

# РАБОЧАЯ ПРОГРАММА ПО ЭЛЕКТИВНОМУ КУРСУ

## «ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C++»

### Содержание учебного предмета

#### 10 класс

#### 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, отдельная 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 C++ (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 C++. 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. 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
- Владение универсальным языком программирования высокого уровня (по выбору), представлениями о базовых типах данных и структурах данных; умением использовать основные управляющие конструкции.

- Владение умением понимать программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня; знанием основных конструкций программирования; умением анализировать алгоритмы с использованием таблиц.
- Владение навыками алгоритмического мышления и понимание необходимости формального описания алгоритмов.
- Овладение понятием сложности алгоритма, знание основных алгоритмов обработки числовой и текстовой информации, алгоритмов поиска и сортировки.
- 

## **11 класс**

### **Программирование на языке C++ (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 C++. 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*

- Владение универсальным языком программирования высокого уровня (по выбору), представлениями о базовых типах данных и структурах данных; умением использовать основные управляющие конструкции.
- Владение умением понимать программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня; знанием основных конструкций программирования; умением анализировать алгоритмы с использованием таблиц.
- Владение навыками алгоритмического мышления и понимание необходимости формального описания алгоритмов.
- 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 C++ (часть 4, глава 2). Объектно-ориентированное программирование -33 ч.**

*Тема 4.1. Классы и объекты в C++.*

Классы и объекты в C++: 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 COcean. 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 C#.*

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

Ключевые слова: RAD – среда, среда NET., язык C#, 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

- 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.

## **Планируемые результаты**

### **Личностные результаты:**

1. Ориентация обучающихся на достижение личного счастья, реализацию позитивных жизненных перспектив, инициативность, креативность, готовность и способность к личностному самоопределению, способность ставить цели и строить жизненные планы.

2. Готовность и способность к образованию, в том числе самообразованию, на протяжении всей жизни; сознательное отношение к непрерывному образованию как условию успешной профессиональной и общественной деятельности;

3. Готовность обучающихся к трудовой профессиональной деятельности как к возможности участия в решении личных, общественных, государственных, общенациональных проблем;

4. Сформированность целостного мировоззрения, соответствующего современному уровню развития науки и общественной практики, учитывающего социальное, культурное, языковое, духовное многообразие современного мира.

5. Освоенность социальных норм, правил поведения, ролей и форм социальной жизни в группах и сообществах, самореализация в группе и организации, ценности «другого» как равноправного партнера, формирование компетенций анализа, проектирования, организации деятельности, рефлексии изменений, способов взаимовыгодного сотрудничества, способов реализации собственного лидерского потенциала).

### **Метапредметные результаты**

#### **Выпускник научится:**

- самостоятельно определять цели, задавать параметры и критерии, по которым можно определить, что цель достигнута;
- ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях;
- оценивать ресурсы, в том числе время и другие нематериальные ресурсы, необходимые для достижения поставленной цели;
- выбирать путь достижения цели, планировать решение поставленных задач, оптимизируя материальные и нематериальные затраты;
- организовывать эффективный поиск ресурсов, необходимых для достижения поставленной цели;
- сопоставлять полученный результат деятельности с поставленной заранее целью;
- искать и находить обобщенные способы решения задач, в том числе, осуществлять развернутый информационный поиск и ставить на его основе новые (учебные и познавательные) задачи;
- выходить за рамки учебного предмета и осуществлять целенаправленный поиск возможностей для широкого переноса средств и способов действия;
- выстраивать индивидуальную образовательную траекторию, учитывая ограничения со стороны других участников и ресурсные ограничения;
- при осуществлении групповой работы быть как руководителем, так и членом команды в разных ролях (генератор идей, критик, исполнитель, выступающий, эксперт и т.д.);
- координировать и выполнять работу в условиях реального, виртуального и комбинированного взаимодействия. Предметные результаты. Выпускник научится:
  - составлять алгоритмы для решения учебных задач различных типов;
  - выражать алгоритм решения задачи различными способами (словесным, графическим, в том числе и в виде блок-схемы, с помощью формальных языков и др.);
  - определять результат выполнения заданного алгоритма или его фрагмента;
  - выполнять без использования компьютера («вручную») несложные алгоритмы управления исполнителями и анализа числовых и текстовых данных, записанные на конкретном языке программирования с использованием основных управляющих конструкций последовательного программирования (линейная программа, ветвление, повторение, вспомогательные алгоритмы);
  - использовать величины (переменные) различных типов и выражения, составленные из этих величин; использовать оператор присваивания;
  - использовать логические значения, операции и выражения с ними;
  - записывать на выбранном языке программирования арифметические и логические выражения и вычислять их значения;
  - определять результат выполнения алгоритма при заданных исходных данных; узнавать изученные алгоритмы обработки чисел и числовых последовательностей; создавать на их основе несложные программы анализа данных; читать и понимать несложные программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня;

создавать на алгоритмическом языке программы для решения типовых задач базового уровня из различных предметных областей с использованием основных алгоритмических конструкций и определять сложность изучаемых в курсе базовых алгоритмов;

использовать готовые прикладные компьютерные программы в соответствии с типом решаемых задач и по выбранной специализации;

анализировать предложенный алгоритм, например определять, какие результаты возможны при заданном множестве исходных значений и при каких исходных значениях возможно получение указанных результатов;

использовать основные понятия, конструкции и структуры данных последовательного программирования, а также правила записи этих конструкций и структур в выбранном для изучения языке программирования;

использовать в программах данные различных типов;

применять стандартные и собственные подпрограммы для обработки символьных строк; выбирать тип цикла в зависимости от решаемой подзадачи; составлять циклы с использованием заранее опред

выделять подзадачи, решение которых необходимо для решения поставленной задачи в полном объеме; реализовывать решения подзадач в виде подпрограмм, связывать подпрограммы в единую программу;

использовать модульный принцип построения программ; использовать библиотеки стандартных подпрограмм;

соблюдать санитарно-гигиенические требования при работе за персональным компьютером в соответствии с нормами действующих СанПиН.

### **3 Выпускник получит возможность научиться:**

создавать программы для решения задач, возникающих в процессе учебы и вне ее;

познакомиться с задачами обработки данных и алгоритмами их решения;

познакомиться с учебной средой составления программ управления автономными роботами и разобрать примеры алгоритмов управления, разработанными в этой среде.

## **Тематическое планирование**

### **10 класс**

№ п/п	Наименование разделов и тем программы	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всего	Контрольные работы	Практические работы	
<b>Раздел 1. Программирование на языке С++ (часть 1, глава 2) Основные управляющие конструкции.</b>					
1.1	Первые программы.	2		1	
1.2	Диалоговые программы.	3		2	
1.3	Компьютерная графика.	2		1	
1.4	Процедуры.	3		2	
1.5	Обработка целых чисел.	4		3	
1.6	Обработка вещественных чисел.	4		3	
1.7	Ветвления.	7		6	
1.8	Циклы.	5		5	
1.9	Анимация.	4		3	

<b>Итого</b>		<b>34</b>		<b>25</b>	
<b>Раздел 2. Программирование на языке C++ (часть 2, глава 2) Процедуры и функции — элементы структуризации программ. Массив — фундаментальная структура данных.</b>					
2.1	Процедуры.	4		3	
2.2	Рекурсия.	3		2	
2.3	Функции.	3		2	
2.4	Символьные строки.	10		8	
2.5	Массивы.	2		1	
2.6	Используем массивы.	3		2	
2.7	Матрицы.	7		6	
2.8	Системы управления версиями. Основные приемы работы с Git.	2		1	
<b>Итого</b>		<b>34</b>		<b>25</b>	
<b>Общее количество часов</b>		<b>68</b>		<b>50</b>	

### 11 класс

№ п/п	Наименование разделов и тем программы	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всего	Контрольные работы	Практические работы	
<b>Раздел 3. Программирование на языке C++ (часть 3, глава 2). Динамические структуры данных.</b>					
3.1	Простые алгоритмы сортировки.	3		2	
3.2	Быстрые алгоритмы сортировки и поиска.	4		3	
3.3	Обработка файлов.	3		2	
3.4	Целочисленные алгоритмы.	2		2	
3.5	Динамические массивы и словари.	4		3	
3.6	Структуры.	3		2	
3.7	Стеки, очереди, деки.	3		3	
3.8	Деревья.	5		4	
3.9	Графы.	5		4	
3.10	Динамическое программирование	3		2	
<b>Итого</b>		<b>35</b>		<b>30</b>	
<b>Раздел 4. Программирование на языке C++(часть 4, глава 2). Объектно-ориентированное программирование.</b>					
4.1	Классы и объекты в C++.	3		2	
4.2	Программа с классами (практикум).	3		3	
4.3	Инкапсуляция.	3		2	
4.4	Наследование.	6		6	
4.5	Полиморфизм.	3		3	
4.6	Взаимодействие	4		4	

	объектов.				
4.7	Простая программа на С#.	4		3	
4.8	Использование компонентов.	3		3	
4.9	Создание новых классов.	4		4	
Итого		33		<b>30</b>	
<b>Общее количество часов</b>		<b>68</b>		<b>60</b>	