на языке программирования. Процедура с параметрами.

Ключевые слова: процедура, интерфейс, реализация, параметр, локальная переменная, глобальная переменная, область видимости, передача по ссылке.

*Тема 2.2. Рекурсия.*

Рекурсия. Примеры рекурсий в математике и литературе. Рекурсивная процедура (функция). Использование рекурсий при написании программ.

Ключевые слова: рекурсивная процедура, фрактал, базовые объекты, условия остановки, анимация

*Тема 2.3. Функции.*

Функция. Виды функций. Запись функций на языке программирования.

Ключевые слова: функция, параметры, вызов функции, результат функции, рекурсивная функция.

*Тема 2.4. Символьные строки.*

Символьная строка. Операции со строками: сцепление, удаление, копирование элементов. Функции поиска подстроки. Преобразование из строки в число и наоборот. Применение строковых данных в процедурах и функциях. Рекурсивный перебор символов. Сравнение и сортировка строк.

Ключевые слова: символьная строка, длина строки, сцепление строк, выход за границы строки, подстрока, удаление символов, вставка символов, поиск подстроки, замена подстроки, преобразование типов.

*Тема 2.5. Массивы.*

Массивы в С++. Обращение к элементу массива. Перебор элементов массива. Вывод и ввод массива. Заполнение массива случайными числами. Алгоритмы обработки массивов.

Ключевые слова: массив, индекс элемента, значение элемента, константа, заполнение массива, вывод массива, ввод массива.

*Тема 2.6.*  *Используем массивы.*

Используем массивы. Игра «Стрельба по тарелкам».

Ключевые слова: массив, константа, инициализация, глобальные переменные, обработчик события.

*Тема 2.7. Матрицы*.

Матрица. Основные операции с матрицами: объявление, заполнение, вывод на экран. Обработка элементов матрицы.

Ключевые слова: матрица, строка, столбец, перебор элементов, вложенный цикл, квадратная матрица, главная диагональ, побочная диагональ, перестановка строк.*Тема 2.8. Тема 2.8. Системы управления версиями. Основные приемы работы с Git.*

Системы управления версиями. Основные приемы работы с Git. Операции с файлами. Работа с удалённым архивом.

*Учащиеся должны знать:*

* применение вспомогательных алгоритмов: процедур и функций.
* применений рекурсивных алгоритмов к решению задач;
* понятие массива; основные операции с массивами: объявление, заполнение, вывод на экран;
* методы сортировки массивов: метод пузырька, метод выбора, быстрая сортировка;
* понятие символьной строки и операции со строками: объединение, удаление, копирование элементов строк;
* понятие матрицы и основные операции с матрицами: объявление, заполнение, вывод на экран, обработка элементов матрицы.

*Учащиеся должны уметь:*

* применять полученные знания к решению задач;
* осуществлять поиск и обработку элементов в массиве, определять минимальный и максимальный элементы массива, осуществлять реверс массива, сдвиг элементов массива, отбор элементов массива по условиям, сортировку в массивах, двоичный поиск в массиве;
* осуществлять операции со строками: объединение, удаление, копирование элементов, функции поиска подстроки, преобразование из строки в число и наоборот;
* применять строковые данные в процедурах и функциях;
* осуществлять рекурсивный перебор символов, сравнивать и сортировать строки;
* производить основные операции с матрицами: объявление, заполнение, вывод на экран, обработку элементов матрицы.

### *Предметные результаты*

1. *Владение универсальным языком программирования высокого уровня (по выбору), представлениями о базовых типах данных и структурах данных; умением использовать основные управляющие конструкции.*
2. *Владение умением понимать программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня; знанием основных конструкций программирования; умением анализировать алгоритмы с использованием таблиц.*
3. *Владение навыками алгоритмического мышления и понимание необходимости формального описания алгоритмов.*
4. *Овладение понятием сложности алгоритма, знание основных алгоритмов обработки числовой и текстовой информации, алгоритмов поиска и сортировки.*

***11 класс***

**Программирование на языке С++ (часть 3, глава 2).**

**Динамические структуры данных. – 35 ч.**

*Тема 3.1. Простые алгоритмы сортировки.*

Сортировка в массивах. Методы сортировки: метод пузырька, метод выбора, сортировка вставками. Массивы в подпрограммах.

Ключевые слова: сортировка, метод пузырька, сортировка вставками, перестановка элементов, сложность алгоритма.

*Тема 3.2. Быстрые алгоритмы сортировки и поиска.*

Сортировка в массивах. Методы сортировки: сортировка слиянием, быстрая сортировка. Стандартная сортировка на языке С++. Двоичный поиск.

Ключевые слова: сортировка слиянием, быстрая сортировка, двоичный поиск, лямбда-функция

*Тема 3.3. Обработка файлов.*

Понятие файла. Типы файлов. Этапы работы с файлами: открытие файла, запись в файл, удаление записей из файла, чтение из файла, закрытие файла. Обработка массивов, записанных в файле. Обработка строк, записанных в файле. Обработка смешанных данных, записанных в файле.

Ключевые слова: файл, файловый поток, открытие файла, закрытие файла, чтение из файла, запись в файл, конец файла, аргументы командной строки.

*Тема 3.4. Целочисленные алгоритмы.*

Целочисленные алгоритмы. Решето Эратосфена. Длинные числа.

Ключевые слова: решето Эратосфена, длинные числа, разряды, перенос.

*Тема 3.5. Динамические массивы и словари.*

Понятие динамического массива. Объявление и заполнение динамического массива. Использование динамических массивов в подпрограммах. Расширение массива. Принципы работы динамического массива. Тип vector из библиотеки STL. Итераторы. Словари. Перебор элементов словаря.

Ключевые слова: динамический массив, указатель, контейнер, вектор, словарь, ключ, значение, перебор элементов, сортировка по ключу, итератор.

*Тема 3.6. Структуры.*

Понятие структуры. Объявление структур. Обращение к полю структуры. Работа с файлами. Сортировка.

Ключевые слова: структура, поле, точечная запись, сортировка, ключ.

*Тема 3.7.* *Стеки, очереди, деки*

Понятие стека. Использование динамического массива для создания стека. Вычисление арифметических выражений. Скобочные выражения. Очередь и деки как разновидности линейного списка.

Ключевые слова: стек, очередь, дек.

*Тема 3.8.* *Деревья.*

Дерево. Двоичное дерево. Дерево поиска. Обход двоичного дерева. Вычисление арифметических выражений. Использование связанных структур. Хранение двоичного дерева в массиве.

Ключевые слова: дерево, двоичное дерево, обход в глубину, обход в ширину, дерево поиска, ключ, хранение в массиве, модуль, проект, заголовочный файл.

*Тема 3.9.* *Графы*

Понятие графа. «Жадные» алгоритмы. Алгоритм Прима-Крускала. Поиск кратчайших путей в графе. Алгоритм Дейкстры. Алгоритм Флойда-Уоршелла.

Ключевые слова: жадный алгоритм, остовное дерево, задача коммивояжёра, Гамильтонов цикл, полный перебор, рекурсия, хвостовая рекурсия, случайный поиск, глобальные переменные, структуры.

*Тема 3.10.* *Динамическое программирование.*

Динамическое программирование. Поиск оптимального решения. Количество решений.

Ключевые слова: динамическое программирование, перебор вариантов, мемоизация, оптимальная программа, редактирование строк, расстояние Левенштейна.

*Учащиеся должны знать:*

* понятие файла, типы, этапы работы;
* алгоритмы работы с файлами;
* алгоритм поиска простых чисел с помощью «решета Эратосфена»;
* понятие «длинного числа», принципы хранения и выполнения операций с «длинными» числами;
* понятие структуры (записи, множества), основные операции со структурами;
* понятие «дерево» и области применения этой структуры данных;
* понятия «граф», «узел», «ребро»;
* простые алгоритмы на графах;
* принципы динамического программирования.

*Учащиеся должны уметь:*

*-* осуществлять обработку массивов строк, смешанных данных, записанных в файле;

* использовать текстовый и типизированные файлы;
* программировать простые операции с «длинными» числами;
* использовать различные структуры, грамотно выбирать структуру для конкретной задачи;
* программировать простые алгоритмы на графах;
* программировать алгоритмы, использующие динамическое программирование.

### *Предметные результаты*

1. *Владение универсальным языком программирования высокого уровня (по выбору), представлениями о базовых типах данных и структурах данных; умением использовать основные управляющие конструкции.*
2. *Владение умением понимать программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня; знанием основных конструкций программирования; умением анализировать алгоритмы с использованием таблиц.*
3. *Владение навыками алгоритмического мышления и понимание необходимости формального описания алгоритмов.*
4. *Овладение понятием сложности алгоритма, знание основных алгоритмов обработки числовой и текстовой информации, алгоритмов поиска и сортировки.*

Динамические структуры данных. Связанный список. Создание элемента списка. Добавление узла. Проход по списку. Поиск узла в списке. Алфавитно-частотный словарь. Удаление узла. Барьеры. Двусвязный список. Операции с двусвязным списком. Циклические списки.

Стек. Реализация стека с помощью массива. Реализация стека с помощью списка. Системный стек в программах. Очередь. Реализация очереди с помощью массива. Реализация очереди с помощью списка. Дек.

Что такое деревья? Реализация двоичных деревьев в языке Си. Поиск с помощью дерева. Разбор арифметического выражения. Дерево игр.

Основные понятия. Задача Прима-Крускала. Кратчайший путь. Оптимальное размещение Задача коммивояжера. Задача о паросочетаниях.

**Программирование на языке С++ (часть 4, глава 2). Объектно-ориентированное программирование -33 ч.**

*Тема 4.1. Классы и объекты в С++.*

Классы и объекты в С++: новая задача и ее анализ. Класс CMap. Пишем свой конструктор. Рефакторинг. Рисуем карты.

Ключевые слова: класс, объект, экземпляр, модификатор доступа, конструктор, инициализация, метод, рефакторинг.

*Тема 4.2. Программа с классами (практикум).*

Класс CCar. Рефакторинг. Основная программа. Разбиение на модули. Основная программа. Разбиение на модули.

Ключевые слова: класс, объект, ссылка, конструктор, метод, рефакторинг.

*Тема 4.3. Инкапсуляция.*

Объект защищает свои данные. Изменение внутреннего устройства. Свойство «только для чтения». Свойства в C#.

Ключевые слова: инкапсуляция, поле, метод, модификатор доступа, интерфейс, свойство.

*Тема 4.4. Наследование.*

Моделирование жизни в океане. Иерархия классов. Базовый класс. Абстрактный класс. Защищенные поля и методы (protected). Неподвижные объекты. Подвижные объекты. Рыбы. Хищники. Вспомогательные процедуры и функции. Основная программа.

Ключевые слова: класс, иерархия классов, базовый класс, наследование, абстрактный класс, виртуальный метод, защищенные поля, утечка памяти.

*Тема 4.5. Полиморфизм.*

Полиморфизм в действии. Класс Океан. Деструктор.

Ключевые слова: базовый класс, класс-наследник, виртуальный метод, полиморфизм, позднее связывание, таблица виртуальных методов, деструктор.

*Тема 4.6. Взаимодействие объектов.*

Столкновения. Изменение базового класса. Изменение других классов. Умные указатели.

Ключевые слова: столкновение, «мертвые» объекты, итератор, преобразование типов.

*Тема 4.7. Простая программа на С#.*

RAD – среды для разработки программ. Язык С# и среда .NET. Проект С#. Свойства объектов. Обработчики событий.

Ключевые слова: RAD – среда, среда NET., язык С#, виртуальная машина, проект, форма, свойство, событие, обработчик события, компонент, статистический класс.

*Тема 4.8. Использование компонентов.*

Программы для просмотра рисунков. Ввод и вывод данных. Обработка ошибок.

Ключевые слова: компонент, свойство, событие, обработчик события, родительский объект, дочерний объект, поле ввода, метка, статический метод, обработка ошибок, исключение.

*Тема 4.9. Создание новых классов.*

Разработка программы для вычисления арифметических выражений: модель, новый класс. Методы класса. Вычисление арифметических выражений: представление. Добавляем новый компонент. Изменение поведения компонента.

Ключевые слова: модуль, модель, представление, класс, компонент, статический класс, методы класса, свойство.

*Учащиеся должны знать:*

* понятия «объект», «свойства объектов»;
* понятие «классы», «иерархия классов их классификацию»;
* понятие «базовый класс, классы-наследники, модульность»;
* принципы работы в RAD-средах;
* сообщения между объектами.

*Учащиеся должны уметь:*

* составлять простые программы в RAD-средах;
* оценивать вычислительную сложность изученных алгоритмов.

### *Предметные результаты*

1. *Владение стандартными приёмами написания на алгоритмическом языке программы для решения стандартной задачи с использованием основных конструкций программирования и отладки таких программ; использование готовых прикладных компьютерных программ по выбранной специализации.*

Понятие объектно-ориентированного программирования. Объекты. Классы.

Создание объектов в программе.

Скрытие внутреннего устройства.

Иерархия классов. Классификация. Иерархия логических элементов. Базовый класс. Классы-наследники. Модульность. Сообщения между объектами.

Программы с графическим интерфейсом. RAD-среды для разработки программ.

Основы программирования в RAD-средах. Общий подход. Создание простейшей программы. Свойства объектов. Обработчики событий.

Программа с компонентами. Ввод и вывод данных. Обработка ошибок.

Модификация компонентов.

Разработка программы вычисления арифметического выражения, записанного в символьной строке.

**4.Тематическое планирование с указанием количества часов на освоение каждой темы по 2 часа в неделю в 10 и 11 классах**

**(всего 138 часов)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Наименование разделов и тем | Количество часов | Виды занятий | |
| Теория | Практика |
| **10 класс** | | | | |
| **1.** | **Программирование на языке С++ (часть 1, глава 2) Основные управляющие конструкции.** | **35** | **9** | **26** |
| 1.1 | Первые программы. | 2 | 1 | 1 |
| 1.2 | Диалоговые программы. | 3 | 1 | 2 |
| 1.3 | Компьютерная графика. | 2 | 1 | 1 |
| 1.4 | Процедуры. | 3 | 1 | 2 |
| 1.5 | Обработка целых чисел. | 4 | 1 | 3 |
| 1.6 | Обработка вещественных чисел. | 4 | 1 | 3 |
| 1.7 | Ветвления. | 7 | 1 | 6 |
| 1.8 | Циклы. | 6 | 1 | 5 |
| 1.9 | Анимация. | 4 | 1 | 3 |
| **2.** | **Программирование на языке С++ (часть 2, глава 2) Процедуры и функции — элементы структуризации программ. Массив — фундаментальная структура данных.** | **35** | **10** | **25** |
| 2.1 | Процедуры. | 4 | 1 | 3 |
| 2.2 | Рекурсия. | 3 | 1 | 2 |
| 2.3 | Функции. | 3 | 1 | 2 |
| 2.4 | Символьные строки. | 10 | 2 | 8 |
| 2.5 | Массивы. | 2 | 1 | 1 |
| 2.6 | Используем массивы. | 3 | 1 | 2 |
| 2.7 | Матрицы. | 7 | 1 | 6 |
| 2.8 | Системы управления версиями. Основные приемы работы с Git. | 2 | 1 | 1 |
|  | Резерв | 1 | 1 | 0 |
| **Итого** | | **70** | **19** | **51** |
| **11 класс** | | | | |
| **3.** | **Программирование на языке С++ (часть 3, глава 2). Динамические структуры данных.** | **35** | **8** | **27** |
| 3.1 | Простые алгоритмы сортировки. | 3 | 1 | 2 |
| 3.2 | Быстрые алгоритмы сортировки и поиска. | 4 | 1 | 3 |
| 3.3 | Обработка файлов. | 3 | 1 | 2 |
| 3.4 | Целочисленные алгоритмы. | 2 | 0 | 2 |
| 3.5 | Динамические массивы и словари. | 4 | 1 | 3 |
| 3.6 | Структуры. | 3 | 1 | 2 |
| 3.7 | Стеки, очереди, деки. | 3 | 0 | 3 |
| 3.8 | Деревья. | 5 | 1 | 4 |
| 3.9 | Графы. | 5 | 1 | 4 |
| 3.10 | Динамическое программирование. | 3 | 1 | 2 |
| **4.** | **Программирование на языке С++(часть 4, глава 2). Объектно-ориентированное программирование.** | **33** | **3** | **30** |
| 4.1 | Классы и объекты в С++. | 3 | 1 | 2 |
| 4.2 | Программа с классами (практикум). | 3 | 0 | 3 |
| 4.3 | Инкапсуляция. | 3 | 1 | 2 |
| 4.4 | Наследование. | 6 | 0 | 6 |
| 4.5 | Полиморфизм. | 3 | 0 | 3 |
| 4.6 | Взаимодействие объектов. | 4 | 0 | 4 |
| 4.7 | Простая программа на С#. | 3 | 0 | 3 |
| 4.8 | Использование компонентов. | 3 | 0 | 3 |
| 4.9 | Создание новых классов. | 4 | 0 | 4 |
|  | Резерв. | 1 | 1 | 0 |
| **Итого** | | **68** | **11** | **57** |
| Всего | | 138 | 30 | 108 |

# Описание модулей, изучаемых в рамках элективного курса

# Программирование на С++ в 10-11 классах информационно-технологического профиля.

Используемые сокращения: СР – самостоятельная работа, ПР – практическая работа.

## 10 класс (70 часов)

### Раздел «Алгоритмы и программирование»

| **Номер урока** | **Тема урока** | **Параграф учебника (номер, название)** | **Практические работы (номер, название)** | **Работы компьютерного практикума (источник, номер, название)** | **Количество часов на тему** | **Содержание программы** |
| --- | --- | --- | --- | --- | --- | --- |
|  | Язык С++. Самая простая программа. | **Часть 1. § 14.** Первые программы. |  |  | **1** | Повторение: понятие алгоритма, свойства алгоритма, исполнитель алгоритма (виды, основные характеристики), способы записи алгоритма.  Структура программы. Операторы ввода-вывода. Переменные.  Ключевые слова: компилятор, препроцессор, раздельная компиляция, вывод данных, поток вывода, поток ввода, пространство имен, символьная строка. |
|  | Вывод текста на экран. | **Часть 1. § 14.** Первые программы. | 1. Оператор вывода. | 1. Вывод текста на экран. | **1** |
|  | Диалоговые программы. | **Часть 1. § 15.** Диалоговые программы. |  | 1. Программа «Как тебя зовут?» | **1** | Диалоговые программы. Переменные и их типы. Сумма чисел.  Ключевые слова: ввод данных, переменная, объявление переменной, входной поток. |
|  | Переменные и их типы. | **Часть 1. § 15.** Диалоговые программы. | 1. Исправление ошибок в диалоговых программах |  | **1** |
|  | Сумма чисел. | **Часть 1. § 15.** Диалоговые программы. |  | 1. Сумма чисел. | **1** |
|  | Ограниченность значений целых чисел. | **Часть 1. § 18.** Обработка целых чисел. | 1. Сокращенная запись операций. |  | **1** | Предельные значения чисел. Различие между вещественными и целыми числами. Дискретное представление чисел. Программное повышение точности вычислений.  Хранение в памяти целых чисел: со знаком, без знака. Повторение: получение дополнительного кода числа.  Арифметические операции с целыми числами: сложение и вычитание, умножение и деление. Операция сравнения чисел. Повторение: поразрядные логические операции, сдвиги.  Ключевые слова: арифметические выражения, частное, остаток, форматный вывод, случайные числа, зерно. |
|  | Арифметические выражения. | **Часть 1. § 18.** Обработка целых чисел. |  | 1. Простые вычисления. | **1** |
|  | Деление и остаток. Вывод данных на экран. | **Часть 1. § 18.** Обработка целых чисел. | 1. Операции «**/»**  и **«%»**. | 1. Применение операций «**/»**  и **«%»** к решению задач. | **1** |
|  | Случайные целые числа. | **Часть 1. § 18.** Обработка целых чисел. |  | 1. Случайное целое число на заданном отрезке [a, b]. | **1** |
|  | Вещественные числа в языке С++. | **Часть 1. § 19.** Обработка вещественных чисел. | 1. Типы вещественных чисел. |  | **1** | Повторение: хранение в памяти вещественных чисел. Нормализованное представление вещественных чисел в компьютере.  Арифметические операции с вещественными числами: сложение и вычитание, умножение и деление.  Ключевые слова: вещественное число, научный формат, мантисса, форматный вывод, округление. |
|  | Ввод и вывод. | **Часть 1. § 19.** Обработка вещественных чисел. |  | 1. Формы записи вещественных чисел в программах. | **1** |
|  | Операции с вещественными числами. | **Часть 1. § 19.** Обработка вещественных чисел. |  | 1. Операции с вещественными числами. | **1** |
|  | Случайные вещественные числа. | **Часть 1. § 19.** Обработка вещественных чисел. | 1. Случайные вещественные числа. | 1. Случайные вещественные числа. | **1** |
|  | Условный оператор. | **Часть 1. § 20.** Ветвления. | 1. Ветвления. | 1. Ветвления. | **1** | Условный оператор. Полная и неполная формы записи условного оператора. Сложные условия в условном операторе. Множественный выбор.  Ключевые слова: условный оператор, полная форма, неполная форма, составной оператор, вложенный условный оператор, логические переменные. |
|  | Вложенный условный оператор. | **Часть 1. § 20.** Ветвления. |  | 1. Вложенные условные операторы. | **1** |
|  | Логические переменные. | **Часть 1. § 20.** Ветвления. | 1. Логические переменные. | 1. Логические переменные. | **1** |
|  | Сложные условия. | **Часть 1. § 20.** Ветвления. | 1. Сложные условия. | 1. Сложные условия. | **1** |
|  | Множественный выбор. | **Часть 1. § 20.** Ветвления. |  | 1. Множественный выбор. | **1** |
|  | Практикум: использование ветвлений. | **Часть 1. § 20.** Ветвления. |  | 1. Задачи на ветвления. | **1** |
|  | Контрольная работа «Ветвления». | **Часть 1. § 20.** Ветвления. |  |  | **1** |
|  | Цикл с предусловием. | **Часть 1. § 21.** Циклы. | 1. Циклы с предусловием. | 1. Циклы с предусловием. | **1** | Понятие циклического алгоритма. Циклы с предусловием. Циклы с постусловием. Циклы с переменными (счетчиком). Вложенные циклы.  Ключевые слова: цикл с предусловием, цикл с постусловием, цикл по переменной. |
|  | Циклы с постусловием. | **Часть 1. § 21.** Циклы. | 1. Циклы с постусловием. | 1. Циклы с постусловием. | **1** |
|  | Вычисление квадратного корня. | **Часть 1. § 21.** Циклы. |  | 1. Вычисление квадратного корня. | **1** |
|  | Циклы по переменной. | **Часть 1. § 21.** Циклы. | 1. Циклы по переменной. | 1. Циклы по переменной. | **1** |
|  | Вложенные циклы. | **Часть 1. § 21.** Циклы. |  | 1. Вложенные циклы. | **1** |
|  | Контрольная работа «Циклы». |  |  |  | **1** |
|  | Библиотека TX Library. Управляем пикселями. | **Часть 1. § 16.** Компьютерная графика. | 1. Компьютерная графика. |  | **1** | Библиотека TX Library. Управляем пикселями. Линии и фигуры. Замкнутые фигуры.  Ключевые слова: окно, координаты, оси координат, пиксель, цвет контура, цвет заливки, код цвета, прозрачный цвет. |
|  | Линии и фигуры. Замкнутые фигуры. | **Часть 1. § 16.** Компьютерная графика. |  | 1. Линии и фигуры. Замкнутые фигуры. | **1** |
|  | Длинная программа. | **Часть 1. § 17.** Процедуры. |  | 1. Процедуры | **1** | Длинная программа. Рефакторинг. Процедуры с параметрами.  Ключевые слова: подпрограмма, процедура, рефакторинг, аргументы, параметры, базовая точка. |
|  | Рефакторинг. | **Часть 1. § 17.** Процедуры. |  | 1. Рефакторинг. | **1** |
|  | Процедуры с параметрами. | **Часть 1. § 17.** Процедуры. |  | 1. Процедуры с параметрами. | **1** |
|  | Принципы анимации. Рисуем шарик. Начальное положение. | **Часть 1. § 22.** Анимация. |  | 1. Рисуем шарик. | **1** | Принципы анимации. Рисуем шарик. Начальное положение. Анимация движения. Обработка нажатия клавиши. Обработка нажатия клавиши.  Ключевые слова: анимация, процедура, пауза, нажатие клавиши. |
|  | Анимация движения. | **Часть 1. § 22.** Анимация. |  | 1. Анимация движения. | **1** |
|  | Обработка нажатия клавиши. | **Часть 1. § 22.** Анимация. |  | 1. Обработка нажатия клавиши. | **1** |
|  | Зачетная проектная работа «Графика на С++» |  |  | 1. Рисуем и анимируем «С Новым годом!» | **1** |
|  | Простые процедуры. | **Часть 2. § 12.** Процедуры. | 1. Процедуры. |  | **1** | Вспомогательные алгоритмы. Процедуры. Запись процедуры на языке программирования. Процедура с параметрами.  Ключевые слова: процедура, интерфейс, реализация, параметр, локальная переменная, глобальная переменная, область видимости, передача по ссылке. |
|  | Процедуры с параметрами. Несколько параметров. | **Часть 2. § 12.** Процедуры. |  | 1. Процедуры с параметрами. | **1** |
|  | Локальные и глобальные переменные. Процедуры, изменяющие аргументы. | **Часть 2. § 12.** Процедуры. | 1. Локальные и глобальные переменные. | 1. Процедуры, изменяющие аргументы. | **1** |
|  | Процедуры в графике. | **Часть 2. § 12.** Процедуры. |  | 1. Процедуры в графике. | **1** |
|  | Рекурсия. | **Часть 2. § 13.** Рекурсия. | 1. Рекурсия. | 1. Рекурсия. | **1** | Рекурсия. Примеры рекурсий в математике и литературе. Рекурсивная процедура (функция). Использование рекурсий при написании программ. Ключевые слова: рекурсивная процедура, фрактал, базовые объекты, условия остановки, анимация |
|  | Дерево Пифагора. | **Часть 2. § 13.** Рекурсия. |  | 1. Дерево Пифагора. | **1** |
|  | Анимация. | **Часть 2. § 13.** Рекурсия. |  | 1. Анимация. | **1** |
|  | Функция. Функции в С++. Примеры функций. | **Часть 2. § 14.** Функции. | 1. Функция. | 1. Функции. | **1** | Функция. Виды функций. Запись функций на языке программирования.  Ключевые слова: функция, параметры, вызов функции, результат функции, рекурсивная функция. |
|  | Логические функции. | **Часть 2. § 14.** Функции. |  | 1. Логические функции. | **1** |
|  | Рекурсивные функции. | **Часть 2. § 14.** Функции. |  | 1. Рекурсивные функции. |  |
|  | Символьные строки. Что такое символьная строка? | **Часть 2. § 15.** Символьные строки. | 1. Символьные строки. | 1. Рекурсия. | **1** | Символьная строка. Операции со строками: сцепление, удаление, копирование элементов. Функции поиска подстроки. Преобразование из строки в число и наоборот. Применение строковых данных в процедурах и функциях. Рекурсивный перебор символов. Сравнение т сортировка строк.  Ключевые слова: символьная строка, длина строки, сцепление строк, выход за границы строки, подстрока, удаление символов, вставка символов, поиск подстроки, замена подстроки, преобразование типов. |
|  | Сравнение, сцепление строк. | **Часть 2. § 15.** Символьные строки. |  | 1. Сравнение, сцепление строк. | **1** |
|  | Обращение к символам. Перебор всех символов. | **Часть 2. § 15.** Символьные строки. |  | 1. Обращение к символам. | **1** |
|  | Подстрока. Удаление и вставка. | **Часть 2. § 15.** Символьные строки. |  | 1. Подстрока. Удаление и вставка. | **1** |
|  | Поиск в символьных строках. Замена символов. | **Часть 2. § 15.** Символьные строки. |  | 1. Поиск в символьных строках. Замена символов. | **1** |
|  | Преобразования «строка – число». | **Часть 2. § 15.** Символьные строки. |  | 1. Преобразования «строка – число». | **1** |
|  | Символьные строки в функциях. | **Часть 2. § 15.** Символьные строки. | 1. Символьные строки в функциях. |  | **1** |
|  | Рекурсивный перебор. | **Часть 2. § 15.** Символьные строки. | 1. Рекурсивный перебор. | 1. Рекурсивный перебор. | **1** |
|  | Решение задач по теме «Процедуры и функции». | **Часть 2. § 12-15**. |  | 1. Решение задач по теме «Процедуры и функции». | **1** |
|  | Контрольная работа «Процедуры и функции». |  |  |  | **1** |
|  | Массивы в С++. Обращение к элементу массива. Перебор элементов массива. Вывод и ввод массива. | **Часть 2. § 16.** Массивы. | 1. Массивы. | 1. Перебор элементов массива. | **1** | Массивы в С++. Обращение к элементу массива. Перебор элементов массива. Вывод и ввод массива. Заполнение массива случайными числами. Алгоритмы обработки массивов.  Ключевые слова: массив, индекс элемента, значение элемента, константа, заполнение массива, вывод массива, ввод массиваа. |
|  | Заполнение массива случайными числами. Алгоритмы обработки массивов. | **Часть 2. § 16.** Массивы. | 1. Заполнение массива случайными числами. | 1. Заполнение массива случайными числами, обработка массивов | **1** |
|  | Используем массивы. Игра «Стрельба по тарелкам». Движение. Бьём тарелки. Показываем счёт. | **Часть 2. § 17.** Используем массивы. |  | 1. Используем массивы. Игра «Стрельба по тарелкам». Движение. Бьём тарелки. Показываем счёт. | **1** | Используем массивы. Игра «Стрельба по тарелкам».  Ключевые слова: массив, константа, инициализация, глобальные переменные, обработчик события. |
|  | Решение задач по теме «Массивы». | **Часть 2. § 16-17**. |  | 1. Решение задач по теме «Массивы». | **1** |
|  | Контрольная работа «Массивы». |  |  |  | **1** |
|  | Что такое матрица? Размещение матрицы в памяти. Заполнение и вывод матрицы на экран. | **Часть 2. § 18.** Матрицы. |  | 1. Заполнение и вывод матрицы на экран. | **1** | Матрица. Основные операции с матрицами: объявление, заполнение, вывод на экран. Обработка элементов матрицы.  Ключевые слова: матрица, строка, столбец, перебор элементов, вложенный цикл, квадратная матрица, главная диагональ, побочная диагональ, перестановка строк. |
|  | Обработка матриц. Квадратные матрицы. | **Часть 2. § 18.** Матрицы. | 1. Диагональные элементы. | 1. Обработка матриц. Квадратные матрицы. | **1** |
|  | Сложение и вычитание матриц. | **Часть 2. § 18.** Матрицы. |  | 1. Сложение и вычитание матриц. | **1** |
|  | Умножение матриц. | **Часть 2. § 18.** Матрицы. |  | 1. Умножение матриц. | **1** |
|  | Транспонирование матриц. | **Часть 2. § 18.** Матрицы. |  | 1. Транспонирование матриц. | **1** |
|  | Решение задач по теме «Матрицы». | **Часть 2. § 18.** Матрицы. |  | 1. Решение задач по теме «Матрицы». | **1** |
|  | Контрольная работа «Матрицы». |  |  |  | **1** |
|  | Системы управления версиями. Основные приемы работы с Git. | **Приложение.** Управление версиями. | 1. Системы управления версиями. Основные приемы работы с Git. |  | **1** | Системы управления версиями. Основные приемы работы с Git. Операции с файлами. Работа с удалённым архивом. |
|  | Операции с файлами. Работа с удалённым архивом. | **Приложение.** Управление версиями. |  | 1. Операции с файлами. Работа с удалённым архивом. | **1** |
|  | Реезерв. |  |  |  | **1** |  |

### Предметные результаты

1. Владение *универсальным языком программирования высокого уровня* С++, представлениями о базовых типах данных и структурах данных; умением использовать основные управляющие конструкции.
2. Владение умением *понимать программы*, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня; знанием основных конструкций программирования; умением анализировать алгоритмы с использованием таблиц.
3. Владение навыками *алгоритмического мышления* и понимание необходимости формального описания алгоритмов.
4. Овладение понятием *сложности алгоритма*, знание основных алгоритмов обработки числовой и текстовой информации, алгоритмов поиска и сортировки.

Учебный материал: 69 часов.

Резерв: 1 час.

## 11 класс (68 часов)

## Раздел «Алгоритмы и программирование»

### Модуль «Алгоритмизация и программирование» (68 часа)

| **Номер урока** | **Тема урока** | **Параграф учебника (номер, название)** | **Практические работы (номер, название)** | **Работы компьютерного практикума (источник, номер, название)** | **Количествочасов на тему** | **Содержание программы** |
| --- | --- | --- | --- | --- | --- | --- |
|  | Метод пузырька (Сортировка простыми обменами). | **Часть 3. § 13.** Простые алгоритмы сортировки. |  | 1. Метод пузырька (Сортировка простыми обменами). | **1** | Сортировка в массивах. Методы сортировки: метод пузырька, метод выбора, сортировка вставками. Массивы в подпрограммах.  Ключевые слова: сортировка, метод пузырька, сортировка вставками, перестановка элементов, сложность алгоритма. |
|  | Метод выбора. Сортировка вставками. | **Часть 3. § 13.** Простые алгоритмы сортировки. |  | 1. Метод выбора. Сортировка вставками. | **1** |
|  | Массивы в подпрограммах. | **Часть 3. § 13.** Простые алгоритмы сортировки. | 1. Массивы в подпрограммах. |  | **1** |
|  | Сортировка слиянием. | **Часть 3. § 14.** Быстрые алгоритмы сортировки и поиска. |  | 1. Массивы в подпрограммах. | **1** | Сортировка в массивах. Методы сортировки: сортировка слиянием, быстрая сортировка. Стандартная сортировка на языке С++. Двоичный поиск.  Ключевые слова: сортировка слиянием, быстрая сортировка, двоичный поиск, лямбда-функция. |
|  | Быстрая сортировка. | **Часть 3. § 14.** Быстрые алгоритмы сортировки и поиска. |  | 1. Быстрая сортировка. | **1** |
|  | Стандартная сортировка на языке С++. | **Часть 3. § 14.** Быстрые алгоритмы сортировки и поиска. |  | 1. Стандартная сортировка на языке С++. | **1** |
|  | Двоичный поиск. | **Часть 3. § 14.** Быстрые алгоритмы сортировки и поиска. |  | 1. Двоичный поиск. | **1** |
|  | Принцип сэндвича. Файловые потоки. | **Часть 3. § 15.** Обработка файлов. | 1. Принцип сэндвича. Файловые потоки. |  | **1** | Понятие файла. Типы файлов. Этапы работы с файлами: открытие файла, запись в файл, удаление записей из файла, чтение из файла, закрытие файла. Обработка массивов, записанных в файле. Обработка строк, записанных в файле. Обработка смешанных данных, записанных в файле.  Ключевые слова: файл, файловый поток, открытие файла, закрытие файла, чтение из файла, запись в файл, конец файла, аргументы командной строки. |
|  | Неизвестное количество данных. Обработка массивов. Чтение файлов по словам. | **Часть 3. § 15.** Обработка файлов. | 1. Обработка массивов. | 1. Чтение файлов по словам. | **1** |
|  | Построчная обработка файлов. Передача имени файла программе. | **Часть 3. § 15.** Обработка файлов. |  | 1. Построчная обработка файлов. Передача имени файла программе. | **1** |
|  | Решето Эратосфена. | **Часть 3. § 16.** Целочисленные алгоритмы. |  | 1. Решето Эратосфена. | **1** | Целочисленные алгоритмы. Решето Эратосфена. Длинные числа.  Ключевые слова: решето Эратосфена, длинные числа, разряды, перенос. |
|  | «Длинные» числа в С++. | **Часть 3. § 16.** Целочисленные алгоритмы. |  | 1. «Длинные» числа в С++. | **1** |
|  | Динамические массивы в С++. | **Часть 3. § 17.** Динамические массивы и словари. | 1. Динамические массивы в С++. |  | **1** | Понятие динамического массива. Объявление и заполнение динамического массива. Использование динамических массивов в подпрограммах. Расширение массива. Принципы работы динамического массива. Тип vector из библиотеки STL. Итераторы. Словари. Перебор элементов словаря.  Ключевые слова: динамический массив, указатель, контейнер, вектор, словарь, ключ, значение, перебор элементов, сортировка по ключу, итератор. |
|  | Тип vector из библиотеки STL. | **Часть 3. § 17.** Динамические массивы и словари. |  | 1. Тип vector из библиотеки STL. | **1** |
|  | Итераторы. Применение. | **Часть 3. § 17.** Динамические массивы и словари. |  | 1. Итераторы. Применение. | **1** |
|  | Словари. Перебор элементов словаря. | **Часть 3. § 17.** Динамические массивы и словари. |  | 1. Алфавитно-частотный словарь. | **1** |
|  | Структуры в С++. Обращение к полям структуры. | **Часть 3. § 18.** Структуры. | 1. Обращение к полям структуры. | 1. Ввод и вывод структур. | **1** | Понятие структуры. Объявление структур. Обращение к полю структуры. Работа с файлами. Сортировка.  Ключевые слова: структура, поле, точечная запись, сортировка, ключ. |
|  | Работа с файлами. | **Часть 3. § 18.** Структуры. | 1. Работа с файлами. | 1. Чтение структур из файла. | **1** |
|  | Сортировка. | **Часть 3. § 18.** Структуры. |  | 1. Сортировка структур. | **1** |
|  | Стек. | **Часть 3. § 19.** Стек, очередь, дек. |  | 1. Функции для работы со стеком из библиотеки STL. | **1** | Понятие стека. Использование динамического массива для создания стека. Вычисление арифметических выражений. Скобочные выражения. Очередь и деки как разновидности линейного списка.  Ключевые слова: стек, очередь, дек. |
|  | Очередь. Хранение очереди в массиве. | **Часть 3. § 19.** Стек, очередь, дек. |  | 1. Тип моделирования очереди queue. | **1** |
|  | Дек. | **Часть 3. § 19.** Стек, очередь, дек. |  | 1. Тип моделирования дека queue. | **1** |
|  | Деревья в С++. | **Часть 3. § 20.** Деревья. | 1. Деревья в С++. |  | **1** | Дерево. Двоичное дерево. Дерево поиска. Обход двоичного дерева. Вычисление арифметических выражений. Использование связанных структур. Хранение двоичного дерева в массиве. Ключевые слова: дерево, двоичное дерево, обход в глубину, обход в ширину, дерево поиска, ключ, хранение в массиве, модуль, проект, заголовочный файл. |
|  | Обходы дерева. | **Часть 3. § 20.** Деревья. | 1. Обходы дерева. | 1. Обходы дерева. | **1** |
|  | Деревья поиска. | **Часть 3. § 20.** Деревья. |  | 1. Деревья поиска. | **1** |
|  | Вычисления арифметических выражений. | **Часть 3. § 20.** Деревья. |  | 1. Вычисления арифметических выражений. | **1** |
|  | Дерево в массиве. Модульность. | **Часть 3. § 20.** Деревья. |  | 1. Дерево в массиве. Модульность. | **1** |
|  | Графы в С++. | **Часть 3. § 21.** Графы. | 1. Графы. |  | **1** | Понятие графа. «Жадные» алгоритмы. Алгоритм Прима-Крускала. Поиск кратчайших путей в графе. Алгоритм Дейкстры. Алгоритм Флойда-Уоршелла. Ключевые слова: жадный алгоритм, остовное дерево, задача коммивояжёра, Гамильтонов цикл, полный перебор, рекурсия, хвостовая рекурсия, случайный поиск, глобальные переменные, структуры. |
|  | Задача коммивояжёра. | **Часть 3. § 21.** Графы. |  | 1. Задача коммивояжёра. | **1** |
|  | Задача коммивояжёра: жадный алгоритм. | **Часть 3. § 21.** Графы. |  | 1. Задача коммивояжёра: жадный алгоритм. | **1** |
|  | Задача коммивояжёра: случайные перестановки. | **Часть 3. § 21.** Графы. |  | 1. Задача коммивояжёра: случайные перестановки. | **1** |
|  | Как избавиться от глобальных переменных? Передача данных по ссылке. | **Часть 3. § 21.** Графы. |  | 1. Передача данных по ссылке. | **1** |
|  | Одномерные задачи. | **Часть 3. § 22.** Динамическое программирование. |  | 1. Одномерные задачи. | **1** | Динамическое программирование. Поиск оптимального решения. Количество решений. Ключевые слова: динамическое программирование, перебор вариантов, мемоизация, оптимальная программа, редактирование строк, расстояние Левенштейна. |
|  | Редактирование строк. | **Часть 3. § 22.** Динамическое программирование. |  | 1. Редактирование строк. | **1** |
|  | Оптимальная стратегия. | **Часть 3. § 22.** Динамическое программирование. | 1. Оптимальная стратегия. | 1. Оптимальная стратегия. | **1** |
|  | Классы и объекты в С++: новая задача и ее анализ. Класс CMap. | **Часть 4. § 12.** Классы и объекты в С++. | 1. Классы и объекты в С++ |  | **1** | Классы и объекты в С++: новая задача и ее анализ. Класс CMap. Пишем свой конструктор. Рефакторинг. Рисуем карты.  Ключевые слова: класс, объект, экземпляр, модификатор доступа, конструктор, инициализация, метод, рефакторинг. |
|  | Пишем свой конструктор. Рефакторинг. | **Часть 4. § 12.** Классы и объекты в С++. |  | 1. Класс CMap. Пишем свой конструктор. | **1** |
|  | Рисуем карты. | **Часть 4. § 12.** Классы и объекты в С++. |  | 1. Рисуем карты. | **1** |
|  | Класс CCar. | **Часть 4. § 13.** Программа с классами (практикум). |  | 1. Класс CCar. | **1** | Класс CCar. Рефакторинг. Основная программа. Разбиение на модули. Основная программа. Разбиение на модули.  Ключевые слова: класс, объект, ссылка, конструктор, метод, рефакторинг. |
|  | Рефакторинг. | **Часть 4. § 13.** Программа с классами (практикум). |  | 1. Рефакторинг. | **1** |
|  | Основная программа. Разбиение на модули. | **Часть 4. § 13.** Программа с классами (практикум). |  | 1. Основная программа. Разбиение на модули. | **1** |
|  | Объект защищает свои данные. | **Часть 4. § 14.** Инкапсуляция. |  | 1. Объект защищает свои данные. | **1** | Объект защищает свои данные. Изменение внутреннего устройства. Свойство «только для чтения». Свойства в C#.  Ключевые слова: инкапсуляция, поле, метод, модификатор доступа, интерфейс, свойство |
|  | Изменение внутреннего устройства. | **Часть 4. § 14.** Инкапсуляция. |  | 1. Изменение внутреннего устройства. | **1** |
|  | Свойство «только для чтения». Свойства в C#. | **Часть 4. § 14.** Инкапсуляция. | 1. Свойства в C#. | 1. Свойство «только для чтения». | **1** |
|  | Моделирование жизни в океане. Иерархия классов. Базовый класс. | **Часть 4. § 15.** Наследование. |  | 1. Моделирование жизни в океане. | **1** | Моделирование жизни в океане. Иерархия классов. Базовый класс. Абстрактный класс. Защищенные поля и методы (protected). Неподвижные объекты. Подвижные объекты. Рыбы. Хищники. Вспомогательные процедуры и функции. Основная программа.  Ключевые слова: класс, иерархия классов, базовый класс, наследование, абстрактный класс, виртуальный метод, защищенные поля, утечка памяти. |
|  | Абстрактный класс. Защищенные поля и методы (protected). | **Часть 4. § 15.** Наследование. |  | 1. Абстрактный класс. Защищенные поля и методы (protected). | **1** |
|  | Неподвижные объекты. | **Часть 4. § 15.** Наследование. |  | 1. Неподвижные объекты. | **1** |
|  | Подвижные объекты. Рыбы. Хищники. | **Часть 4. § 15.** Наследование. |  | 1. Подвижные объекты. Рыбы. Хищники. | **1** |
|  | Вспомогательные процедуры и функции. | **Часть 4. § 15.** Наследование. |  | 1. Вспомогательные процедуры и функции. | **1** |
|  | Основная программа. Указатели на базовый класс. | **Часть 4. § 15.** Наследование. |  | 1. Основная программа. Указатели на базовый класс. | **1** |
|  | Полиморфизм в действии. | **Часть 4. § 16.** Полиморфизм. |  | 1. Полиморфизм в действии. | **1** | Полиморфизм в действии. Класс Океан. Деструктор.  Ключевые слова: базовый класс, класс -наследник, виртуальный метод, полиморфизм, позднее связывание, таблица виртуальных методов, деструктор. |
|  | Класс Океан. | **Часть 4. § 16.** Полиморфизм. |  | 1. Класс Океан. | **1** |
|  | Деструктор. | **Часть 4. § 16.** Полиморфизм. |  | 1. Деструктор. | **1** |
|  | Столкновения. | **Часть 4. § 17.** Взаимодействие объектов. |  | 1. Столкновения. | **1** | Столкновения. Изменение базового класса. Изменение других классов. Умные указатели.  Ключевые слова: столкновение, «мертвые» объекты, итератор, преобразование типов. |
|  | Изменение базового класса. | **Часть 4. § 17.** Взаимодействие объектов. |  | 1. Изменение базового класса. | **1** |
|  | Изменение других классов. | **Часть 4. § 17.** Взаимодействие объектов. |  | 1. Изменение других классов. | **1** |
|  | Умные указатели. | **Часть 4. § 17.** Взаимодействие объектов. |  | 1. Умные указатели. | **1** |
|  | RAD – среды для разработки программ. Язык С# и среда .NET. | **Часть 4. § 18.** Простая программа на С#. |  | 1. Создание формы в RAD-среде. | **1** | RAD – среды для разработки программ. Язык С# и среда .NET. Проект С#. Свойства объектов. Обработчики событий.  Ключевые слова: RAD – среда, среда NET., язык С#, виртуальная машина, проект, форма, свойство, событие, обработчик события, компонент, статистический класс. |
|  | Проект С#. Свойства объектов. | **Часть 4. § 18.** Простая программа на С#. |  | 1. Проект С#. Свойства объектов. | **1** |
|  | Обработчики событий. | **Часть 4. § 18.** Простая программа на С#. |  | 1. Обработчики событий. | **1** |
|  | Программы для просмотра рисунков. | **Часть 4. § 19.**  Использование компонентов. |  | 1. Программы для просмотра рисунков. | **1** | Программы для просмотра рисунков. Ввод и вывод данных. Обработка ошибок.  Ключевые слова: компонент, свойство, событие, обработчик события, родительский объект, дочерний объект, поле ввода, метка, статический метод, обработка ошибок, исключение. |
|  | Ввод и вывод данных. | **Часть 4. § 19.**  Использование компонентов. |  | 1. Ввод и вывод данных. | **1** |
|  | Обработка ошибок. | **Часть 4. § 19.**  Использование компонентов. |  | 1. Обработка ошибок. | **1** |
|  | Вычисление арифметических выражений: модель, новый класс. Методы класса. | **Часть 4. § 20.** Создание новых классов. |  | 1. Вычисление арифметических выражений: модель, новый класс. Методы класса. | **1** | Разработка программы для вычисления арифметических выражений: модель, новый класс. Методы класса. Вычисление арифметических выражений: представление. Добавляем новый компонент. Изменение поведения компонента.  Ключевые слова: модуль, модель, представление, класс, компонент, статический класс, методы класса, свойство. |
|  | Вычисление арифметических выражений: представление. | **Часть 4. § 20.** Создание новых классов. |  | 1. Вычисление арифметических выражений: представление. | **1** |
|  | Добавляем новый компонент. Изменение поведения компонента. | **Часть 4. § 20.** Создание новых классов. |  | 1. Добавляем новый компонент. Изменение поведения компонента. | **1** |
|  | Защита проектов | **Часть 4. § 20.** Создание новых классов. |  | Проект на свободную тему. | **1** |  |
|  | Резерв |  |  |  | **1** |  |

### Предметные результаты

1. Владение навыками *алгоритмического мышления* и понимание необходимости формального описания алгоритмов.
2. Овладение понятием *сложности алгоритма*, знание основных алгоритмов обработки числовой и текстовой информации, алгоритмов поиска и сортировки.
3. Владение стандартными приёмами *написания на алгоритмическом языке программы* для решения стандартной задачи с использованием основных конструкций программирования и отладки таких программ; использование готовых прикладных компьютерных программ по выбранной специализации.

Учебный материал: 67 часов.

Резерв: 1 час.

1. **Описание учебно-методического и материально-технического обеспечения учебного процесса**

Для реализации учебного курса «Программирование на Си» необходимо наличие компьютерного класса в соответствующей комплектации:

### Требования к комплектации компьютерного класса

Наиболее рациональным с точки зрения организации деятельности детей в школе является установка в компьютерном классе 13–15 компьютеров (рабочих мест) для школьников и одного компьютера (рабочего места) для педагога.

Предполагается объединение компьютеров в локальную сеть с возможностью выхода в Интернет, что позволяет использовать сетевые цифровые образовательные ресурсы.

Минимальные требования к техническим характеристикам каждого компьютера следующие:

* процессор – не ниже *Celeron* с тактовой частотой 2,8 ГГц;
* оперативная память – не менее 1Гб;
* жидкокристаллический монитор с диагональю не менее 15 дюймов;
* жёсткий диск – не менее 80 Гб;
* клавиатура;
* мышь;
* устройство для чтения компакт-дисков (желательно);
* аудиокарта и акустическая система (наушники или колонки).

Кроме того, в кабинете информатики должны быть:

* принтер на рабочем месте учителя;
* проектор на рабочем месте учителя;
* сканер на рабочем месте учителя;
* телекоммуникационный блок,
* устройства, обеспечивающие подключение к сети;
* устройства вывода звуковой информации;
* устройства для ручного ввода текстовой информации и манипулирования экранными элементами;
* устройства создания графической информации;
* устройства для создания музыкальной информации;
* устройства для записи (ввода) визуальной и звуковой информации.

### Требования к программному обеспечению компьютеров

На компьютерах, которые расположены в кабинете информатики, должна быть установлена операционная система *Windows* или *Linux*, а также **необходимое** программное обеспечение:

* интегрированная среда для программирования на языках С и C++, работающая под управлением операционной системы Windows. Среда Dev-C++, распространяемая свободно с исходными кодами (на Delphi) по лицензии GPL Dev-C++ http://www.bloodshed.net/dev/devcpp.html ,

а также:

* текстовый редактор (*Блокнот* или *Gedit*) и текстовый процессор (*Word* или *OpenOffice.org Writer*);
* табличный процессор (*Excel* или *OpenOffice.org Calc*);
* средства для работы с баз данных (*Access* или *OpenOffice.org Base*);
* графический редактор Gimp (<http://gimp.org>);
* редактор звуковой информации Audacity (<http://audacity.sourceforge.net>);
* среда программирования КуМир (<http://www.niisi.ru/kumir/>);
* среда программирования FreePascal (<http://www.freepascal.org/>);
* среда программирования Lazarus (<http://lazarus.freepascal.org/>)

и другие программные средства.

**Дополнительное** программное обеспечение

* файловый менеджер (в составе операционной системы или др.);
* антивирусная программа;
* программа-архиватор;
* клавиатурный тренажер;
* звуковой редактор;
* простая геоинформационная система;
* виртуальные компьютерные лаборатории;
* программа-переводчик;
* система оптического распознавания текста;
* мультимедиа проигрыватель (входит в состав операционных систем или др.);
* почтовый клиент (входит в состав операционных систем или др.);
* браузер (входит в состав операционных систем или др.).

**Учебно-методические средства обучения**

**Основная литература**

1. Поляков К.Ю. Программирование. Python. C++. Часть 1: учебное пособие /К.Ю. Поляков. – М.:БИНОМ. Лаборатория знаний, 2019.- 144 с.: ил.
2. Поляков К.Ю. Программирование. Python. C++. Часть 2: учебное пособие /К.Ю. Поляков. – М.:БИНОМ. Лаборатория знаний, 2019.- 176 с.: ил.
3. Поляков К.Ю. Программирование. Python. C++. Часть 3: учебное пособие /К.Ю. Поляков. – М.:БИНОМ. Лаборатория знаний, 2019.- 208 с.: ил.
4. Поляков К.Ю. Программирование. Python. C++. Часть 4: учебное пособие /К.Ю. Поляков. – М.:БИНОМ. Лаборатория знаний, 2019.- 192 с.: ил.

**Дополнительная литература**

1. Страуструп Б. Программирование: принципы и практика с использованием С++, 2-е изд.; Пер. с англ. - М.: ООО «И. Д. Вильямс», 2016.
2. Липпман С. Б., Лажойе Ж., Му Б.Э. Язык программирования С++. Базовый курс, 5-е изд. Пер. с англ. - М.: ООО «И. Д. Вильямс», 2014.
3. Шилдт Г. С++. Базовый курс, 3-е изд. Пер. с англ. - М.: ООО «И. Д. Вильямс», 2010.
4. Лафоре Р. Объектно-ориентированное программирование в С++, 4-е изд. Пер. с англ. – СПб.: Питер, 2004.
5. Мюссер Д., Дердж Ж., Сейни А. С++ и STL: справочное руководство, 2-е изд. Пер. с англ. - М.: ООО «И. Д. Вильямс», 2010.
6. Литвиненко Н. А. Технология программирования на С++. Начальный курс.. – СПб.: БХВ-Петербург, 2010 . – 281 с.
7. Романов Е.Л. Си++. От дилетанта до профессионала – СПб.: БХВ-Петербург, 2014 . – 600 с.
8. Федеральный государственный образовательный стандарт среднего (полного) общего образования. Утвержден приказом Министерства образования и науки Российской Федерации от 17 мая 2012 г. № 413.
9. К.Ю. Поляков, Е.А. Еремин. Информатика. 10-11 классы. Программа для старшей школы. Углубленный уровень. — М.: Бином, 2016.

**Дидактический материал**

К учебному пособию прилагаются дополнительные материалы:

* компьютерный практикум в электронном виде с комплектом электронных учебных средств, размещенный на сайте авторского коллектива: http://kpolyakov.spb.ru/school/pycpp.htm;
* материалы для подготовки к итоговой аттестации по информатике в форме ЕГЭ, размещенные на сайте http://kpolyakov.spb.ru/school/ege.htm;
* методические материалы для учителя, размещенные на сайте автора http://kpolyakov.spb.ru/school/pycpp.htm;
* подборка электронных образовательных ресурсов (далее ЭОР) с портала ФЦИОР (http://www.fcior.edu.ru);
* сетевая методическая служба авторского коллектива для педагогов на сайте издательства http://metodist.lbz.ru/authors/informatika/7/;
* интегрированная среда для программирования на языках С и C++, работающая под управлением операционной системы Windows. Среда Dev-C++, распространяемая свободно с исходными кодами (на Delphi) по лицензии GPL [Dev-C++](http://bloodshed.net/dev/devcpp.html) .