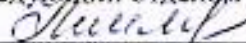


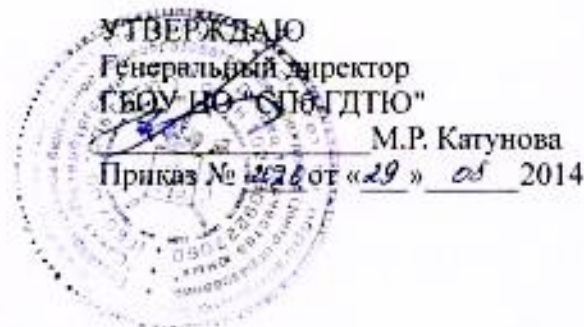
СОГЛАСОВАНО

Заведующий отделом техники

 Г.А. Тимофеева

Протокол педагогического совета

№ 4 от « 28 » мая 2014



ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ (ОБЩЕРАЗВИВАЮЩАЯ)  
ПРОГРАММА  
«ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ»

Срок реализации программы: 2 года  
Возраст обучающихся: 12-16 лет

Автор-составитель:  
Хайдаров Геннадий Гасимович  
педагог дополнительного образования

Рассмотрено Методическим советом  
ГБОУ ЦО «СПб ГДПО»  
Протокол № 9 от « 29 » 08 2014г

Санкт-Петербург  
2014

## Содержание

1 Пояснительная записка.....	3
1.1 Вступление.....	3
1.2 Направленность дополнительной образовательной программы.....	3
1.3 Новизна.....	3
1.4 Актуальность.....	3
1.5 Педагогическая целесообразность.....	3
1.6 Цели и задачи дополнительной образовательной программы.....	4
1.7 Отличительные особенности данной программы.....	4
1.8 Возраст детей.....	4
1.9 Формы и режим занятий.....	4
1.10 Ожидаемые результаты и способы определения результативности.....	5
1.11 Формы подведения итогов реализации дополнительной образовательной программы..	5
2. Учебно-тематический план.....	6
2.1 Перечень разделов и тем.....	6
2.2.1 Количество часов с разбивкой на виды занятия с общим объемом 144 часа в год.....	6
2.2.2 Количество часов с разбивкой на виды занятия с общим объемом 72 часа в год.....	6
3. Содержание дополнительной образовательной программы.....	7
3.1 Теоретические занятия первого полугодия первого года обучения.....	7
3.2 Теоретические занятия второго полугодия первого года обучения.....	8
3.3 Теоретические занятия первого полугодия второго года обучения.....	10
3.4 Теоретические занятия второго полугодия второго года обучения.....	10
3.5 Практические занятия.....	11
3.6 Консультационные занятия.....	11
4. Методическое обеспечение дополнительной образовательной программы.....	11
4.1 Обеспечение программы методическими видами продукции.....	11
4.2 Рекомендации по проведению теоретических, практических и консультационных занятий.....	12
4.3 Дидактические и лекционные материалы.....	12
4.4 Техническое оснащение занятий.....	12
Список использованной литературы.....	13
Список литературы для первого года программы обучения.....	13
Список литературы для второго года программы обучения.....	13
Приложение 1.....	15
Приложение 2.....	17
Приложение 3.....	19
Приложение 4.....	21
Приложение 5.....	23
Приложение 6.....	24

# **1 Пояснительная записка**

## **1.1 Вступление**

Программа обучения профессиональному программированию на языке С++ – «Программирование на языке С++ и объектно-ориентированное программирование» написана для двух вариантов обучения. **Первый вариант обучения рассчитан на 288 часов за 2 года (144 часа в год) и включает в себя курс обучения с выполнением всего объема заданий в учебных классах центра компьютерных технологий отдел техники. Второй вариант обучения рассчитан на 144 часа за 2 года (72 часа в год) и включает в себя курс обучения с выполнением части заданий в домашних условиях по выдаваемым домашним заданиям.** Оба варианта написаны с учетом современных требований к освоению информационных технологий. Для освоения курса обязательны навыки пользования компьютером и наличие дома компьютера для выполнения дополнительных домашних, самостоятельных и конкурсных заданий. Данная программа является основной в системе высококачественного многолетнего обучения. В качестве продолжения данной программы рекомендуется программа изучения программирования OpenGL с трехмерной графикой, а также программирования на языках С# и Java.

Данная программа – это модернизация ранее разработанных программ, утвержденных, лицензированных и прошедших практическую проверку в течении 10 лет в отделе техники ГОУ СПбГДТЮ: «Программирование С++ и С++Builder. Часть 1», «Программирование в С++Builder. Часть 2». Качество обучения указанной системы подтверждено международными дипломами учащихся.

Программа рассчитана как на учащихся договорных групп, так и на учащихся бюджетных групп. Темы лекций обоих вариантов одинаковы, а меняется лишь наполнение лекции и практических заданий в зависимости от конкретной группы учащихся. Так как технология С++ стремительно развивается, авторы оставляют за собой право ежегодно редактировать список тем занятий в сторону усложнения материала.

## **1.2 Направленность дополнительной образовательной программы**

Данная программа имеет техническую направленность и направлена на развитие навыков профессионального программирования. Для освоения курса обязательны навыки пользования компьютером и наличие дома компьютера для выполнения домашних заданий и подготовки к конкурсам. Программа является основной в изучении языка программирования С++ и может применяться для учебных групп первого года обучения и для учебных групп после изучения программирования на языках Basic, Pascal.

## **1.3 Новизна**

Данная программа специально разработана и адаптирована для учебных групп компьютерного центра городского дворца творчества юных в рамках выделяемого объема часов и основного направления обучения компьютерного центра.

## **1.4 Актуальность**

В настоящее время языки программирования на основе С++ заняли ведущее место в сфере программирования. На базе данных языков разработано много сред и других языков программирования: С++ Builder, Visual С++, С#, J Builder, Java, Perl и другие. Для освоения основ программирования на языке С++ и предназначена данная программа обучения.

## **1.5 Педагогическая целесообразность**

Педагогическая целесообразность данной программы связана с целями и задачами программы и может варьироваться от общей развивающей для учащегося до профессиональной.

## **1.6 Цели и задачи дополнительной образовательной программы**

Целями программы является обучение ученика основам программирования на языке C++, воспитание и развитие в нем желания к самостоятельной творческой работе с программными продуктами, адаптация его в мире современного программирования, а также создание условий для развития творческих возможностей.

Существует несколько задач:

### **1. Обучающие задачи.**

- Получение начальных знаний о программировании.
- Изучение основ C++.
- Умение поставить и реализовать самостоятельную задачу.

### **2. Развивающие задачи.**

- Умение составлять алгоритмы и реализовывать их на конкретном языке программирования.
- Развитие памяти за счет запоминания большого объема материала по функциям C++.
- Развитие навыков логического мышления
- Приобретение навыков работы с электронными справочниками - помощью (Help).

### **3. Воспитательные задачи.**

- Воспитание настойчивости при поиске ошибок в программах.
- Формирование понимания значимости компьютерных технологий для успешной карьеры в будущем.
- Развитие чувства ответственности за качество своих программ, представляемых на отчетную конференцию или конкурс
- Воспитание коллективизма при подготовке к поездкам на конференции, форумы и конкурсы, коллективные выезды в университеты города на факультеты информационных технологий
- Воспитание личностных волевых качеств во время самоподготовки докладов и выступлений на групповых и городских мероприятиях

## **1.7 Отличительные особенности данной программы**

Отличительными особенностями данной программы от типовых программ обучения программированию на языке высокого уровня является включение разработок, связанных с темой «Компьютерная графика». Данные особенности позволяют школьникам получить обзорную информацию об университетских дисциплинах в области информационных технологий и программирования, что позволит в дальнейшем более легко университетский курс обучения.

## **1.8 Возраст детей**

Рекомендуемый возраст учащихся 12-16 лет (7-11 классы). Коллектив учащихся по данной программе обучения формируется в основном из групп после изучения объектно-ориентированного программирования в ЦКТ ГДТЮ. На основе технических возможностей компьютерного центра и санитарных норм формируется численность групп.

## **1.9 Формы и режим занятий**

Данный курс обучения рассчитан на два учебных года. Каждый год обучения рассчитан на два варианта обучения.

К первому варианту обучения относится курс обучения, продолжительность которого составляет 144 академических часов за один учебный год. Из них 144 часа теория, 72 часа – практика или консультация в компьютерном классе. Занятия проводятся по 4 часа в неделю. Ежеженедельно по 2 час – теоретические занятия, по 2 час - практические занятия или

консультации. Этот вариант курса обучения следует применять для учащихся, выполняющих все практические задания в классе.

Ко второму варианту обучения относится курс обучения, продолжительность которого составляет 72 академических часа за один учебный год. Из них 36 часов теория, 36 часов – практика в компьютерном классе, практико-консультационные в компьютерном классе или консультационные занятия и интернет- диалоги для учащихся, пропустивших занятия по уважительным причинам. Занятия проводятся по 2 часа в неделю. Ежеженедельно по 1 часу – теоретические занятия и по 1 часу - практические занятия или консультационные занятия.

### ***1.10 Ожидаемые результаты и способы определения результативности***

За период обучения ожидаемым результатом для учащегося является получение знаний и навыков программирования на языке C++, работы в объектно-ориентированном программировании, что соответствует основной цели данной программы. В процессе достижения основной цели учащиеся выполняют обучающие и развивающие задачи данной программы. Кроме того, в процессе обучения происходит воспитание личностных качеств, описанных в задачах данной программы, умения общаться и работать в коллективе.

В качестве способа определения результативности обучения учащийся создает самостоятельную программу заданной сложности. Оценкой результативности обучения является практическая реализация, полученных в процессе обучения, знаний ребенком в виде законченной компьютерной программы. Ребенок может выступить с докладом и продемонстрировать свою разработку на внутригрупповом конкурсе программ, на «Ежегодной итоговой конференции Компьютерного центра» или на выездном смотре- конкурсе «ГДТЮ-ВУЗы». Для получения промежуточных результатов в январе месяце устраивается групповая промежуточная конференция. Способом проверки результатов учащихся их родителями два раза в год устраиваются открытые уроки. При необходимости устраиваются родительские собрания с целью демонстрации результатов учащихся.

В течение учебного года учащимся предлагается выполнить одно или несколько творческих заданий для дальнейшего выступления на конференциях внутри группы, на городских и на международных конференциях. Кроме того, учащимся предлагаются организованные выезды в ведущие университеты города. Данные мероприятия проводятся только для желающих, во внеурочное время. Проведение таких мероприятий приводит к качественным результатам в подготовке учащихся (дипломы, опыт общения, опыт выступления). В этом случае способом проверки результатов учащихся являются грамоты, благодарственные письма и дипломы соответствующих конкурсов.

### ***1.11 Формы подведения итогов реализации дополнительной образовательной программы***

Результаты обучения обсуждаются на городских и международных конференциях.

Основной формой подведения итогов учебного года в бюджетных группах является выпускная самостоятельная работа учащегося. Основной формой оценки результата в договорной группе является желание учащегося продолжить занятия по данному направлению обучения. Кроме данной формы оценки результата обучения существуют другие формы оценки. К ним относятся выступления: на конференциях внутри группы, на городских конференциях, организованные выезды в ведущие университеты города. При высокой оценке качества работы учащегося в бюджетной группе он может продолжить занятия по данному направлению обучения на следующий год.

## ***2. Учебно-тематический план***

Основное содержание курса изложено в настоящей программе. В него входят темы программирования, связанные с программированием на языке C++ и объектно-ориентированным программированием (предположительно в среде C++ Builder).

## 2.1 Перечень разделов и тем

Перечень разделов и тем с объемом 144 часа в год и 72 часа в год по двухлетнему обучению.

### 2.2.1 Количество часов с разбивкой на виды занятия с общим объемом 144 часа в год

Перечень разделов и тем с объемом 144 часа в год по первому году обучения.

N	Темы первого года обучения	Теория	Практика	Консультации	Всего
1	<u>Тема 1. Основные понятия языка C++</u>	16	16	0	32
2	<u>Тема 2. Циклы, массивы, функции</u>	16	16	0	32
3	<u>Тема 3. Графика C++</u>	8	8	0	16
4	<u>Тема 4. Структуры и файлы</u>	12	12	0	24
5	<u>Тема 5. Основы работы в визуальной среде</u>	10	10	0	20
6	<u>Тема 6. Работа с графикой в визуальной среде</u>	10	10	0	20
	Всего:	72	72	0	144

Перечень разделов и тем с объемом 144 часа в год по второму году обучения.

N	Темы первого года обучения	Теория	Практика	Консультации	Всего
1	<u>Тема 7. Теория классов. Применение классов</u>	40	40	0	80
2	<u>Тема 8. Графические классы, DirectDraw, основы трехмерного моделирования</u>	32	32	0	64
	Всего:	72	72	0	144

### 2.2.2 Количество часов с разбивкой на виды занятия с общим объемом 72 часа в год

Перечень разделов и тем с объемом 72 часа в год по первому году обучения.

N	Темы первого года обучения	Теория	Практика	Консультации	Всего
1	<u>Тема 1. Основные понятия языка C++</u>	8	8	0	16
2	<u>Тема 2. Циклы, массивы, функции</u>	8	8	0	16
3	<u>Тема 3. Графика C++</u>	4	4	0	8
4	<u>Тема 4. Структуры и файлы</u>	6	6	0	12
5	<u>Тема 5. Основы работы в визуальной среде</u>	5	5	0	10
6	<u>Тема 6. Работа с графикой в визуальной среде</u>	5	5	0	10
	Всего:	36	36	0	72

Перечень разделов и тем с объемом 72 часа в год по второму году обучения.

N	Темы первого года обучения	Теория	Практика	Консультации	Всего
1	<u>Тема 7. Теория классов. Применение классов</u>	20	20	0	40
2	<u>Тема 8. Графические классы, DirectDraw, основы трехмерного моделирования</u>	16	16	0	32
	Всего:	36	36	0	72

### **3. Содержание дополнительной образовательной программы**

#### **3.1 Теоретические занятия первого полугодия первого года обучения**

*Основы C++ (16 лекций).*

#### **Тема 1. Основные понятия языка C++(8 лекций).**

##### **Лекция 1.** Установка Borland C++ 3.0 на ПЭВМ.

Порядок установки пакетов среды программирования Borland C++ 3.0 на компьютере. Настройка системных файлов и самого пакета. Разбор возможных ошибок, вызванных некорректной установкой среды.

##### **Лекция 2.** Описание имен в C++ программах.

Правила записи C++ программ. Описание имен в C++ программах: объявление имени и его определение.

##### **Лекция 3.** Арифметические выражения, операторы вывода и ввода.

Арифметические выражения. Правила построения выражений в C++. Операция присваивания. Функции стандартной библиотеки для ввода и вывода переменных.

##### **Лекция 4.** Форматированный вывод.

Синтаксис функций стандартной библиотеки для вывода переменных. Примеры написания программ.

##### **Лекция 5.** Символы псевдографики на примере.

Символьные константы. Особенности синтаксиса C++ для выводов кодов ASCII

##### **Лекция 6.** Цветной вывод текста.

Функции для вывода цветного текста на дисплей. Примеры написания программ. Строковые константы, их тип и размер. Техника работы со строками. Фрагменты программы, определяющие длину строки и копирующие строку. Функции, делающие то же самое.

##### **Лекция 7.** Операторы: условный, выбора и перехода

Правильное применение операторов: условный, выбора и перехода. Примеры написания программ. Особенности синтаксиса C++.

##### **Лекция 8.** Коды.

Скан и ANSI коды клавиатуры и их чтение с помощью операторов ввода. Примеры написания программ. Управление с помощью стрелок.

#### **Тема 2. Циклы, массивы, функции(8 лекций).**

### **Лекция 9.** Операторы циклов

Операторы циклов: арифметический `for(;;...;..)`, с предусловием `while(..){...}`, с постусловием `do..while(..)`.

### **Лекция 10.** Массивы

Одномерные массивы. Объявление и инициализация массивов.

### **Лекция 11.** Массивы

Двухмерные и трехмерные массивы.

### **Лекция 12.** Массивы и указатели

Связь массивов и указателей на примере байтового массива.

### **Лекция 13.** Указатели и работа со строками

Значение указателей в языке C++. Указатель на переменную и адрес переменной в памяти компьютера. Особенности синтаксиса C++ при работе со строками.

### **Лекция 14.** Функции.

Функции в C++: объявление и определение.

### **Лекция 15.** Функции.

Передача параметров по значению. Возвращаемые значения.

### **Лекция 16.** Функции.

Передача параметров по указателю. Возвращаемые значения.

## ***3.2 Теоретические занятия второго полугодия первого года обучения***

### **Тема 3.** *Графика C++ (4 лекций).*

#### **Лекция 17.** Графический режим.

Инициализация и завершение работы с графической библиотекой. Понятие драйвера и режима. Таблицы драйверов и режимов.

#### **Лекция 18.** Графические примитивы.

Работа с отдельными точками. Рисование линейных объектов. Рисование прямолинейных отрезков. Рисование окружностей. Рисование дуг эллипса. Рисование сплошных объектов.

#### **Лекция 19.** Работа с цветами.

Закрашивание объектов. Понятие палитры. Работа с изображениями. Понятие окна (порта вывода).

#### **Лекция 20.** Улучшенная графика.

Понятие видеостраниц и работа с ними. Работа со шрифтами: стандартными и русскими.

### **Тема 4.** *Структуры и файлы (6 лекций).*

#### **Лекция 21.** Структуры. Вызов переменной по значению и по указателю.

Объявление структуры. Инициализация структуры. Присвоение значений элементам структуры.

#### **Лекция 22.** Массивы структур. Вызов переменной по значению и по указателю.

Объявление массива структур. Инициализация структуры. Присвоение значений элементам массива структур.

#### **Лекция 23.** Работа с файлами. Функции чтения и записи данных в файл.

Функции стандартной библиотеки для работы с файлами. Чтение и запись в текстовой и бинарный файл.



**Лекция 24.** Работа с файлами. Функции чтения и записи на низком уровне.

Особенности функций языка C++ для работы с файлами. Чтение и запись в файл на низком уровне.

**Лекция 25.** Позиционирование в файле.

Функции для чтения и записи данных в любой позиции файла. Типовые примеры программ.

**Лекция 26.** Запуск программы из командной строки.

Аргументы главной функции main(). Типовые примеры использования этих аргументов.

**Тема 5. Основы работы в визуальной среде (5 лекций).**

**Лекция 27.** Среда программирования Borland C++Builder

Общие сведения и история создания. Главное меню. Инспектор объектов. Демонстрация возможных программ. Задание опций интегрированной среды. Компиляция, запуск программы на выполнение. Написание программы с тремя кнопками выхода Button.

**Лекция 28.** Компоненты на закладках палитры компонентов.

Понятие компонента. Визуальные и не визуальные компоненты. Общие свойства компонентов элементов. Общие события управляющих элементов. Управление компонентами при проектировании: помещение компонентов на форму и их удаление. Перемещение и изменение размеров компонента.

**Лекция 29.** Инспектора объектов.

Понятие свойств объекта. Применение свойства Visible на конкретном примере. Понятие метода объекта. Применение метода OnClick() на конкретном примере. Методы Show(), ShowModal(), Hide().

**Лекция 30.** Окно редактора кода. Окно редактора формы.

Работа в редакторе кода. Добавление и стирание в редактор кода методов из инспектора объектов. Форма: Понятие формы. Имя и заголовок формы. Стиль формы. Размеры и положение формы на экране.

**Лекция 31.** Рабочие файлы C++Builder.

Структура приложений в C++ Builder. Проект. Понятие проекта. Файлы описания форм. Файлы программных модулей. Главный программный файл и файл проекта. Другие файлы проекта. Управление проектом: Создание, сохранение и открытие проекта. Компиляция и компоновка проекта.

**Тема 6. Работа с графикой в визуальной среде (5 лекций).**

**Лекция 32.** Графические функции C++Builder.

Графические примитивы Canvas. Работа с отдельными точками. Рисование линейных объектов. Рисование прямолинейных отрезков. Рисование окружностей. Рисование дуг эллипса.

**Лекция 33.** Графические функции C++Builder для заливки областей.

Заливка областей. Примеры программ с заливкой областей.

**Лекция 34.** Вывод графического текста функциями C++Builder.

Вывод графического текста. Создание прозрачного фона за текстом. Выбор типа и размера шрифта.

**Лекция 35.** Вывод рисунков на форму приложения в C++ Builder.

Функция загрузки рисунка из файла. Загрузка рисунка из ресурса.

**Лекция 36.** Примеры программ с выводом рисунков.

Особенность вывода рисунков в C++ Builder. Примеры программ с выводом рисунков областей. Примеры программ с движением рисунков на форме приложения.

### ***3.3 Теоретические занятия первого полугодия второго года обучения***

#### ***Тема 7. Теория классов. Применение классов (16 лекций).***

***Лекция 1.*** Обзорная лекция по средам программирования под Windows.

***Лекция 2.*** Основные принципы объектно-ориентированного программирования (ООП).

***Лекция 3.*** Создание функций классов в C++ Builder с помощью инспектора объектов.

***Лекция 4.*** Особенности работы с классами в C++ Builder.

***Лекция 5.*** Классы C++. Реализация идеи инкапсуляции в языке C++: концепция класса как определяемого пользователем типа данных. Синтаксис. Объявление класса, данные класса и функции-члены класса. Закрытая и открытая части класса.

***Лекция 6.*** Определение функций-членов класса. Вызов по значению и по указателю.

***Лекция 7.*** Перегрузка функций.

***Лекция 8.*** Конструктор класса в C++, синтаксис описания и семантика использования. Конструктор по умолчанию. Деструктор класса в C++, синтаксис описания и семантика использования.

***Лекция 9.*** Перегрузка конструктора класса в C++.

***Лекция 13.*** Операторы new,delete. Их применение для работы с классами.

***Лекция 14.*** Операторы new,delete. Проверка выделения памяти. Правила работы с выделением памяти.

***Лекция 15.*** Производные классы: основные сведения.

***Лекция 15.*** Производные классы: Порядок работы конструкторов и деструкторов.

***Лекция 16.*** Производные классы: вызов функций по значению и по указателю.

### ***3.4 Теоретические занятия второго полугодия второго года обучения***

#### ***Тема 7. Продолжение теории классов. Применение классов (4 лекции).***

***Лекция 17.*** Производные классы: вызов виртуальной функций по указателю.

***Лекция 18.*** Абстрактные классы: вызов чисто виртуальной функций по указателю.

***Лекция 19.*** Темплеты функций. Темплеты классов.

***Лекция 20.*** Дружественные функции, необходимость доступа с закрытой части класса. Спецификатор friend.

#### ***Тема 8. Графические классы, DirectDraw, основы трехмерного моделирования (16 лекций).***

***Лекция 21.*** Графические классы. Классы TCanvas, TPicture, TBitmap. Общие свойства, отличия и области применения.

***Лекция 22.*** Работа с цветами. Закрашивание объектов. Понятие палитры. Работа с изображениями.

***Лекция 23.*** Класс TPicture и функция Draw для написания игровой программы.

***Лекция 24.*** Улучшенная графика. Работа с растровыми изображениями в классе TBitmap. Отображение картинок, программируемые эффекты в классе TBitmap.

***Лекция 25.*** Музыкальное сопровождение. Компоненты System: MediaPlayer. Открытие звуковых файлов \*.wav. Работа с навигатором. Программирование работы MediaPlayer с учетом

времени показа слайда фотоальбома. Написание программы.

Лекция 26. Комплексные задания на примере механики программы «музыкальный альбом».

Лекция 27. Быстрая графика DirectDraw. Установка графического режима. Поверхность рисования. Вывод точки и графических примитивов.

Лекция 28. Быстрая графика DirectDraw. Загрузка картинок Bitmap.

Лекция 29. Быстрая графика DirectDraw. Смена видеостраниц.

Лекция 30. Быстрая графика DirectDraw. Работа со спрайтами.

Лекция 31. Комплексные задания на примере механики программы «мозаика1». Комплексные задания на примере механики программы «мозаика2».

Примеры программ с движением рисунков на форме приложения.

Лекция 32. Пример программы «арканойд».

Примеры программ с движением рисунков на форме приложения.

Лекция 33. Пример программы «лабиринт».

Примеры программ с движением рисунков на форме приложения.

Лекция 34. Простые способы улучшения вывода динамического изображения.

Вывода рисунков в C++ Builder с использованием события OnPaint и компонента PaintBox.

Лекция 35. Комплексные задания "Определение принадлежности точки  $A(x,y)$  к заданной области".

Уравнения линий. Неравенства. Системы неравенств и их программирование. Изучение и разбор текста программы.

Лекция 36. Комплексные задания «Геометрическое моделирование трехмерных объектов. Каркасная модель».

Ортогональные проекции. Таблицы точек. Таблицы линий. Подготовка к работе. Аксонометрические проекции.

### **3.5 Практические занятия**

Темы практических занятий и практико-консультационных занятий совпадают с темами теоретических занятий. При необходимости могут устраиваться дополнительные внеурочные консультационные занятия. Например, при подготовке учащихся для выступлений на городских или выездных конференциях, семинарах и конкурсах.

### **3.6 Консультационные занятия**

При необходимости могут устраиваться дополнительные внеурочные консультационные занятия. Например, при подготовке учащихся для выступлений на городских или выездных конференциях, семинарах и конкурсах.

## **4. Методическое обеспечение дополнительной образовательной программы**

### **4.1 Обеспечение программы методическими видами продукции**

На первых занятиях учащемуся дается обзор методического и программного обеспечения к данной программе занятий. Даются рекомендации по приобретению необходимого программного обеспечения. Указывается список методической литературы, существующей в магазинах города. Почти на каждом практических занятиях учащемуся выдается задание в бумажном или электронном виде. Пример задания имеется в приложении 5.

### **4.2 Рекомендации по проведению теоретических, практических и консультационных**

## **занятий**

Занятия строятся таким образом, чтобы учащийся после первых практических занятий мог дома выполнять дополнительные задания по программированию на компьютере. Практическая направленность обучения обуславливает следующее методическое построение теоретических занятий. После первых вводных занятий в сентябре начинается теоретическая часть в виде лекций. Задания выдаются с целью освоения программирования и подготовке учащегося к самостоятельной творческой работе по выбранной им теме. Теория, по своей тематике служат для ознакомления учащегося с основными конструкциями языка программирования. Практика – для закрепления теории и навыков самостоятельной работы. Каждому учащемуся на время практики предоставляется персональный компьютер. Консультации – для творческого развития учащегося.

Рекомендуются следующие виды теоретических, практических и консультационных занятия.

Теоретические занятия – это аудиторные занятия в форме:

- лекций, что является основной формой занятий,
- участие в семинарах,
- инструктажа по технике безопасности и правилам дорожного движения.

Практические аудиторные занятия.

- решение задач у доски,
- обсуждение итогов конференций и конкурсов,

Практические занятия на компьютерах:

- самостоятельная работа по заданию преподавателя
- самостоятельная работа над оригинальным авторским проектом,
- консультация по сложным вопросам программирования непосредственно за компьютером,

Консультационные занятия, выезды на конкурсы в ВУЗы СПб:

- консультаций по подготовке и репетиции сценария выступлений на выездных смотрах-конкурсах в ВУЗах города,
- консультации и беседы со специалистами ВУЗов, при проведении выездов,
- интернет- консультации по E-mail.

### **4.3 Дидактические и лекционные материалы**

Дидактические материалы по данной программе обучения собраны на сайте поддержки данной программы обучения <http://gmo.dtu.spb.ru> . К ним относятся материалы, созданные автором: примеры заданий, для выполнения учащимися в классе и дома; темы заданий для выполнения самостоятельных работ учащихся; презентации по темам обучения и по перспективам развития данной программы; картинки с заданиями; экранные копии работ, выполненных учащимися; ссылки на готовые дидактических материалы (книги) других авторов.

Пример лекционного материала имеется в приложении 6.

### **4.4 Техническое оснащение занятий**

#### **4.3.1 Необходимое техническое обеспечение**

Персональный компьютер не ниже Pentium 2, локальная сеть и выход в интернет.

#### **4.3.2. Необходимое программное обеспечение.**

Операционная система Windows XP, Borland C++ 3.0, предположительно C++ Builder или VisualStudio, офисные программы

## **Список использованной литературы**

### **Список литературы для первого года программы обучения**

#### **Основная литература для первого года программы обучения**

##### Основная литература для учащихся

1. Березин Б.И., Березин С.Б. Начальный курс С и С++. –М.: ДИАЛОГ-МИФИ, 2005, - 288с.
2. Культин Н.Б. С/С++ в задачах и примерах. –СПб.: БХВ- Петербург, 2006, -288с.

##### Основная литература для педагога

3. Касаткин А.И., Вальвачев А.Н. Профессиональное программирование на языке Си от TurboС к Borland С++; Справ. Пособие. Под общ. Ред. А.И. Касаткина. -Мн.: Выш. Шк., 1992. -240с.
4. Белецкий Я. Энциклопедия языка Си. -М.: Мир, 1992. -687с.
5. Прокофьев Б.П., Сухарев Н.Н., Храмов Ю.Е. Графические средства TurboС и TurboС++/ Под ред. Г.В. Генса, Ю.Е. Храмова. -М.: Финансы и статистика, СП "Ланит", 1992. -160с.
6. А.Я. Архангельский. Программирование в С++Builder5. –М.: ЗАО «Издательство БИНОМ», 2000. -1152с.
7. Страуструп Б. Язык программирования С++ (вторая редакция) Киев:"Диасофт", 1993 ч.1 - 264 стр., ч.2 - 296 стр.
8. Сурков К.А., Сурков Д.А., Вальвачев А.Н. Программирование в среде С++Builder. - Мн.: ООО"Попури", 1998. -576с.
9. Кальвет Чарльз и др. Borland С++Builder3. Энциклопедия пользователя: Пер. с англ. – К.: Издательство «ДиаСофт», 1998. -804с.

#### **Дополнительная литература для первого года программы обучения**

- 1 Нортон П., Уилтон Р. IBM PC и PS/2. Руководство по программированию. -М.: Радио и связь, 1994. -336с.
- 2 М Теллес Borland С++Builder: библиотека программиста. -СПб.: ПитерКом, 1998. -512с.
- 3 Тондо К., Гимпел С. Язык Си. Книга ответов: Пер. с англ. –М.: Финансы и статистика, 1994. -160с.
- 4 Ла Мот А., Ратклифф Д., Семинаторе М., Тайлер Д. Секреты программирования игр: Пер. с англ. –СПб.: Питер, 1995. -720с.

### **Список литературы для второго года программы обучения**

#### **Основная литература для второго года программы обучения**

##### Основная литература для учащихся

1. Березин Б.И., Березин С.Б. Начальный курс С и С++. –М.: ДИАЛОГ-МИФИ, 2005, -288с.
2. Культин Н.Б. Самоучитель С++ Builder. –СПб.: БХВ- Петербург, 2006, -320с.

##### Основная литература для педагога

3. А.Я. Архангельский. Программирование в С++Builder5. –М.: ЗАО «Издательство БИНОМ», 2000. -1152с.
4. Сурков К.А., Сурков Д.А., Вальвачев А.Н. Программирование в среде С++Builder. -Мн.:

ООО"Попури", 1998. -576с.

5. Кальвет Чарльз и др. Borland C++Builder3. Энциклопедия пользователя: Пер. с англ. –К.: Издательство «ДиаСофт», 1998. -804с.
6. М Теллес Borland C++Builder: библиотека программиста. -СПб.: ПитерКом, 1998. -512с.
7. Е.В. Шикин, А. В. Боресков Компьютерная графика. Динамика, реалистические изображения. -М. "ДИАЛОГ-МИФИ", 1995. -288 с.
8. Березин Б.И., Березин С.Б. Начальный курс С и С++. –М.: ДИАЛОГ-МИФИ, 2001, -288с.
9. Страуструп Б. Язык программирования С++ (вторая редакция) Киев:"Диасофт", 1993 ч.1 - 264 стр., ч.2 - 296 стр.

#### **Дополнительная литература для второго года программы обучения**

10. Нортон П., Уилтон Р. IBM PC и PS/2. Руководство по программированию. -М.: Радио и связь, 1994. -336с.
11. Тондо К., Гимпел С. Язык Си. Книга ответов: Пер. с англ. –М.: Финансы и статистика, 1994. -160с.
12. Ла Мот А., Ратклифф Д., Семинаторе М., Тайлер Д. Секреты программирования игр: Пер. с англ. –СПб.: Питер, 1995. -720с.
13. Касаткин А.И., Вальвачев А.Н. Профессиональное программирование на языке Си от TurboC к Borland C++; Справ. Пособие. Под общ. Ред. А.И. Касаткина. -Мн.: Выш. Шк., 1992. -240с.
14. Белецкий Я. Энциклопедия языка Си. -М.: Мир, 1992. -687с.
15. Прокофьев Б.П., Сухарев Н.Н., Храмов Ю.Е. Графические средства TurboC и TurboC++/ Под ред. Г.В. Генса, Ю.Е. Храмова. -М.: Финансы и статистика, СП "Ланит", 1992. -160с.

файл С++ В ООП-2013 г.

## Приложение 1

Утверждаю:

\_\_\_\_\_ /Тимофеева Г.А./

Календарный учебно-тематический план на 200\_/200\_ учебный год

Образовательная программа: “Программирование на языке С++ и объектно-ориентированное программирование” Первый год обучения. Объем -144 часа

**Педагог: Хайдаров Г.Г.**

Сроки	Название раздела, темы	Теория (кол-во часов)	Практика (кол-во часов)	примечания
	<u>Занятие 1.</u> Установка Borland C++ 3.0 на ПЭВМ.	2	2	
	<u>Занятие 2.</u> Описание имен в С++ программах.	2	2	
	<u>Занятие 3.</u> Арифметические выражения, операторы вывода и ввода.	2	2	
	<u>Занятие 4.</u> Форматированный вывод.	2	2	
	<u>Занятие 5.</u> Символы псевдографики на примере.	2	2	
	<u>Занятие 6.</u> Цветной вывод текста.	2	2	
	<u>Занятие 7.</u> Операторы: условный, выбора и перехода	2	2	
	<u>Занятие 8.</u> Управление с помощью стрелок.	2	2	
	<u>Занятие 9.</u> Операторы циклов	2	2	
	<u>Занятие 10.</u> Массивы	2	2	
	<u>Занятие 11.</u> Массивы	2	2	
	<u>Занятие 12.</u> Массивы и указатели	2	2	
	<u>Занятие 13.</u> Указатели и работа со строками	2	2	
	<u>Занятие 14.</u> Функции.	2	2	

	<u>Занятие 15.</u> Функции.	2	2	
	<u>Занятие 16.</u> Функции.	2	2	
	<u>Занятие 17.</u> Графический режим.	2	2	
	<u>Занятие 18.</u> Графические примитивы.	2	2	
	<u>Занятие 19.</u> Работа с цветами.	2	2	
	<u>Занятие 20.</u> Улучшенная графика.	2	2	
	<u>Занятие 21.</u> Структуры. Вызов переменной по значению и по указателю.	2	2	
	<u>Занятие 22.</u> Массивы структур. Вызов переменной по значению и по указателю.	2	2	
	<u>Занятие 23.</u> Работа с файлами. Функции чтения и записи данных в файл.	2	2	
	<u>Занятие 24.</u> Работа с файлами. Функции чтения и записи на низком уровне.	2	2	
	<u>Занятие 25.</u> Позиционирование в файле.	2	2	
	<u>Занятие 26.</u> Запуск программы из командной строки.	2	2	
	<u>Занятие 27.</u> Среда программирования Borland C++Builder	2	2	
	<u>Занятие 28.</u> Компоненты на закладках палитры компонентов.	2	2	
	<u>Занятие 29.</u> Инспектора объектов.	2	2	
	<u>Занятие 30.</u> Окно редактора кода. Окно редактора формы.	2	2	
	<u>Занятие 31.</u> Рабочие файлы C++Builder.	2	2	
	<u>Занятие 32.</u> Графические функции C++Builder.	2	2	
	<u>Занятие 33-36.</u> Резерв педагога.	8	8	

дата

подпись педагога



## Приложение 2

Утверждаю:

\_\_\_\_\_/Тимофеева Г.А./

Календарный учебно-тематический план на 200\_/200\_ учебный год

Образовательная программа: “Программирование на языке С++ и объектно-ориентированное программирование” Первый год обучения. Объем -72 часа

**Педагог: Хайдаров Г.Г.**

Сроки	Название раздела, темы	Теория (кол-во часов)	Практика (кол-во часов)	примечания
	<u>Занятие 1.</u> Установка Borland C++ 3.0 на ПЭВМ.	1	1	
	<u>Занятие 2.</u> Описание имен в С++ программах.	1	1	
	<u>Занятие 3.</u> Арифметические выражения, операторы вывода и ввода.	1	1	
	<u>Занятие 4.</u> Форматированный вывод.	1	1	
	<u>Занятие 5.</u> Символы псевдографики на примере.	1	1	
	<u>Занятие 6.</u> Цветной вывод текста.	1	1	
	<u>Занятие 7.</u> Операторы: условный, выбора и перехода	1	1	
	<u>Занятие 8.</u> Управление с помощью стрелок.	1	1	
	<u>Занятие 9.</u> Операторы циклов	1	1	
	<u>Занятие 10.</u> Массивы	1	1	
	<u>Занятие 11.</u> Массивы	1	1	
	<u>Занятие 12.</u> Массивы и указатели	1	1	
	<u>Занятие 13.</u> Указатели и работа со строками	1	1	
	<u>Занятие 14.</u> Функции.	1	1	

	<u>Занятие 15.</u> Функции.	1	1	
	<u>Занятие 16.</u> Функции.	1	1	
	<u>Занятие 17.</u> Графический режим.	1	1	
	<u>Занятие 18.</u> Графические примитивы.	1	1	
	<u>Занятие 19.</u> Работа с цветами.	1	1	
	<u>Занятие 20.</u> Улучшенная графика.	1	1	
	<u>Занятие 21.</u> Структуры. Вызов переменной по значению и по указателю.	1	1	
	<u>Занятие 22.</u> Массивы структур. Вызов переменной по значению и по указателю.	1	1	
	<u>Занятие 23.</u> Работа с файлами. Функции чтения и записи данных в файл.	1	1	
	<u>Занятие 24.</u> Работа с файлами. Функции чтения и записи на низком уровне.	1	1	
	<u>Занятие 25.</u> Позиционирование в файле.	1	1	
	<u>Занятие 26.</u> Запуск программы из командной строки.	1	1	
	<u>Занятие 27.</u> Среда программирования Borland C++Builder	1	1	
	<u>Занятие 28.</u> Компоненты на закладках палитры компонентов.	1	1	
	<u>Занятие 29.</u> Инспектора объектов.	1	1	
	<u>Занятие 30.</u> Окно редактора кода. Окно редактора формы.	1	1	
	<u>Занятие 31.</u> Рабочие файлы C++Builder.	1	1	
	<u>Занятие 32.</u> Графические функции C++Builder.	1	1	
	<u>Занятие 33-36.</u> Резерв педагога.	4	4	

дата

подпись педагога

## Приложение 3

Утверждаю:

\_\_\_\_\_ /Тимофеева Г.А./

Календарный учебно-тематический план на 200\_\_/200\_\_ учебный год

Образовательная программа: “Программирование на языке С++ и объектно-ориентированное программирование”. Второй год обучения Объем -144 часа

**Педагог: Хайдаров Г.Г.**

Сроки	Название раздела, темы	Теория (кол-во часов)	Практика (кол-во часов)	примечания
	<u>Занятие 1.</u> Обзорная лекция по средам программирования под Windows.	2	2	
	<u>Занятие 2.</u> Основные принципы объектно-ориентированного программирования (ООП).	2	2	
	<u>Занятие 3.</u> Создание функций классов в С++ Builder с помощью инспектора объектов.	2	2	
	<u>Занятие 4.</u> Особенности классов в С++ Builder.	2	2	
	<u>Занятие 5.</u> Классы С++. Реализация идеи инкапсуляции в языке С++: концепция класса как	2	2	
	<u>Занятие 6.</u> Определение функций-членов класса. Вызов по значению и по указателю.	2	2	
	<u>Занятие 7.</u> Перегрузка функций.	2	2	
	<u>Занятие 8.</u> Конструктор класса в С++, синтаксис описания и семантика использования.	2	2	
	<u>Занятие 9.</u> Перегрузка конструктора класса в С++.	2	2	
	<u>Занятие 10.</u> Указатель на класс this.	2	2	
	<u>Занятие 11.</u> Вложенный класс	2	2	
	<u>Занятие 12.</u> Перегрузка конструктора класса в С++.	2	2	
	<u>Занятие 13.</u> Операторы new,delete. Проверка выделения памяти. Правила работы с выделением памяти.	2	2	
	<u>Занятие 14.</u> Производные классы: основные сведения.	2	2	

	<b><u>Занятие 15.</u></b> Производные классы: Порядок работы конструкторов и деструкторов.	2	2	
	<b><u>Занятие 16.</u></b> Производные классы: вызов функций по значению и по указателю.	2	2	
	<b><u>Занятие 17.</u></b> Производные классы: вызов виртуальной функций по указателю.	2	2	
	<b><u>Занятие 18.</u></b> Абстрактные классы: вызов чисто виртуальной функций по указателю.	2	2	
	<b><u>Занятие 19.</u></b> Темплеты функций. Темплеты классов.	2	2	
	<b><u>Занятие 20.</u></b> Дружественные функции, необходимость доступа с закрытой части класса. Спецификатор friend.	2	2	
	<b><u>Занятие 21.</u></b> Графические классы. Классы TCanvas, TPicture, TBitmap.	2	2	
	<b><u>Занятие 22.</u></b> Работа с цветами. Закрашивание объектов. Понятие палитры. Работа с изображениями.	2	2	
	<b><u>Занятие 23.</u></b> Класс TPicture и функция Draw для написания игровой программы.	2	2	
	<b><u>Занятие 24.</u></b> Улучшенная графика. Работа с растровыми изображениями в классе TBitmap	2	2	
	<b><u>Занятие 25.</u></b> Музыкальное сопровождение. Компоненты System: MediaPlayer	2	2	
	<b><u>Занятие 26.</u></b> Комплексные задания на примере механики программы «музыкальный альбом».	2	2	
	<b><u>Занятие 27.</u></b> Быстрая графика DirectDraw. Установка графического режима. Поверхность рисования. Вывод точки и графических примитивов.	2	2	
	<b><u>Занятие 28.</u></b> Быстрая графика DirectDraw. Загрузка картинок Bitmap.	2	2	
	<b><u>Занятие 29.</u></b> Быстрая графика DirectDraw. Смена видеостраниц.	2	2	
	<b><u>Занятие 30.</u></b> Быстрая графика DirectDraw. Работа со спрайтами.	2	2	
	<b><u>Занятие 31.</u></b> Комплексные задания на примере механики программы «мозаика 1».	2	2	
	<b><u>Занятие 32.</u></b> Пример программы «арканойд».	2	2	
	<b><u>Занятие 33-36.</u></b> Резерв педагога.	4	4	

дата

подпись педагога

## Приложение 4

Утверждаю:

\_\_\_\_\_ /Тимофеева Г.А./

\_\_\_\_\_

Календарный учебно-тематический план на 200\_\_/200\_\_ учебный год

Образовательная программа: “Программирование на языке С++ и объектно-ориентированное программирование”. Второй год обучения. Объем -72 часа

**Педагог: Хайдаров Г.Г.**

Сроки	Название раздела, темы	Теория (кол-во часов)	Практика (кол-во часов)	примечания
	<u>Занятие 1.</u> Обзорная лекция по средам программирования под Windows.	1	1	
	<u>Занятие 2.</u> Основные принципы объектно-ориентированного программирования (ООП).	1	1	
	<u>Занятие 3.</u> Создание функций классов в С++ Builder с помощью инспектора объектов.	1	1	
	<u>Занятие 4.</u> Особенности классов в С++ Builder.	1	1	
	<u>Занятие 5.</u> Классы С++. Реализация идеи инкапсуляции в языке С++: концепция класса как	1	1	
	<u>Занятие 6.</u> Определение функций-членов класса. Вызов по значению и по указателю.	1	1	
	<u>Занятие 7.</u> Перегрузка функций.	1	1	
	<u>Занятие 8.</u> Конструктор класса в С++, синтаксис описания и семантика использования.	1	1	
	<u>Занятие 9.</u> Перегрузка конструктора класса в С++.	1	1	
	<u>Занятие 10.</u> Указатель на класс this.	1	1	
	<u>Занятие 11.</u> Вложенный класс	1	1	
	<u>Занятие 12.</u> Перегрузка конструктора класса в С++.	1	1	
	<u>Занятие 13.</u> Операторы new,delete. Проверка выделения памяти. Правила работы с выделением памяти.	1	1	
	<u>Занятие 14.</u> Производные классы: основные сведения.	1	1	

	<b><u>Занятие 15.</u></b> Производные классы: Порядок работы конструкторов и деструкторов.	1	1	
	<b><u>Занятие 16.</u></b> Производные классы: вызов функций по значению и по указателю.	1	1	
	<b><u>Занятие 17.</u></b> Производные классы: вызов виртуальной функций по указателю.	1	1	
	<b><u>Занятие 18.</u></b> Абстрактные классы: вызов чисто виртуальной функций по указателю.	1	1	
	<b><u>Занятие 19.</u></b> Темплеты функций. Темплеты классов.	1	1	
	<b><u>Занятие 20.</u></b> Дружественные функции, необходимость доступа с закрытой части класса. Спецификатор friend.	1	1	
	<b><u>Занятие 21.</u></b> Графические классы. Классы TCanvas, TPicture, TBitmap.	1	1	
	<b><u>Занятие 22.</u></b> Работа с цветами. Закрашивание объектов. Понятие палитры. Работа с изображениями.	1	1	
	<b><u>Занятие 23.</u></b> Класс TPicture и функция Draw для написания игровой программы.	1	1	
	<b><u>Занятие 24.</u></b> Улучшенная графика. Работа с растровыми изображениями в классе TBitmap	1	1	
	<b><u>Занятие 25.</u></b> Музыкальное сопровождение. Компоненты System: MediaPlayer	1	1	
	<b><u>Занятие 26.</u></b> Комплексные задания на примере механики программы «музыкальный альбом».	1	1	
	<b><u>Занятие 27.</u></b> Быстрая графика DirectDraw. Установка графического режима. Поверхность рисования. Вывод точки и графических примитивов.	1	1	
	<b><u>Занятие 28.</u></b> Быстрая графика DirectDraw. Загрузка картинок Bitmap.	1	1	
	<b><u>Занятие 29.</u></b> Быстрая графика DirectDraw. Смена видеостраниц.	1	1	
	<b><u>Занятие 30.</u></b> Быстрая графика DirectDraw. Работа со спрайтами.	1	1	
	<b><u>Занятие 31.</u></b> Комплексные задания на примере механики программы «мозаика I».	1	1	
	<b><u>Занятие 32.</u></b> Пример программы «арканойд».	1	1	
	<b><u>Занятие 33-36.</u></b> Резерв педагога.	4	4	

дата

подпись педагога

## Приложение 5

Пример практического задания.

Файл Ch05\_задания\_print\_05\_03\_2005.doc

```
// var1.Cpp // Демонстрация объявления структуры
#include <stdio.h>
#include <string.h>
#include <conio.h>
// Объявление имени шаблона
struct sotrudnik {
int number; char name[50]; int year; float oklad;};
// Точка с запятой в предыдущей строчке
// после фигурной скобки СТАВИТСЯ!
// Описание структуры с присвоением ей рабочего имени
struct sotrudnik first=
// Инициализация переменных структуры
{ 1, "Иванов И.И.", 30,4400.0};
// Описание структуры с присвоением ей рабочего имени
struct sotrudnik second=
// Инициализация переменных структуры
{ 2, "Петров А.А.", 40,5500.0};
void main(void){ clrscr();
printf("Номер: %2d. Фамилия: %20s. "
"Возраст: %d год. Оклад: %7.2f\n",
first.number, first.name, first.year, first.oklad);
printf("Номер: %2d. Фамилия: %20s. "
"Возраст: %d год. Оклад: %7.2f\n",
second.number, second.name, second.year, second.oklad);
getch(); }
```

## Глава 6. Работа с файлами

В профессиональном программировании почти все исполняемые файлы программ при своей работе взаимодействуют с другими файлами. Как правило, исполняемые модули читают информацию из входного файла и сохраняют ее в выводной файл. За примерами далеко ходить не надо. Вставьте в дисковод CD-ROM компакт-диск с любой современной игрой и посмотрите структуры папок, в каждой из которых вы найдете несколько десятков файлов. Типовой схемой работы с графическими файлами является загрузка файла данных в динамическую память. Для этого функцией `malloc` или оператором `new` выделяется память под данные файла. Далее из динамической памяти происходит перенаправление файла в видеопамять. Затем обязательно следует освобождение динамической памяти функцией `free` или оператором `delete`. При загрузке следующего графического файла процесс повторяется. Таким образом, для работы с большими объемами данных умение работать с файлами жизненно необходимо. Об этом еще в начале девяностых годов писал Питер Нортон, что кончилось время больших и длинных программ «спагетти», а началось время многофайловых программ-«сэндвичей».

### Функции чтения и записи

Простейшая последовательность действий при работе с файлом сводится к следующим операциям:

1. Создать указатель на файл.
2. Написать функцию открытия файла.
3. Написать функцию чтения или записи.
4. Использовать функцию закрытия файла в явном виде, а не ждать конца работы программы.

Данная последовательность действий напоминает порядок работы с графическими функциями (см. многочисленные примеры в главе 4), а именно: открыть графический режим, вывести графические функции, закрыть графический режим.

Приведем пример простейшей работы с файлом (листинг 6.1) и проиллюстрируем сказанное на конкретной программе.

#### Листинг 6.1. Чтение из файла и запись в файл

```
// file1.cpp 14.11.02
#include <stdio.h>
#include <conio.h>
int main(void)
{FILE *in, *out; // Указатели на структуру файла
  clrscr();
  if ((in = fopen("myfile1.cpp", "rt")) == NULL)
  // Проверка открытия файла myfile1.cpp для чтения
  {fprintf(stderr, "Не могу открыть файл ввода данных.\n");
  /* Файловый вывод информации в стандартный поток для сообщений об ошибках */
  getch(); return 1;}
  if ((out = fopen("myfile1.txt", "wt")) == NULL)
```



```

// Проверка открытия файла myfile1.cpp для записи
{fprintf(stderr, "Не могу открыть файл вывода данных.\n");
getch(); return 1;}
while (!feof(in))
/* Цикл чтения байтов входного файла, пока не достигнут
символ конца (EOF - end of file) файла */
fputc(fgetc(in), out);
/* Чтение очередного байта из входного файла и запись его в выходной файл */
fclose(out); fclose(in);
// Закрытие файлов
getch(); return 0;
}

```

Вторым действием работы с файлами является создание указателей на файл. В отличие от простой переменной указатель на файл является указателем на структуру. Эта структура объявлена в заголовочном файле среды программирования и имеет вид:

```

typedef struct {
short level;
unsigned flags;
char fd;
unsigned char hold;
short bsize;
unsigned char *buffer;
unsigned char *curp;
unsigned istemp;
short token;
} FILE; /* Это структура с именем FILE */

```

Каждый элемент данной структуры имеет собственное имя и назначение. В структуре `FILE` хранится информация о буферизации, флаги индикации ошибок и достижения конца файла, указатель текущей позиции файла. Например, элемент `char fd` — это дескриптор открытого файла ([file descriptor](#)).

Еще одна структура с именем `ffblk`, полезная для понимания работы с файлами, включает в себя служебную информацию о файле.

```

struct ffblk {
char ff_reserved[21];
char ff_attrib;
unsigned ff_ftime;
unsigned ff_fdate;
long ff_fsize;
char ff_name[13];
};

```

В этой структуре хранится информация об атрибутах файла, времени и дате его создания, длине имени файла и само его имя. Например, поле `long ff_fsize` содержит текущее значение длины файла в байтах.

Две представленные структуры созданы для удобства работы программиста, которому в самом простом случае необходимо объявить только указатель на структуру `FILE`, а далее он может работать со всеми файловыми функциями (начинающимися с буквы `f`).

После объявления указателя следует присваивание указателю ссылки на структуру `FILE` по адресу памяти, возвращаемому функцией `fopen()`. Например, `in = fopen("myfile1.cpp", "rt")`, то есть здесь указатель `in` получит значение адреса файла с именем `myfile1.cpp` для дальнейшего с ним оперирования. Если функция по какой-то причине не может получить адрес памяти (например, нет файла с таким именем), тогда в указатель записывается нулевое значение. Последнее применяется как проверка правильности открытия файла: `if (in == NULL) {fprintf (stderr, " Не могу открыть файл ввода данных.\n"); getch(); return 1;}`. Эта строка предписывает при отсутствии нужного файла вывести сообщение и выйти из программы со значением `1`. Похожие действия производятся с файлом `myfile1.txt` (см. листинг 6.1)

Вторым действием работы с файлами идут функции открытия файлов.

Функция `fopen("myfile1.txt", "wt")` выясняет возможность создания и записи информации в файл. Если эту программу (листинг 6.1) записать на компакт-диск и запустить с него на выполнение, то программа не сумеет создать на компакт-диске файл с именем `myfile1.txt`, а значит будет выдано сообщение “Не могу открыть файл вывода данных”.

Далее на очереди функции чтения или записи в файл. В этом же примере чтение осуществляется в цикле `while` от начала файла до его конца. Функция `fgetc(in)` читает данные побайтово из входного файла `myfile1.cpp` и сразу же записывает их в выходной файл `myfile1.txt` функцией `fputc(fgetc(in), out)`.

Финальным действием работы с файлами является их закрытие функциями `fclose(out); fclose(in);`.

Теперь более подробно рассмотрим функции работы с файлами.

Главная функция, без которой мы не сможем работать с другими, это функция `fopen()`. Первый ее аргумент задает путь к файлу и его имя. Второй аргумент функции указывает на строку символов, определяющих способ доступа к файлу, и может принимать следующие значения:

- `r` — открыть для чтения (файл должен существовать);
- `w` — открыть пустой файл для записи; если файл существует, то его содержимое теряется;
- `a` — открыть для записи в конец файла (добавления); файл создается, если он не существует;
- `r+` — открыть для чтения и записи (файл обязан существовать);
- `w+` — открыть пустой файл для чтения и записи; если файл с таким именем уже есть, его содержание теряется;
- `a+` — открыть для чтения и добавления; файл создается, если он не существует.

Для того чтобы указать, что файл должен быть открыт в режиме последовательного доступа (текстового режима), необходимо в строку добавить атрибут `t` («текстовый»), если нужен произвольный доступ, файл следует открыть в бинарном (двоичном) режиме — `b`. Когда не указан ни атрибут `b`, ни атрибут `t`, принимается режим, задаваемый по умолчанию глобальной переменной, то есть последовательный.

## ПРИМЕЧАНИЕ

Для работы со всеми типами файлов, кроме текстового типа, применяйте бинарный режим. Типичной ошибкой является отсутствие атрибута `b` при открытии графического файла.

Для того, чтобы открыть нужный файл, нужно знать его имя. Следующий пример (листинг 6.2) программы работы с файлом осуществляет требуемые подготовительные действия.

**Листинг 6.2.** Ввод имени файла с клавиатуры

```
// file2.cpp 14.11.02
```

```

#include <stdio.h>
#include <conio.h>
#include <string.h>
int main(void)
{ FILE *in, *out;
  unsigned char file_name[20],sin[24],sout[24];int len;
  begin:: clrscr();
  puts("Введите имя файла [расширение CPP вводить не обязательно]");
  gets(file_name); len=strlen(file_name);
  printf("Длина введенного имени файла \"%s\" =%i\n",file_name,len);
  if(len<1 || len >12) goto begin;
  for(int i=0;i<13;i++) if(file_name[i]=='.')file_name[i]='\0';
  file_name[8]='\0';
  strcpy(sin, file_name); strcat(sin, ".cpp");
  strcpy(sout,file_name); strcat(sout, ".txt");
  if ((in = fopen(sin, "rt")) == NULL)
  {fprintf(stderr, "Не могу открыть %s файл.\n",sin);
  getch(); return 1;}
  if ((out = fopen(sout, "wt")) == NULL)
  {fprintf(stderr, "Не могу открыть %s файл.\n",sout);
  getch(); return 1;}
  while (!feof(in))
  fputc(fgetc(in), out);
  fclose(in); fclose(out);
  getch(); return 0;
}

```

В данном примере мы вводим с клавиатуры имя файла с расширением CPP для чтения его со стандартного устройства ввода, то есть с той же клавиатуры. Если имя не введено, следует повторный запрос. Если мы задаем имя файла без расширения, то программа добавляет расширение CPP. Если длина имени до расширения превышает 8 символов, то программа усекает его до этого значения, согласно соглашению DOS об именах файла. В результате работы

П  
р  
о  
г  
р  
а  
м  
м  
ы

П  
о  
л  
у  
ч  
а  
е  
м