

Министерство образования и науки Украины
Донбасская государственная машиностроительная академия

ПРОГРАММИРОВАНИЕ И АЛГОРИТМИЧЕСКИЕ ЯЗЫКИ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к выполнению лабораторных и самостоятельных работ
(для студентов направления «Системный анализ»)

Утверждено
на заседании кафедры ИСПР
Протокол № 2 от 9 сентября 2014г.

Краматорск 2014

Программирование и алгоритмические языки : методические указания к выполнению лабораторных и самостоятельных работ для студентов направления «Системный анализ» всех форм обучения / Сост. А. Ю. Мельников. – Краматорск: ДГМА, 2014. – 28 с.

Содержат задания к работам по составлению диаграмм классов и программ на языках объектно-ориентированного программирования Object-Pascal и Turbo-C++.

Составитель	Мельников А.Ю., канд. техн. наук, доцент
-------------	------------------------------------------

Отв. за выпуск	Мельников А.Ю., канд. техн. наук, доцент
----------------	------------------------------------------

СОДЕРЖАНИЕ

1	Указания к самостоятельной работе.....	4
1.1	Теоретическая подготовка.....	4
1.2	Практическая подготовка.....	7
2	Задания к лабораторным работам.....	7
	Тема 1. Объектная модель и объектно-ориентированный под- ход.....	7
	Лабораторная работа № 1. Объектная модель.....	7
	Лабораторная работа № 2. Классы и объекты.....	9
	Тема 2. Объектно-ориентированное программирование на языке C++	9
	Лабораторная работа № 3. Программирование на языке C.....	9
	Лабораторная работа № 4. Перегрузка функций.....	19
	Лабораторная работа № 5 Передвижение графических объектов на языке C++	21
	Лабораторная работа № 6 Построение диаграммы классов.....	26
	
	Список рекомендованной литературы.....	28

1 УКАЗАНИЯ К САМОСТОЯТЕЛЬНОЙ РАБОТЕ

Самостоятельная работа студентов по дисциплине «Объектно-ориентированное программирование» включает в себя:

- теоретическую подготовку (изучение конспекта лекций и дополнительной литературы);
- практическую подготовку (составление программ на языках объектно-ориентированного программирования Object-Pascal и Turbo-C++).

Уровень практической подготовки оценивается выполнением лабораторных и контрольных работ, теоретической – устными опросами и экзаменом.

1.1 Теоретическая подготовка

Поскольку дисциплина состоит из двух тем («Объектная модель и объектно-ориентированный подход» и «Объектно-ориентированное программирование на языке C++»), конспект лекций [1] содержит два относительно независимых раздела.

В первом рассматриваются основные понятия объектно-ориентированного подхода: определение и преимущества объектной модели, перечень ее главных и дополнительных элементов, определения 4ROUPсов и объектов, их состояния, поведения и идентичности, разновидности отношений между классами и между объектами, подходы к классификации знаний и к объектно-ориентированному анализу. Рекомендуемая дополнительная литература к первому разделу – [2-3].

Вопросы для самоконтроля по теме «Объектно-ориентированный подход»:

1. Методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного класса, называется...
2. Методология проектирования, соединяющая в себе процесс объектной декомпозиции и приемы представления логической и физической, а также статической и динамической моделей проектируемой системы, называется...
3. Методология, при которой требования к системе воспринимаются с точки зрения классов и объектов, выявленных в предметной области, называется...
4. Абстрагирование относится к...
5. Инкапсуляция относится к...
6. Модульность относится к...
7. Иерархия относится к...
8. Типизация относится к...
9. Параллелизм относится к...
10. Сохраняемость относится к...

11. Отношение «is a» называется...
12. Отношение «part of» называется...
13. Операция, которая изменяет состояние объекта, называется...
14. Операция, которая считывает состояние объекта, но не меняет его, называется...
15. Операция, которая позволяет организовать доступ ко всем частям объекта в строго определенной последовательности, называется...
16. Между объектами могут быть такие отношения (указать все варианты)...
17. Между классами могут быть такие отношения (указать все варианты) ...
18. Если отношение между классами характеризуется мощностью отношения, то оно называется...
19. Если отношение между классами изображается простым отрезком прямой линии, то оно называется...
20. Если отношение между классами изображается отрезком прямой линии с обычной стрелкой, то оно называется...
21. Если отношение между классами изображается отрезком прямой линии с закрашенным кружком, то оно называется...
22. Если отношение между классами изображается отрезком прямой линии с незакрашенным кружком, то оно называется...
23. Подходами к идентификации классов и объектов являются (указать все возможные варианты)...
24. Подход к анализу объектно-ориентированных систем, основанный на классической категоризации, называется...
25. Подход к анализу объектно-ориентированных систем, рассматривающий поведение как первоисточник классов и объектов, называется...
26. Подход к анализу объектно-ориентированных систем, основанный на мнении экспертов предметной области, называется...
27. Подход к анализу объектно-ориентированных систем, основанный на переборе сценариев классических подходов, поведения и предметной области, называется...
28. Подход к анализу объектно-ориентированных систем, основанный на «ручном» выстраивании иерархии классов в виде карточек специального вида, называется...
29. Подход к анализу объектно-ориентированных систем, основанный на описании задачи обычным языком с подчеркиванием существительных (будущих классов) и глаголов (будущих операций), называется...
30. Подход к анализу объектно-ориентированных систем, основанный на расширении имеющейся модели, описанной диаграммами потоков данных и некоторыми другими, называется...

Во втором разделе изучается объектно-ориентированное программирование в среде C++: понятие класса как расширения понятия структуры, компонентные данные и компонентные функции, конструкторы и деструкторы, уровни доступности (видимости) компонент класса, друже-

ственные функции, расширение действия (перегрузка) стандартных операций и функций, указатель `this`, множественное наследование и виртуальные базовые классы, виртуальные функции и абстрактные классы, локальные `GROUP`сы. Рекомендуемая дополнительная литература ко второму разделу – [4-7].

Вопросы для самоконтроля по теме «Объектно-ориентированное программирование на языке C++»:

1. Задание данных вместе с функциями их обработки, что превращает их в новый тип данных «класс», называется...
2. Описание класса с дальнейшим его использованием для порождения иерархии классов с наследованием доступа каждого из них к коду и данным предка, называется...
3. Задание действию одного имени, которое передается вверх или вниз по иерархии классов, причем способ реализации действий для разных классов по иерархии может отличаться, называется...
4. Данные класса называются...
5. Функции класса называются...
6. Если класс объявляется вне любого блока, он называется...
7. Если класс объявляется внутри какого-либо блока, он называется...
8. Если тело метода определено внутри класса, то этот метод называется...
9. Метод, имя которого совпадает с именем класса, и который вызывается автоматически при создании объекта класса, называется...
10. Метод, имя которого содержит имя класса, и который вызывается автоматически при ликвидации объекта класса, называется...
11. Переменные типа «класс» называются (перечислить все возможные варианты)...
12. Для ссылок на элементы объекта внутри метода используется указатель...
13. Может ли конструктор возвращать значения?
14. Может ли класс иметь несколько конструкторов с разными параметрами для разных видов инициализации?
15. Может ли деструктор иметь аргументы и возвращать значения?
16. Элементы класса, доступные только в пределах этого класса, задаются с помощью спецификатора доступа...
17. Элементы класса, доступные в пределах этого класса и его потомков, задаются с помощью спецификатора доступа...
18. Общедоступные элементы класса задаются с помощью спецификатора доступа...
19. Наследование, при котором производный класс имеет одного родителя, называется...
20. Наследование, при котором производный класс имеет более одного родителя, называется...
21. Как описывается чистый виртуальный метод?
22. Класс, содержащий хотя бы один чисто виртуальный метод, называется...

1.2 Практическая подготовка

Практическая подготовка заключается в составлении программ на языках объектно-ориентированного программирования (Object-Pascal и Turbo-C++). Студент самостоятельно должен составить все программы согласно индивидуальному заданию, руководствуясь примерами из данных методических указаний. Для создания приложений в среде Borland-Delphi (Lazarus) используются материалы предыдущего модуля.

2 ЗАДАНИЯ К ЛАБОРАТОРНЫМ РАБОТАМ

Тема 1. Объектная модель и объектно-ориентированный подход

Лабораторная работа № 1. Объектная модель

Цель работы: закрепить навыки создания приложений в среде визуального программирования, основанной на языке Object-Pascal (Borland-Delphi или Lazarus).

Задание к работе

1. Создать текстовый файл, содержащий данные по студентам в следующем виде: Фамилия Имя Группа Год_рождения R1 R2 R3 R4 R5 (Например: Иванов Иван ИС-06зт 1989 90 68 57 75 82), где R1 .. R5 – рейтинги по пяти предметам.
2. Разработать полнофункциональное Windows-приложение в среде визуального программирования Delphi (Lazarus), которое бы считывало информацию из файла в массив типа «запись» и осуществляло его обработку. Программа должна содержать, как минимум, главное меню («Вывод данных на экран», «Вывод данных в текстовый файл», «Расчет дополнительного поля», «Сортировка данных по заданному полю», «Построение диаграмм», «Выход»), мемо-поле для вывода результатов и специальная компонента для построения диаграмм. Возможно наличие других компонент для улучшения интерфейса приложения (например, дополнительные пункты меню или кнопки «Печать диаграмм» и «Печать таблиц результатов»).
3. Имена файлов для загрузки/сохранения данных должны задаваться с применением стандартных диалоговых компонент (OpenDialog, SaveDialog). Вывод данных предполагается как на экран, так и в текстовый файл (по выбору пользователя).
4. Дополнительное поле вычисляется согласно индивидуальному заданию (табл.1). Массив записей сортируется по заданному полю заданным методом (табл.2).
5. Столбчатая диаграмма должна отражать сравнение рейтингов по заданному предмету и RS (RM) для каждого студента (номера предметов приведены в табл.1, названия придумать самостоятельно). Примерный вид диаграммы приведен на рис.1. Построение диаграмм возможно лю-

бым способом (при помощи компоненты Chart, при помощи свойства Canvas компоненты PaintBox, путем размещения нескольких компонент Shape с автоматическим вычислением размеров и т.п.)

Таблица 1 – Варианты для расчета дополнительного поля

Вар.	Условие
0..8	Найти максимальный рейтинг RM по всем предметам у каждого студента
9..16	Найти минимальный рейтинг RM по всем предметам у каждого студента
17..27	Найти средний рейтинг RS по всем предметам у каждого студента

Таблица 2 – Варианты заданий

Вар.	Поле для сортировки	Алгоритм сортировки	Номер предмета
0,1	Фамилия	Сортировка обменами	Четвертый
2	Имя	Сортировка вставками	Пятый
3	Группа	Сортировка выбором	Первый
4	Год рождения	Быстрая сортировка Хоора	Второй
5	Средний рейтинг	Пирамида Уильямса-Флойда	Третий
6	Средний рейтинг	Сортировка обменами	Четвертый
7	Фамилия	Сортировка вставками	Пятый
8	Имя	Сортировка выбором	Первый
9	Группа	Быстрая сортировка Хоора	Второй
10	Год рождения	Пирамида Уильямса-Флойда	Третий
11	Год рождения	Сортировка обменами	Четвертый
12	Средний рейтинг	Сортировка вставками	Пятый
13	Фамилия	Сортировка выбором	Первый
14	Имя	Быстрая сортировка Хоора	Второй
15	Группа	Пирамида Уильямса-Флойда	Третий
16	Группа	Сортировка обменами	Четвертый
17	Год рождения	Сортировка вставками	Пятый
18	Средний рейтинг	Сортировка выбором	Первый
19	Фамилия	Быстрая сортировка Хоора	Второй
20	Имя	Пирамида Уильямса-Флойда	Третий
21	Имя	Сортировка обменами	Четвертый
22	Группа	Сортировка вставками	Пятый
23	Год рождения	Сортировка выбором	Первый
24	Средний рейтинг	Быстрая сортировка Хоора	Второй
25	Фамилия	Пирамида Уильямса-Флойда	Третий
26	Имя	Сортировка вставками	Пятый
27	Группа	Сортировка выбором	Первый

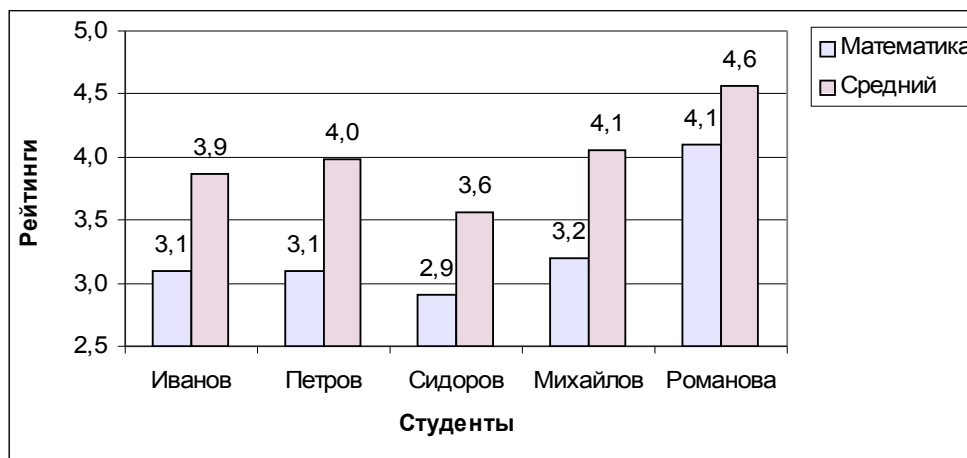


Рисунок 1 – Примерный вид результатов расчетов

Лабораторная работа № 2. Классы и объекты

Цель работы: закрепить знания об объектной модели, классах и объектах путем построения основной структурной диаграммы – диаграммы классов.

Задание к работе: для приложения, разработанного в ходе лабораторной работы №1, построить диаграмму классов.

Тема 2. Объектно-ориентированное программирование на языке C++

Лабораторная работа № 3. Программирование на языке C

Цель работы: восстановление навыков создания программ на языке программирования C.

Задание к работе: создать программы для решения задач согласно индивидуальному заданию (табл.3).

Таблица 3 – Варианты заданий

Вар.	Задание
1	2
1	Используя цикл с предусловием, найти сумму $y = \sum \frac{F_1}{F_2}$, где $a \leq x \leq b$, x меняется с шагом $h = c$. $F_1: 2\sqrt{x^3} \sin x^3$; $F_2: x^4 + 2x^3 - x$; $a = 0.3$; $b = 3.12$; $c = 0.15$
2	Используя цикл с постусловием, найти сумму $y = \sum \frac{F_1}{F_2}$, где $a \leq x \leq b$, x меняется с шагом $h = c$. $F_1: x^3 - \ln x$; $F_2: x^4 - x^{2-x}$; $a = 1.2$; $b = 13.4$; $c = 0.6$

Продолжение таблицы 3

1	2
3	Используя цикл с параметром, найти сумму $y = \sum \frac{F_1}{F_2}$, где $a \leq x \leq b$, x меняется с шагом $h = c$. $F_1: x^3 + 2x^2 - 2$; $F_2: x^{2x-1} + \cos x$; $a = 3.6$; $b = 7.2$; $c = 0.2$
4	Вычислить таблицу значений функции $y = \begin{cases} F_1(x), & \text{если } x \leq a; \\ F_2(x), & \text{если } x > a, \end{cases}$ для значений аргумента x в интервале от x_n до x_k с шагом h_x . $F_1: \arcsin \frac{x}{30}$; $F_2: \sqrt{ \ln x^2 }$; $x_n = 2.3$; $x_k = 8.9$; $h_x = 0.4$; $a = 5.4$
5	Дана функция $y = \begin{cases} f_1(x), & \text{если } x < 0, \\ f_2(x), & \text{если } 0 \leq x \leq 1, \\ f_3(x), & \text{если } x > 1, \end{cases}$ Методом перебора найти экстремумы данной функции на отрезке. Начальное и конечное значения отрезка, а также шаг табуляции задавать произвольно. $F_1: x^5 \operatorname{ctg} 2x^3$; $F_2: \ln(x+1)$; $F_3: e^{-2x} - \sqrt[3]{x}$
6	Дана функция $y = \begin{cases} f_1(x), & \text{если } x < 0, \\ f_2(x), & \text{если } 0 \leq x \leq 1, \\ f_3(x), & \text{если } x > 1, \end{cases}$ Методом перебора найти экстремумы данной функции на отрезке. Начальное и конечное значения отрезка, а также шаг табуляции задавать произвольно. $F_1: \operatorname{ctg}(3x-1)^2$; $F_2: 2 + xe^{-x}$; $F_3: \sin^3 x^2$
7	Найти суммы четных положительных элементов каждой строки матрицы $A(3,3)$ и сохранить их в одномерном массиве В
8	Найти суммы нечетных отрицательных элементов каждого столбца матрицы $A(3,3)$ и сохранить их в одномерном массиве В
9	Составить программу с обязательным использованием подпрограммы для ввода матрицы с экрана, ее обработки и вывода на экран. Задание: из каждого элемента матрицы $A(3,3)$ вычесть сумму ее нечетных положительных элементов
10	Составить программу с обязательным использованием подпрограммы для ввода матрицы с экрана, ее обработки и вывода на экран. Задание: каждый элемент матрицы $A(3,3)$ разделить на сумму ее четных положительных элементов

Продолжение таблицы 3

1	2
11	Составить программу с обязательным использованием подпрограммы для ввода матрицы с экрана, ее обработки и вывода на экран. Задание: каждый элемент матрицы $A(3,3)$ разделить на произведение ее нечетных отрицательных элементов
12	Составить программу, которая вводит строку символов, выполняет ее обработку в соответствии с заданием и выводит результаты. Задание: удалить все символы, не являющиеся цифрами
13	Составить программу, которая вводит строку символов, выполняет ее обработку в соответствии с заданием и выводит результаты. Задание: заменить все восклицательные знаки («!») на символ «*», а символ «точка» («.») – многоточием (три точки «...»)
14	Составить программу, которая вводит строку символов, выполняет ее обработку в соответствии с заданием и выводит результаты. Задание: удалить из строки все удвоенные, строенные и т.д. символы
15	Составить программу, которая вводит строку символов, выполняет ее обработку в соответствии с заданием и выводит результаты. Задание: вставить пробел после каждого символа «.» «,» «!» или «?», если за этими символами не следует пробел (т.е. следует любой символ, кроме пробела)
16	Сформировать файл из модулей целых чисел, найти сумму квадратов четных компонентов
17	Приняв, что координаты точек на плоскости задаются двумя числами x и y , составить программу, которая вводит с клавиатуры координаты точек и записывает их последовательно в файл: сначала x , а затем y . После завершения ввода осуществляется просмотр файла и его обработка: найти сумму расстояний каждой точки от центра координат
18	Сформировать файл из чисел последовательности $(-1)^k \cdot 0.2^k / k$. Найти наибольший из компонентов файла
19	Создать файл, содержащий следующую структуру данных: «Фамилия студента; Наименование группы; Место жительства; Место рождения; Количество братьев и сестер». Выбрать из файла и выдать на экран список студентов, проживающих не там, где родились (сменивших место жительства)
20	Создать файл, содержащий следующую структуру данных: «Фамилия студента; Наименование группы; Дата рождения; Средний рейтинг». Выбрать из файла и выдать на экран список студентов заданной группы, имеющих средний рейтинг не ниже 4,5
21	Создать файл, содержащий следующую структуру данных: «Номер книги (код, шифр); Фамилия автора; Наименование книги; Год издания; Количество страниц». Выбрать из файла и выдать на экран список книг, изданных ранее заданного года

Продолжение таблицы 3

1	2
22	<p>С использованием модуля работы с графикой составить программу для отрисовки изображения:</p> 
23	<p>С использованием модуля работы с графикой составить программу для отрисовки изображения:</p> 
24	<p>Сформировать динамический список «Стек» (LIFO) структур (не менее 5), содержащий данные по студентам в следующем виде: «<u>Фамилия Имя Группа ГР RS</u>», где ГР – год рождения, RS – средний рейтинг. Вывести на экран данные обо всех студентах, чьи имена начинаются с буквы «Н»</p>
25	<p>Сформировать динамический список «Очередь» (FIFO) структур (не менее 5), содержащий данные по студентам в следующем виде: «<u>Фамилия Имя Группа ГР RS</u>», где ГР – год рождения, RS – средний рейтинг. Вывести на экран данные обо всех студентах старше 18 лет</p>

Примеры выполнения заданий

8 Найти экстремумы функции

$$y = \begin{cases} \sqrt{\sin(x)} + 1, & \text{если } x \geq 0; \\ x^2 + 2x + 3, & \text{если } x < 0 \end{cases}$$

на интервале изменения аргумента от $-\pi$ до π .

```

#include <stdio.h>
#include <math.h>
void main()
{
    float Pi=M_PI;
    float x,y,xn=-Pi,xk=Pi,xh=Pi/3,min=1E+10,max=-1E+10,xmin,xmax;
    printf("\n X Y\n");
    for (x=xn;x <= xk;x+=xh)
    {
        if (x < 0) {y=pow(x,2)+2*x+3;} else {y=sqrt(sin(x)+1);}
        printf("%8.5f %8.5f\n",x,y);
        if (y > max) {max=y; xmax=x;}
        if (y < min) {min=y; xmin=x;}
    }
    printf("Минимум= %8.5f при x=%8.5f\n",min,xmin);
    printf("Максимум= %8.5f при x=%8.5f\n",max,xmax);
}

```

2 Найти минимальные элементы каждого столбца матрицы $A(3,3)$ и сохранить их в одномерном массиве B

```

#include <stdio.h>
#include <math.h>
void main()
{
    int min,I,j,b[3];
    int a[3][3]={4,3,1,6,2,7,1,9,3};
    printf(«Исходная матрица:\n»);
    for (i=0;i<3;i++) {
        for (j=0;j<3;j++) {printf(«%3i»,a[i][j]);}
        printf(«\n»);}
    for (j=0;j<3;j++) { min=a[0][j];
        for (i=1;i<3;i++) {min=a[i][j]<min?a[i][j]:min;}
        b[j]=min;}
    printf(«\nРезультат:\n»);
    for (j=0;j<3;j++) {printf(«%3i»,b[j]);} printf(«\n»);
}

```

3 К каждому элементу матрицы $M(3,3)$ прибавить сумму ее нечетных отрицательных элементов.

```

#include <stdio.h>
#include <conio.h>
void vvod(int a[3][3])
{

```

```

printf("Введите матрицу:\n");
for (int i=0;i<3;i++) {for (int j=0;j<3;j++) {scanf("%i",&a[i][j]);}}
}
void vyvod(int a[3][3])
{
for (int i=0;i<3;i++) {
for (int j=0;j<3;j++) {printf("%4i",a[i][j]);}
printf("\n");}
}
int summa(int a[3][3])
{
int sum=0;
for (int i=0;i<3;i++) {
for (int j=0;j<3;j++) {
if ((a[i][j] < 0) && (a[i][j]%2!=0)) {sum+=a[i][j];}
}}
return sum;
}
void work(int s,int a[3][3])
{
for (int i=0;i<3;i++) {
for (int j=0;j<3;j++) {a[i][j]+=s;}}
}
void main()
{
int m[3][3];
clrscr(); vvod(m);
printf("\nИсходная матрица:\n"); vyvod(m);
int s=summa(m);
printf("\nСумма = %3i\n",s); work(s,m);
printf("\nРезультат:\n"); vyvod(m);
getch();
}

```

4 Удалить из заданной строки символов все цифры.

```

#include <stdio.h>
#include <conio.h>
#include <string.h>
void main()
{
int l,k;char *st="";
clrscr();
printf("Введите строку символов:\n"); scanf("%s",st);
printf("\nИсходная строка:\n%s\n",st); i=0;

```

```

while (i<strlen(st)) {
if ((* (st+i)>='0')&&(* (st+i)<='9')) {
for (k=1;k<=strlen(st);k++) {
*(st+k)=*(st+k+1);}}
else {i++;}
}
printf("Результат:\n%s\n",st); getch();
}

```

5 Сформировать файл из модулей целых чисел, найти сумму четных компонентов файла. Вывод результатов продублировать в текстовый файл.

```

#include <stdio.h>
#include <conio.h>
#include <math.h>
void main(void)
{
FILE *f,*g;
// запись
f=fopen("lab8.dat","w");
printf("Вводите целые числа, признак конца – 0:\n");
int k;
do { scanf("%i",&k); if (k!=0) {fwrite(&k,sizeof(k),1,f);}} while (k);
fclose(f);
// чтение
printf("Содержимое файла:\n");
if ((f=fopen("lab8.dat","r"))==NULL) {
printf("Ошибка при чтении файла!\n");return;}
g=fopen("lab8.txt","w");
fprintf(g,"Содержимое файла:\n");
int s=0;
do { fread(&k,sizeof(k),1,f);
if (!feof(f)) {printf("%4i",k);fprintf(g,"%4i",k);if (!(k%2)) {s+=k;}}}
while (!feof(f));
printf("\nCумма = %i\n",s);fprintf(g,"\nCумма = %i\n",s);
fclose(f);fclose(g);getch();
}

```

6 Сформировать файл «список студентов» форматом «фамилия-группа-год_рождения-5_рейтингов», предварительно вычислив дополнительное поле – максимальный рейтинг. Вывести на экран и в текстовый файл список студентов заданной группы.

```

#include <stdio.h>
#include <conio.h>

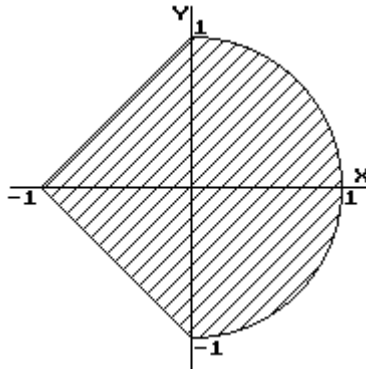
```

```

#include <string.h>
void main(void)
{
    struct stud {char fam[10],gr[10]; int year,rt[5],max;};
    stud ek[3]={{"Иванов"},"ЭК-03-1",1986,{70,55,55,60,75}},
               {"Петров"},"ЭК-03-2",1986,{61,62,63,74,55}},
               {"Сидоров"},"ЭК-03-2",1986,{98,99,97,90,91}}, s1;
    // ищем максимальный рейтинг
    int I,j,max;
    for (i=0;i<3;i++) { max=ek[i].rt[1];
    for (j=0;j<5;j++) { if (ek[i].rt[j] > max) {max=ek[i].rt[j];}}
    ek[i].max=max;}
    FILE *f,*g;
    // запись в файл
    f=fopen("rgr2.dat","w");
    for (i=0;i<3;i++) { fwrite(&ek[i],sizeof(stud),1,f);} fclose(f);
    // чтение
    printf("Содержимое файла:\n");
    if ((f=fopen("rgr2.dat","r"))==NULL) {
    printf("Ошибка при чтении файла!\n");return;}
    g=fopen("rgr2.txt","w"); fprintf(g,"Содержимое файла:\n");
    do { fread(&s1,sizeof(stud),1,f);
    if (!feof(f)) {
    printf("%12s%12s%6i ",s1.fam,s1.gr,s1.year);
    fprintf(g,"%12s%12s%6i ",s1.fam,s1.gr,s1.year);
    for (j=0;j<5;j++) {printf("%3i",s1.rt[j]);fprintf(g,"%3i",s1.rt[j]);}
    printf("%6i\n",s1.max);fprintf(g,"%3i\n",s1.max);
    }} while (!feof(f));
    // чтение с поиском
    char 16ROUP[10];
    printf("Введите группу для поиска:\n"); scanf("%s",&group);
    printf("Студенты группы %s:\n",group);
    fprintf(g,"Студенты группы %s:\n",group); fseek(f,0L,SEEK_SET);
    do { fread(&s1,sizeof(stud),1,f);
    if ((!feof(f))&&(!strcmp(s1.gr,group))) {
    printf("%12s%12s%6i ",s1.fam,s1.gr,s1.year);
    fprintf(g,"%12s%12s%6i ",s1.fam,s1.gr,s1.year);
    for (j=0;j<5;j++) {printf("%3i",s1.rt[j]);fprintf(g,"%3i",s1.rt[j]);}
    printf("%6i\n",s1.max);fprintf(g,"%3i\n",s1.max);
    }} while (!feof(f));
    //printf("\nCумма = %i\n",s);fprintf(g,"\nCумма = %i\n",s);
    fclose(f);fclose(g);getch();
}

```


7 Нарисовать изображение



```
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
void axes(int xc, int yc)
{
    line(xc-100,yc,xc+100,yc); line(xc+100,yc,xc+95,yc-2);
    line(xc+100,yc,xc+95,yc+2); line(xc,yc-100,xc,yc+100);
    line(xc,yc-100,xc-2,yc-95); line(xc,yc-100,xc+2,yc-95);
}
void main()
{
    int xc,yc,*gd,*gm;
    *gd=0;
    initgraph(gd,gm,»»»);
    if (graphresult() !=0 ) {printf(«Error!»);}
    xc=200;yc=200;axes(xc,yc); setfillstyle(3,getmaxcolor());
    pieslice(xc,yc,0,90,50); pieslice(xc,yc,270,360,50);
    line(xc,yc-50,xc-50,yc);floodfill(xc-2,yc-2,getmaxcolor());
    line(xc-50,yc,xc,yc+50);floodfill(xc-2,yc+2,getmaxcolor());
    getch(); closegraph();
}
```

8 Сформировать связный список структур, содержащий данные по студентам. Вывести на экран список студентов, у которых фамилии начинаются на букву А.

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
struct stud { char fam[10],name[10],group[10]; int gr,rs;};
struct dstud {
    stud data;
    dstud *pPrior;
    dstud *pNext;
};
```

```

void sread(stud &s)
{
printf("Family: \n");scanf("%s",&s.fam);
if (s.fam[0] != '*') {
printf("Name: \n");scanf("%s",&s.name);
printf("Group: \n");scanf("%s",&s.group);
printf("Year: \n");scanf("%i",&s.gr);
printf("Rating: \n");scanf("%i",&s.rs);
}}
void main(void)
{
dstud *pBegin=NULL,*pEnd=NULL,*pList=NULL;
stud s;
int k=0;
// Создание списка
clrscr();
pList=new(dstud);
(*pList).pPrior=NULL;
(*pList).pNext=NULL;
sread(s);(*pList).data=s;
pBegin=pList;
// Добавление данных
while (s.fam[0] != '*') {
sread(s);
if (s.fam[0] != '*') {
pEnd=new(dstud);
(*pEnd).pPrior=pList;
(*pEnd).pNext=NULL;
(*pEnd).data=s;
(*pList).pNext=pEnd;
pList=pEnd;
}}
printf("Весь список:\n");
pList=pBegin;
while (pList) {
printf("F=%s N=%s G=%s Y=%i Rs=%i\n",(*pList).data.fam,
(*pList).data.name,(*pList).data.group,(*pList).data.gr,(*pList).data.rs);
pList=(*pList).pNext;
}
printf("Требуемые студенты:\n");
pList=pBegin;
while (pList) {
if ((*pList).data.fam[0]=='A') {
k++;

```

```

printf("F=%s N=%s G=%s Y=%i Rs=%i\n",(*pList).data.fam,
(*pList).data.name,(*pList).data.group,(*pList).data.gr,(*pList).data.rs);
}
pList=(*pList).pNext;
}
printf("Всего найдено %i студентов.",k);
}

```

Лабораторная работа № 4. Перегрузка функций

Цель работы: изучить такое понятие объектно-ориентированного программирования, как «перегрузка функций».

Задание к работе: создать подпрограмму-функцию, позволяющую выполнять действия согласно индивидуальному заданию (табл. 4). Во всех случаях, где необходимо, использовать перегрузку функций.

Таблица 4 – Варианты заданий

№	Задание
1	2
1	Нахождение суммы трех целых чисел или двух вещественных
2	Нахождение суммы трех целых чисел или разности двух вещественных
3	Нахождение суммы четырех целых чисел или произведения трех вещественных
4	Нахождение произведения трех целых чисел или двух вещественных
5	Нахождение произведения трех целых чисел или суммы двух вещественных
6	Нахождение суммы трех целых чисел или разности двух
7	Нахождение суммы четырех целых чисел или произведения трех
8	Нахождение произведения трех целых чисел или суммы двух
9	Нахождение среднего арифметического трех целых чисел или суммы двух вещественных
10	Нахождение среднего арифметического трех целых чисел или суммы двух вещественных
11	Нахождение среднего арифметического двух целых чисел или произведения трех вещественных
12	Нахождение среднего арифметического двух целых чисел или произведения трех вещественных
13	Нахождение среднего геометрического трех целых чисел или суммы двух вещественных
14	Нахождение среднего геометрического трех целых чисел или произведения двух вещественных

Продолжение таблицы 4

1	2
15	Нахождение среднего геометрического трех целых чисел или суммы двух вещественных
16	Нахождение среднего геометрического трех целых чисел или произведения двух вещественных
17	Нахождение суммы произвольного набора целых чисел
18	Нахождение суммы произвольного набора вещественных чисел
19	Нахождение произведения произвольного набора целых чисел
20	Нахождение произведения произвольного набора вещественных чисел
21	Нахождение среднего арифметического произвольного набора целых чисел
22	Нахождение среднего арифметического произвольного набора вещественных чисел
23	Нахождение среднего геометрического произвольного набора целых чисел
24	Нахождение среднего геометрического произвольного набора вещественных чисел
25	Нахождение суммы произвольного набора целых чисел

Примеры выполнения заданий

1 Создать функцию для нахождения минимального из трех целых чисел или максимального из двух вещественных (с использованием «перегрузки»).

```
#include <stdio.h>
int minimax(int a, int b, int c)
{
    int m;
    m=a<b?a:b;
    m=m<c?m:c;
    return m;
}
float minimax(float a, float b)
{
    float m;
    m=a>b?a:b;
    return m;
}
void main()
{
    int k;
    k=minimax(6,1,9);
```

```
printf("Минимальное целое = %i\n",k);
float x;
x=minimax(6.27,5.98);
printf("Максимальное вещественное = %5.2f\n",x);
}
```

2 Создать функцию для нахождения произведения произвольного набора целых чисел.

```
#include <stdio.h>
int proizv1(int a,...)
// 0 - признак окончания списка параметров
{
    int pr=1,*p=&a;
    while (*p) {pr*=*(p++);}
    return pr;
}
int proizv2(int a,...)
// Первый параметр - число параметров
{
    int pr=1,*p=&a;
    while (a--) {pr*=*(++p);}
    return pr;
}
void main()
{
    int p1=proizv1(2,3,4,0),p2=proizv2(3,2,3,4);
    printf("P1= %i\nP2= %i\n",p1,p2);
}
```

Лабораторная работа № 5. Передвижение графических объектов на языке C++

Цель работы: изучить основные определения объектно-ориентированного программирования на примере программы для перемещения графических объектов

Задание к работе: составить программу для решения задачи (табл. 5). Программа должна содержать новые классы (включая абстрактные), конструкторы и деструкторы, поддерживать наследование, полиморфизм и видимость компонент.

Таблица 5 – Варианты заданий

Вар.	Задание
1	2
1	Нарисовать на экране десять концентрических окружностей
2	Нарисовать в центре экрана десять вложенных друг в друга прямоугольников
3	Переделать программу «кругов на воде» (см. пример) так, чтобы круги сначала расходились из одной точки, а потом обратно в нее сходились
4	Перемещать круг по горизонтали с заданным шагом и задержкой в одну секунду
5	Перемещать круг по вертикали с заданным шагом и задержкой в одну секунду
6	Перемещать прямоугольник по горизонтали с заданным шагом и задержкой в одну секунду
7	Перемещать прямоугольник по вертикали с заданным шагом и задержкой в одну секунду
8	Переделать программу «кругов на воде» (см. пример), заменив круги на прямоугольники
9	Перемещать круг по диагонали с заданным шагом и задержкой в полторы секунды
10	Перемещать прямоугольник по диагонали с заданным шагом и задержкой в полторы секунды
11	Разместить на экране две расположенные рядом «перемигивающиеся» окружности
12	Перемещать круг, вписанный в прямоугольник, по горизонтали с заданным шагом и задержкой в одну секунду
13	Перемещать круг, вписанный в прямоугольник, по вертикали с заданным шагом и задержкой в одну секунду
14	Перемещать круг, вписанный в прямоугольник, по диагонали с заданным шагом и задержкой в полторы секунды
15	Перемещать круг по периметру расположенного в центре экрана прямоугольника
16	Перемещать круг по вершинам расположенного в центре экрана прямоугольника
17	Нарисовать треугольник произвольной формы и перемещать круг по его периметру
18	Нарисовать треугольник произвольной формы и перемещать круг по его вершинам
19	Перемещать треугольник по горизонтали с заданным шагом и задержкой в одну секунду
20	Перемещать треугольник по вертикали с заданным шагом и задержкой в одну секунду
21	Перемещать треугольник по диагонали с заданным шагом и задержкой в полторы секунды

Продолжение таблицы 5

1	2
22	Перемещать круг по экрану случайным образом
23	Перемещать прямоугольник по экрану случайным образом
24	Перемещать треугольник по экрану случайным образом
25	Сделать так, чтобы на экране поочередно возникали и исчезали круг, прямоугольник и треугольник

Пример выполнения задания

Изобразить на экране расходящиеся концентрические окружности («круги на воде»).

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <graphics.h>
class gr_init
{
public:
    gr_init(int driver=0)
    {
        *gd=driver;
        initgraph(gd,gm,"");
        if (graphresult() !=0 ) {printf("Error!");abort();}
    }
    ~gr_init()
    {
        closegraph();
    }
private:
    int *gd,*gm;
};
class gr_obj
{
public:
    gr_obj(int col=7)
    {_color=col;}
    virtual void draw()=0;
    int color()
    {
        return _color;
    }
public:
```

```

void show()
{
    setcolor(color()); draw();
}
void hide()
{
    cback=getcolor(); setcolor(getbkcolor()); draw(); setcolor(color());
}
private:
    int _color,cback;
};
class point: public gr_obj
{
public:
    point(int xp=0, int yp=0, int col=7): gr_obj(col)
    {setpx(xp); setpy(yp); show();}
    ~point()
    {
        hide();
    }
    void draw()
    {
        putpixel(x,y,color());
    }
    int getpx()
    {
        return x;
    }
    void setpx(int px)
    {
        x=px;
    }
    int getpy()
    {
        return y;
    }
    void setpy(int py)
    {
        y=py;
    }
private:
    int x,y;
};

```



```

class krug: public point
{
public:
    krug(int xc=0, int yc=0, int rad=0, int col=7): point(xc,yc,col)
    {r=rad; show();}
    ~krug()
    {
        hide();
    }
    void draw()
    {
        circle(getpx(),getpy(),getrad());
    }
    void move(int newx, int newy, int newr)
    {
        hide();
        setpx(newx); setpy(newy); setrad(newr);
        show();
    }
    int getrad()
    {
        return r;
    }
    void setrad(int rad)
    {
        r=rad;
    }
private:
    int r;
};

void main()
{
    gr_init gr(0);
    krug kr(300,200,100);
    for (int i=1; i<=10; i++) {
        kr.move(300,200,10+i*15); delay(200); }
    getch();
}

```

Лабораторная работа № 6. Построение диаграммы классов

Цель работы: научиться проведению объектно-ориентированного анализа и закрепить навыки построения диаграммы классов.

Задание к работе: провести объектно-ориентированный анализ задачи из лабораторной работы № 5 и построить для нее диаграмму классов.

Пример выполнения задания

1 Объектно-ориентированный анализ задачи

В рассматриваемой области можно выделить следующие классы:

1 **Графический режим**. Свойства: графический драйвер (gd), графический режим (gm). Методы: открытие (конструктор), закрытие (деструктор).

2 **Графический объект**. Свойства: цвет линий, цвет фона. Методы: возвращение цвета линий, нарисовать объект, показать объект, скрыть объект. «Нарисовать объект» – чисто виртуальная функция, точное содержание которой будет определено в наследниках класса. «Показать объект» устанавливает цвет линий и вызывает «Нарисовать объект»; «Скрыть объект» устанавливает цвет линий идентичным цвету фона и опять вызывает «Нарисовать объект». Поскольку данный класс служит основой для создания иерархии графических классов, его можно считать абстрактным (изображается курсивом).

3 **Точка** (наследник «Графического объекта»). Новые свойства: координаты. Новые методы: возвращение и установка значений координат, нарисовать объект.

4 **Круг** (наследник «Точки»). Новое свойство: радиус. Новые методы: возвращение и установка значений радиуса, нарисовать объект.

2 Построение диаграммы классов

Диаграмма классов представлена на рис. 2.

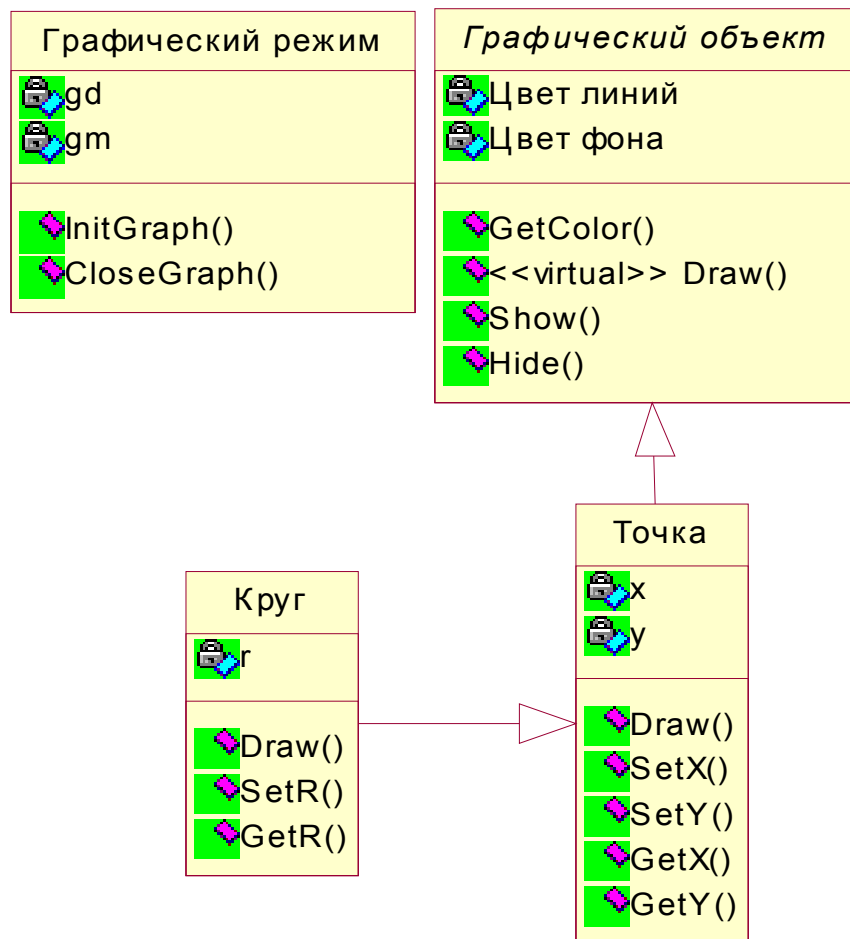


Рисунок 2 – Диаграмма классов

СПИСОК РЕКОМЕНДОВАННОЙ ЛИТЕРАТУРЫ

- 1 **Мельников, А. Ю.** Программирование и алгоритмические языки : конспект лекций для студентов направления «Системный анализ» всех форм обучения. Часть 2 / Сост. А.Ю. Мельников – Краматорск : ДГМА, 2012. – 33 с.
- 2 **Мельников, А. Ю.** Объектно-ориентированный анализ и проектирование информационных систем : учебное пособие для студентов специальностей «Экономическая кибернетика» и «Интеллектуальные системы принятия решений» / А. Ю. Мельников. – Краматорск : ДГМА, 2006. – 184 с. – ISBN 966-379-103-9.
- 3 **Буч, Гради.** Объектно-ориентированный анализ и проектирование с примерами приложений на С++ / Гради Буч. – 2-е изд. / Пер. с англ. – М. : Изд-во «Бином»; СПб.: Невский диалект, 2001. – 560 с.
- 4 **Подбельский, В. В.** Язык С++ : учебное пособие / В. В. Подбельский. – 5-е изд. – М. : Финансы и статистика, 2001. – 560 с. – ISBN 5-279-02204-7.
- 5 **Страуструп, Б.** Язык программирования С++ / Б. Страуструп. – М. : Радио и связь, 1991. – 352 с.
- 6 **Павловская, Т. А.** С/С++. Программирование на языке высокого уровня / Т. А. Павловская. – СПб. : Питер, 2001. – 464с. – ISBN 5-318-00001-0.
- 7 **Павловская, Т. А.** С/С++. Структурное программирование : практикум / Т. А. Павловская, Ю. А. Щупак. – СПб. : Питер, 2002. – 240 с. – ISBN 5-94723-447-5.
- 8 **Мельников, А. Ю.** Алгоритмизация и программирование : учебное пособие для студентов специальности «Интеллектуальные системы принятия решений» / А. Ю. Мельников. – Издание 2-е, с изменениями. – Краматорск: ДГМА, 2010. – 96 с. – ISBN 978-966-379-437-2.