



**Некоммерческое
акционерное
общество**

**АЛМАТИНСКИЙ
УНИВЕРСИТЕТ
ЭНЕРГЕТИКИ
И СВЯЗИ**

Кафедра
компьютерных
технологий

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

Методические указания по выполнению лабораторных работ
для студентов специальности
5В070400 – Вычислительная техника и программное обеспечение

Алматы 2014

СОСТАВИТЕЛИ: Бимурзаев С.Б., Турганбаев Е.С. Объектно-ориентированное программирование. Методические указания к выполнению лабораторных работ для студентов специальности 5В070400 – Вычислительная техника и программное обеспечение. – Алматы: АУЭС, 2012. – 65 с.

Методические указания посвящены изучению основ объектно-ориентированного программирования в свободной интегрированной среде разработки на языках С/С++. В них рассмотрены такие вопросы, объекты и классы, инкапсуляция, наследование, конструкторы и деструкторы, средства ввода и вывода, использование статических массивов объектов и массивом указателей на объекты, а также построение диаграмм классов в среде Visual Studio.

Методические указания предназначены для студентов всех форм обучения по специальности 5В070400 – Вычислительная техника и программное обеспечение. Они могут быть использованы и для организации лабораторных занятий по идентичным темам аналогичных дисциплин, запланированных для других специальностей.

Ил. - 22, рисунков - 22, библиогр. – 5 назв.

Рецензент: доц. А.А.Куликов

Печатается по плану издания Некоммерческое акционерное общество «Алматинский университет энергетики и связи» на 2012 г.

© НАО «Алматинский университет энергетики и связи», 2014 г.

1 Лабораторная работа №1. Организация консольного ввода/вывода в ++»

Цели работы: основная цель: ознакомиться с организацией консольного ввода и вывода информации в C++ посредством объектов `cin`, `cout`.

Дополнительные цели:

- 1) Освоить основные приемы работы с интегрированной средой разработки DevC++.
- 2) Повторить операторы управления языка C.
- 3) Ознакомиться с концепцией пространства имен.

1.1 Практическая часть

Задание 1.

Ознакомиться со средой разработки Dev-C++ и составить программу «Приветствие».

1. Запустить среду разработки Dev-C++. Выбрать пункт меню:

Файл->Создать->Исходный файл:

2. Ввести текст программы:

```
//#include<iostream>      /* подключение заголовочного
файла, содержащего описание классов,
//                          прототипов функций и
предопределенных констант, необходимых для
//                          организации ввода/вывода на
языке C++ */
using namespace std;      // использование стандартного
пространства имен
main()                    // заголовок головной
программы (главной функции)
3. {
    cout<<"Hello, Comp"<<endl;
    cout<<"Hello, student"<<endl; // вывод сообщений с
помощью объекта <cout>
    system("pause");       // вызов
системной команды <pause>
}
```

4. Выполнить компиляцию программы (рисунок 1.1).

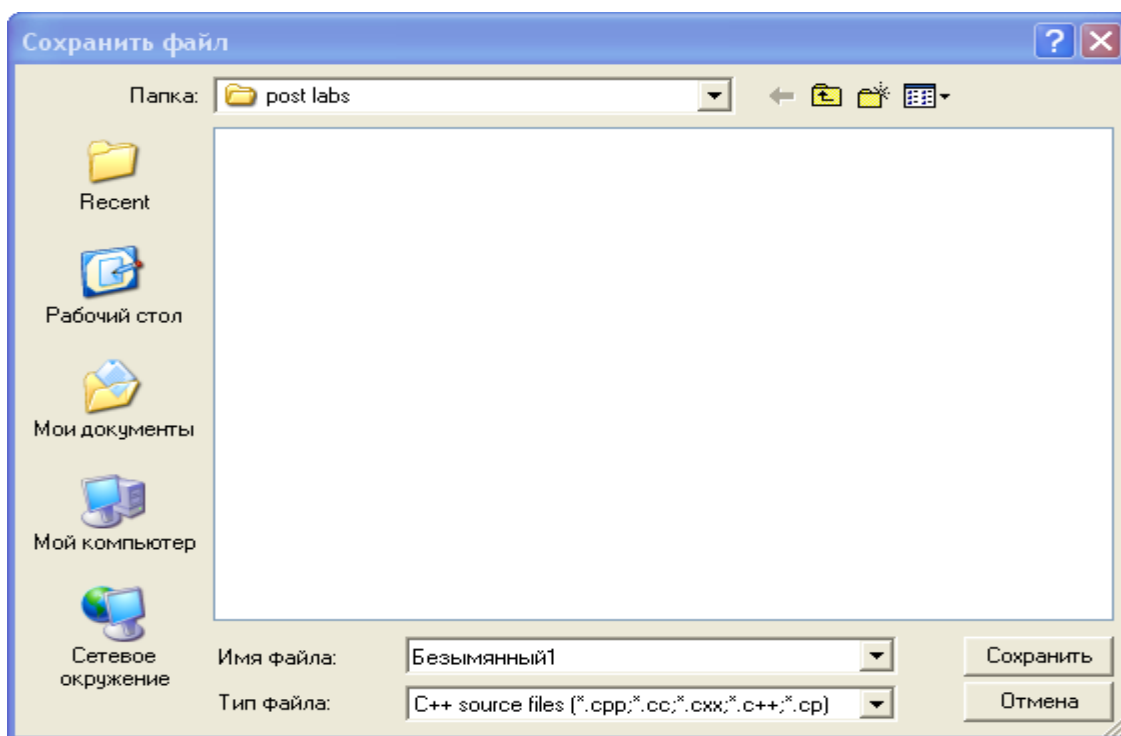


Рисунок 1.1 - Компиляция программы

5. Убедиться в отсутствии явных ошибок (рисунок 1.2):

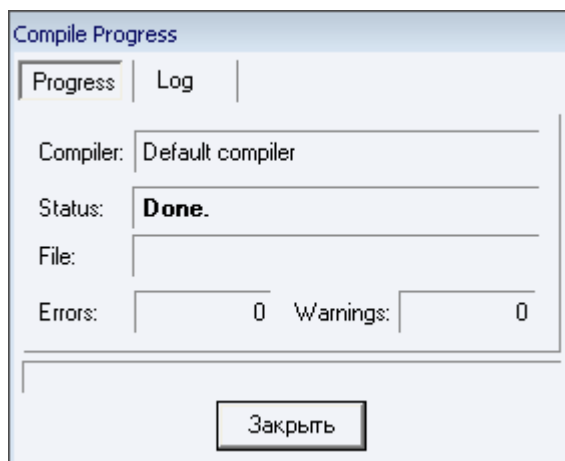
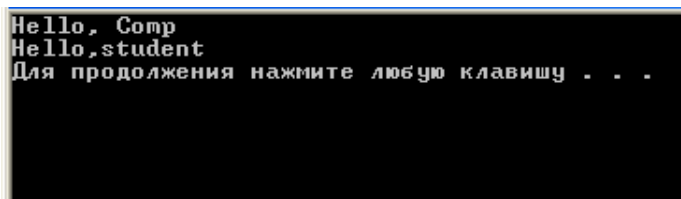


Рисунок 1.2 - Результат компиляции

6. Выполнить программу и получить результат выполнения программы на экране пользователя (рисунок 1.3):



```
Hello, Comp
Hello, student
Для продолжения нажмите любую клавишу . . .
```

Рисунок 1.3 - Вид экрана

Задание 2.

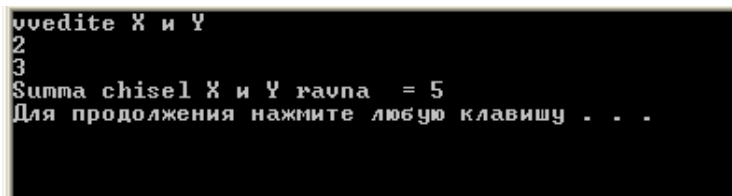
Найти сумму двух целых чисел.

Алгоритм:

- 1) Ввести с консоли числа X, Y.
- 2) Вычислить Summa по формуле : $Summa = X + Y$.

Код программы

```
#include <iostream> // заголовочный файл
стандартного ввода/вывода
using namespace std; // использование стандартного
пространства имен
main()
{
int X, Y, Summa;
cout<<"Введите X и Y"<<endl;
cin>>X>>Y; // Ввод X, Y
Summa=X+Y;
cout<<"Summa chisel X и Y равна = "<<Summa<<endl;
// Вывод результата
system("pause");
```



```
Введите X и Y
2
3
Summa chisel X и Y равна = 5
Для продолжения нажмите любую клавишу . . .
```

Рисунок 1.4 - Вид экрана

Задание 3.

Составить программу, вычисляющую значение переменной m при данных i, j, k.

Алгоритм:

1) Ввести с консоли числа I, J, K.

2) Вычислить m по формуле: $M = \frac{i+j}{k+1}$.

Код программы

```
#include<iostream>

main()
{
float i, j, k;
float m;
cout<<"vvedite i, j, k"<<endl;
cin>>i>>j>>k;
m=(i+j)/(k+1);
cout.precision(3);
cout<<"m= "<<m<<endl;
system("pause");
}
```

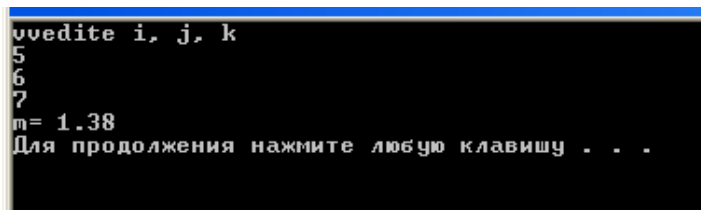


Рисунок 1.5 - Вид экрана

Задание 4.

Разработать программу для пересчета расстояния в милях в километры.

Алгоритм:

1.Получить расстояние в милях.

2.Перевести расстояние в км (расстояние в км равно 1,609 умножить на расстояние в милях).

3.Вывести расстояние в км.

Код программы

```
// Упражнение 4 - программа перевода миль в километры
#include <iostream>
using namespace std; // использование стандартного пространства имен
#define KMS_PER_MILE 1.609 /* константа преобразования*/
```

```

/*main()
{
float miles,      // расстояние в милях
kms;  // эквивалентное расстояние километрах
cout<<"Enter the distance in miles"<<endl;
cin>>miles;  // Получить расстояние в милях
kms=    KMS_PER_MILE*miles;      // Преобразовать
расстояние в километры
cout<<"That equals "<<kms<<" kilometers "<<endl; //
Вывести расстояние в километрах
system("pause");
}

```

```

Enter the distance in miles
10
That equals 16.09 kilometers
Для продолжения нажмите любую клавишу . . .

```

Рисунок 1.6 - Вид экрана

1.2 Индивидуальные задания

Задание 1.

Общая часть.

Разработать программу, обеспечивающую: ввод входных параметров, вычисление по формулам, вывод результата на экран. Справочная информация в [2, 3].

Варианты заданий.

1. Дан радиус окружности, подсчитать длину окружности.
2. Дан радиус окружности, подсчитать площадь круга.
3. Дан прямоугольный треугольник с катетами a и b . Найти гипотенузу c .
4. Дан произвольный треугольник. Известны стороны a и b и угол между ними. Найти третью сторону c .
5. Дан произвольный треугольник со сторонами a , b и c . Найти площадь треугольника.
6. Вычислить объём шара радиуса R .
7. Найти среднее арифметическое и среднее геометрическое трёх заданных чисел.
8. Найти расстояние между двумя точками с данными координатами.
9. По ребру найти площадь грани, площадь боковой поверхности и объём куба.
10. Вычислить периметр и площадь правильного 10-угольника, вписанного в окружность заданного радиуса.

000. Даны два действительных числа a и b . Получить их сумму, разность и произведение.

Пример выполнения индивидуального задания для варианта 000.

Постановка задачи.

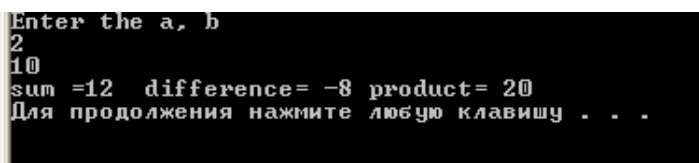
Даны два действительных числа a и b . Получить их сумму, разность и произведение.

Алгоритм:

- 1) Ввести с консоли числа a , b .
- 2) Провести расчет.
- 3) Вывести результат расчета на экран.

Код программы

```
// Задание 1 - Даны два действительных числа a и b.
Получить их сумму, разность и произведение.
#include <iostream>
using namespace std; // использование стандартного
пространства имен
main()
{
float a, b;
cout<<"Enter the a, b"<<endl;
cin>>a>>b;
cout<<"sum ="<<a+b<<" difference= "<<a-b<<" product=
"<<a*b<<endl; // Вывести расстояние в километрах
system("pause");
}
```



```
Enter the a, b
2
10
sum =12 difference= -8 product= 20
Для продолжения нажмите любую клавишу . . .
```

Рисунок 1.7 - Вид экрана

Задание 2.

Общая часть.

Разработать программу, обеспечивающую: ввод входных параметров с консоли, использование условных операторов, вывод результата на экран

Варианты заданий.

- 1) Даны действительные числа x, y . Получить:
 - а) $\max(x, y)$;
 - б) $\min(x, y)$;
 - в) $\max(x, y), \min(x, y)$.

2) Даны действительные числа x, y, z . Вычислить:

а) $\max(x + y + z, x \cdot y \cdot z)$;

б) $\min^2(x + y + z/2, x \cdot y \cdot z) + 1$.

3) Даны действительные числа a, b, c . Проверить выполняется ли неравенство $a < b < c$.

4) Найти \min значение из трёх величин, определяемых арифметическими выражениями $a = \sin(x)$, $b = \cos(x)$, $c = \ln(x)$ при заданных значениях x .

5) Даны действительные числа a, b, c . Удвоить эти числа, если $a > b > c$ и заменить их абсолютными значениями, если это не так.

6) Даны два действительных числа. Заменить первое число нулём, если оно меньше или равно второму, и оставить числа без изменения иначе.

7) Даны действительные числа x, y . Меньшее из этих двух чисел заменить их полусуммой, а большее – их удвоенным произведением.

8) Даны действительные числа a, b, c, d . Если $a < b < c < d$, то каждое число заменить наибольшим из них; если $a > b > c > d$, то числа оставить без изменения; иначе все числа заменяются их квадратами.

9) Даны действительные числа a, b, c . Выяснить, имеет ли уравнение $ax^2 + bx + c = 0$ действительные корни. Если действительные корни имеются, то найти их. В противном случае ответом должно служить сообщение, что действительных корней нет.

10) Даны два действительных числа. Вывести первое число, если оно больше второго, и оба числа если это не так.

000) Даны три действительных числа. Выбрать из них те, которые принадлежат интервалу (1,3).

Пример выполнения индивидуального задания для варианта 000.

Постановка задачи.

Даны три действительных числа. Выбрать из них те, которые принадлежат интервалу (1,3).

Алгоритм:

1) Ввести с консоли числа a, b, c .

2) Провести расчет.

3) Вывести результат расчета на экран.

Код программы

```
// Задание 2 - Даны три действительных числа.  
Выбрать из них те, которые принадлежат интервалу (1,3).  
#include <iostream>  
using namespace std; // использование стандартного  
пространства имен  
main()  
{
```

```

float a, b, c;
cout<<"Enter the a, b, c"<<endl;
cin>>a>>b>>c;
float x1=1., x2=3.;
if ( (a>x1) && (a<x2) ) cout<<"This number belongs to
the interval (1;3) ="<<a<<endl;
if ( (b>x1) && (b<x2) ) cout<<"This number belongs to
the interval (1;3) ="<<b<<endl;
if ( (c>x1) && (c<x2) ) cout<<"This number belongs to
the interval (1;3) ="<<c<<endl;
system("pause");
}

```

```

Enter the a, b, c
0.3
1.4
2.5
This number belongs to the interval (1;3) =1.4
This number belongs to the interval (1;3) =2.5
Для продолжения нажмите любую клавишу . . .

```

Рисунок 1.8 - Вид экрана

1.3 Контрольные вопросы

- 1) Структура консольного приложения.
- 2) Подключение заголовочных файлов.
- 3) Описание прототипов функций.
- 4) Назначение объекта `cout`.
- 5) Манипуляторы формата.
- 6) Условный оператор.
- 7) Оператор цикла.
- 8) Оператор выбора.
- 9) Средства консольного ввода.
- 10) Средства консольного вывода.

2 Лабораторная работа №2. Классы. Инкапсуляция

Цели работы: основная цель: ознакомиться с принципом инкапсуляции.

Дополнительные цели:

- 1) Повторить использование указателей в С.
- 2) Получить навыки программирования конструкторов и деструкторов.
- 3) Получить навыки использования статических элементов класса.

- 4) Обеспечить ввод/вывод символов кириллицы из DevC++.
- 5) Ознакомиться с концепцией пространства имен.

2.1 Практическая часть

Задание 1.

Обеспечить возможность вывода символов кириллицы на экран.

В консольных приложениях (похожих на текстовый режим) русские буквы выводятся в кодировке CP-866, а в оболочке Dev-C++ они набираются в кодировке CP-1251. Это значит, что вместо русских букв вы увидите «кракозябры». Чтобы этого не случилось, кодировку нужно исправить с помощью простой дополнительной программы. Сделайте следующее:

1) Скачайте архив, содержащий программы `gccrus.exe` и `g++rus.exe` (примерно 189 Кб), по следующей ссылке:

<http://rapidshare.com/files/245834998/devcpprus.zip>

2) Скопируйте программы `gccrus.exe` и `g++rus.exe` в папку C:

\Dev-Cpp\bin (если при установке оболочки вы указали другой каталог вместо стандартного C:\Dev-Cpp, скопируйте программы в его подкаталог bin).

3) Запустите оболочку Dev-C++ и войдите в меню Сервис-Параметры компилятора.

4) Перейдите на вкладку Программа и исправьте названия двух первых программ так, как показано на рисунке 2.1.

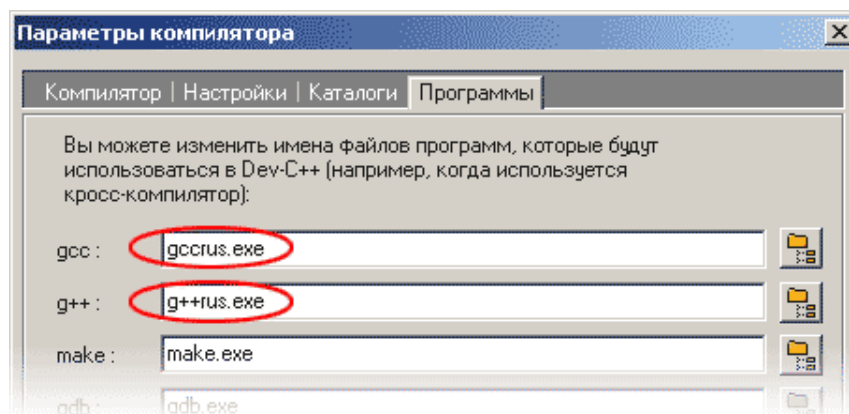


Рисунок 2.1- Настройка кириллицы

Теперь при выводе на экран можно смело использовать русские буквы, они будут «на лету» перекодироваться.

Задание 2.

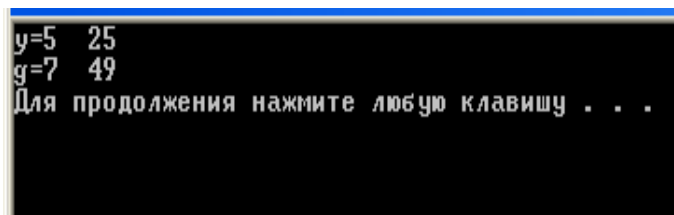
Инкапсуляция.

Определить класс «Квадрат» (внутренняя компонента – сторона; внешние компоненты для просмотра и установки полей данных и вычисления площади квадрата). Написать демонстрационную программу, иллюстрирующую работу с объектами класса «Квадрат» и принцип инкапсуляции.

```

Код программы
#include<iostream>
using namespace std;
// описание классов
class Tsquare
{
float a;
public:
void set_a(float);
float get_a();
float area();
};
// Реализация методов
void Tsquare::set_a(float s) {a=s;};
float Tsquare::get_a() {return a;};
float Tsquare::area() {return a*a;};
main()
{
Tsquare y, g;
y.set_a(5); g.set_a(7);
cout<<"y="<<y.get_a()<<" "<<y.area()<<endl;
cout<<"g="<<g.get_a()<<" "<<g.area()<<endl;
system("pause");
}

```



```

y=5 25
g=7 49
Для продолжения нажмите любую клавишу . . .

```

Рисунок 2.2- Вид экрана

Задание 3.

Конструктор, деструктор, статические элементы класса.

Определить в классе «Квадрат» статический элемент – «номер квадрата». Определить в классе «Квадрат» следующие конструкторы: без параметров, с параметрами. Определить в классе деструктор. Написать демонстрационную программу, в которой создаются и разрушаются объекты пользовательского класса и каждый вызов конструктора и деструктора сопровождается выдачей соответствующего сообщения (какой объект какой конструктор или деструктор вызвал).

```

Код программы
#include<iostream>
using namespace std;
// описание классов
class Tsquare
{
static int count;
float a;
public:
Tsquare();
Tsquare(float);
~Tsquare();
void set_a(float);
float get_a();
float area();
};
int Tsquare::count=0; //Инициализация статических элементов класса
// Реализация методов
Tsquare::Tsquare(){count++;
cout<<"Конструктор Tsquare: Создание экземпляра класса, число
элементов "<<count<<endl;};
Tsquare::Tsquare(float a1) {count++; a=a1;
cout<<"Конструктор Tsquare: Создание экземпляра класса, число
элементов "<<count<<endl;};
Tsquare::~~Tsquare(){count--; cout<<"Деструктор Tsquare: удаление
экземпляра класса, число элементов "<<count<<endl;};
void Tsquare::set_a(float s) {a=s;};
float Tsquare::get_a() {return a;};
float Tsquare::area() {return a*a;};
main()
{
{Tsquare y, g; y.set_a(5); g.set_a(7); cout<<"y="<<y.get_a()<<"
"<<y.area()<<endl;
cout<<"g="<<g.get_a()<<" "<<g.area()<<endl; };
Tsquare *t; t= new Tsquare; t->set_a(2); cout<<"*t="<<t->get_a()<<" "<<t-
>area()<<endl; delete t;
system("pause");
}

```

```

Конструктор Tsquare: Создание экземпляра класса, число элементов 1
Конструктор Tsquare: Создание экземпляра класса, число элементов 2
y=5 25
g=7 49
Деструктор Tsquare: удаление экземпляра класса, число элементов 1
Деструктор Tsquare: удаление экземпляра класса, число элементов 0
Конструктор Tsquare: Создание экземпляра класса, число элементов 1
*t=2 4
Деструктор Tsquare: удаление экземпляра класса, число элементов 0
Для продолжения нажмите любую клавишу . . .

```

Рисунок 2.3 - Вид экрана

2.2 Индивидуальные задания

Общая часть.

Для своего варианта (см. ниже) выполнить следующее:

- 1) Определить пользовательский класс в соответствии с вариантом задания.
- 2) Определить статический элемент – счетчик.
- 3) Определить в классе следующие конструкторы: без параметров, с параметрами. Каждый конструктор должен выводить сообщение о количестве объектов.
- 4) Определить в классе деструктор. Каждый деструктор должен выводить сообщение о количестве объектов.
- 5) Определить в классе внешние компоненты-функции для получения и установки полей данных.
- 6) Написать демонстрационную программу, в которой объекты пользовательского класса создаются с помощью неявного использования конструкторов без параметра.
- 7) Показать в программе явное использование конструкторов с параметрами.
- 8) Показать в программе неявное использование деструкторов.
- 9) Показать в программе использование операторов new и delete.
- 10) Показать в программе использование указателя на объект и указателя на функцию.

Варианты задания.

А. Предметная область – высшее учебное заведение.

Вариант 1. Студент (фамилия, курс, группа, средний балл)

Вариант 2. Группа (название, количество студентов, кафедра, курс).

Вариант 3. Кафедра (название, заведующий, количество студентов, количество преподавателей, количество дисциплин).

Вариант 4. Преподаватель (фамилия, кафедра, должность, стаж, количество дисциплин).

Примечание: должность = {заведующий, профессор, доцент, ..., ассистент}.

Вариант 5. Дисциплина (название, кафедра, количество преподавателей, количество кредитов).

Б. Предметная область – научно-исследовательский институт.

Вариант 6. Сотрудник (фамилия, должность, ученая степень, оклад, лаборатория, количество проектов).

Вариант 7. Лаборатория (название, заведующий, количество сотрудников, количество проектов).

Вариант 8. Проект (название, заказчик, лаборатория, количество сотрудников, объем финансирования).

С. Предметная область – Торговля.

Вариант 9. Покупка (название покупателя, название товара, количество товара, стоимость единицы товара).

Вариант 10. Магазин (название, адрес, фамилия директора, выручка за месяц).

Вариант 000. Товар (название, стоимость единицы товара).

Примечание: идентификаторы переменных и функций должны соответствовать заданию.

Пример выполнения индивидуального задания для варианта 000.

Текст программы

```
#define lname 10
#include<iostream>
using namespace std;
//Описание классов
class Ttovar
{
private:
static int count;
char name[lname];
float price;
public:
Ttovar();
Ttovar(float);
~Ttovar();
void set_name(const char*);
char* get_name();
void set_price(float);
float get_price();
void show();
};
```

```

int Ttovar::count = 0; //Инициализация статических элементов классов
// Реализация методов
Ttovar::Ttovar(){count++;cout<<"Конструктор Ttovar: Создание
экземпляра класса, число элементов "<<count<<endl;};
Ttovar::~Ttovar(){count--; cout<<"Деструктор Ttovar: удаление
экземпляра класса, число элементов "<<count<<endl;};
Ttovar::Ttovar(float r) {count++;cout<<"Конструктор Ttovar: Создание
экземпляра класса, число элементов "<<count<<endl; price=r;};
void Ttovar::set_name(const char*r) {if (strlen(r)<= lname) strcpy(name,
r);};
char *Ttovar::get_name() {return &name[0];};
void Ttovar::set_price(float r) {price=r;};
float Ttovar::get_price() {return price;};
void Ttovar::show(){cout<<"Экземпляр класса Ttovar номер "<<count<<":
name="<<name<<" price="<<price<<endl;};
// Головная программа
main()
{
{
{
Ttovar x;
x.set_name("сахар"); x.set_price(120); x.show();
{Ttovar z1, z2, z3, z4, z5;};
Ttovar d(400); cout<<d.get_price()<<endl;
};
system("pause");
}
}

```

```

Конструктор Ttovar: Создание экземпляра класса, число элементов 1
Экземпляр класса Ttovar номер 1: name=сахар price=120
Конструктор Ttovar: Создание экземпляра класса, число элементов 2
Конструктор Ttovar: Создание экземпляра класса, число элементов 3
Конструктор Ttovar: Создание экземпляра класса, число элементов 4
Конструктор Ttovar: Создание экземпляра класса, число элементов 5
Конструктор Ttovar: Создание экземпляра класса, число элементов 6
Деструктор Ttovar: удаление экземпляра класса, число элементов 5
Деструктор Ttovar: удаление экземпляра класса, число элементов 4
Деструктор Ttovar: удаление экземпляра класса, число элементов 3
Деструктор Ttovar: удаление экземпляра класса, число элементов 2
Деструктор Ttovar: удаление экземпляра класса, число элементов 1
Конструктор Ttovar: Создание экземпляра класса, число элементов 2
400
Деструктор Ttovar: удаление экземпляра класса, число элементов 1
Деструктор Ttovar: удаление экземпляра класса, число элементов 0
Для продолжения нажмите любую клавишу . . .

```

Рисунок 2.4 - Вид экрана

2.3 Контрольные вопросы

- 1) Понятие «класс».
- 2) Понятие «объект».
- 3) Понятие «экземпляр класса».
- 4) Инкапсуляция.
- 5) Конструктор.
- 6) Деструктор.
- 7) Статический элемент класса.
- 8) Модификатор доступа.
- 9) Члены класса.
- 10) Открытые члены класса.
- 11) Закрытые члены класса.
- 12) Члены-поля.
- 13) Члены функции.
- 14) Функции доступа.

3 Лабораторная работа №3. Классы. Наследование

Цель работы: ознакомиться с принципом наследования.

Дополнительные цели:

- 1) Получить начальные сведения о языке UML.
- 2) Научиться строить диаграммы классов на основе кода в Visual Studio.

3.1 Практическая часть

Задание 1.

С помощью механизма наследования, на основе класса «Квадрат», создать классы «Прямоугольник», «Равносторонний треугольник». Из класса «Равносторонний треугольник» создать класс «Равнобедренный треугольник». В каждом из указанных классов Реализовать внешний компонент «Вычисление площади». Написать демонстрационную программу, иллюстрирующую работу с объектами созданных классов.

```
Код программы
#include<iostream>
#include<cmath>
using namespace std;
```

```
// класс Квадрат - описание
class Tsquare
```

```

{
float a;
public:
Tsquare(float);
void set_a(float);
float get_a();
float area();
};
// класс Квадрат - реализация
Tsquare::Tsquare(float a1) {a=a1;};
void Tsquare::set_a(float a1) {a=a1;};
float Tsquare::get_a() {return a;};
float Tsquare::area() {return a*a;};

// класс Прямоугольник - описание
class Trectangle: public Tsquare
{
float b;
public:
Trectangle(float, float);
float area();
};
// класс Прямоугольник - реализация
Trectangle::Trectangle(float a1, float b1): Tsquare(a1) {b=b1;};
float Trectangle::area(){return get_a()*b;};
// класс Равносторонний треугольник - описание
class TequilateralTriag: public Tsquare
{
public:
TequilateralTriag(float);
float area();
};
// класс Равносторонний треугольник - реализация
TequilateralTriag::TequilateralTriag(float a1):Tsquare(a1){};
float TequilateralTriag::area(){return sqrt(3.)/4.*get_a()*get_a();};
// класс Равнобедренный треугольник - описание
class TisoscelesTriag: public Tsquare
{
float angle_degree;
public:
TisoscelesTriag(float, float);
float area();
};

```

```

};
// класс Равнобедренный треугольник - реализация
TisoscelesTriag::TisoscelesTriag(float a1, float
angle_degree1):Tsquare(a1){angle_degree=angle_degree1;};
float TisoscelesTriag::area(){float angle_rad= angle_degree*3.14/180.; return
get_a()*get_a()*sin(angle_rad)*0.5;};

main()

{
Tsquare d(3); cout<<d.area()<<endl;
Trectangle r(2,4); cout<<r.area()<<endl;
TequilateralTriag a(4); cout<<a.area()<<endl;
TisoscelesTriag u(5, 90.); cout<<u.area()<<endl;
system("pause");
}

```

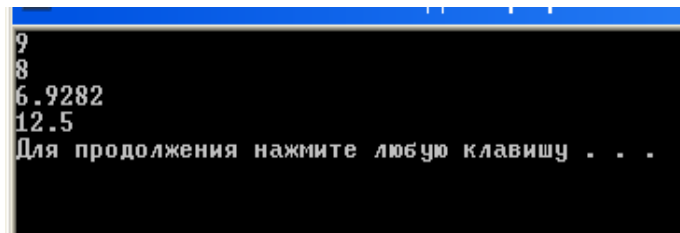


Рисунок 3.1 - Вид экрана

Задание 2.

С помощью интегрированной системы разработки Visual Studio создать диаграмму классов.

- 1) Запустите Visual Studio.
- 2) В главном меню выберите команду File->New->Project from Existing Code. На экране отобразится окно.

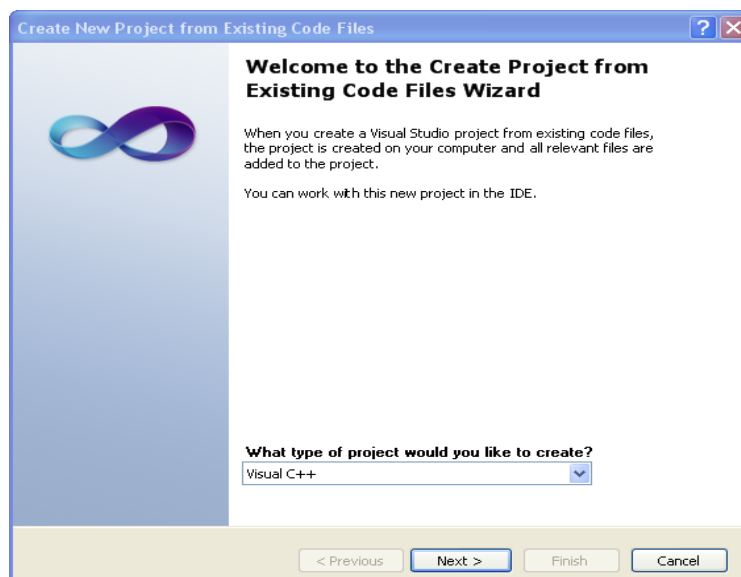


Рисунок 3.2 - Создание диаграммы классов – шаг 1

3) Нажмите кнопку Next. Укажите расположение файла с исходным кодом и название проекта

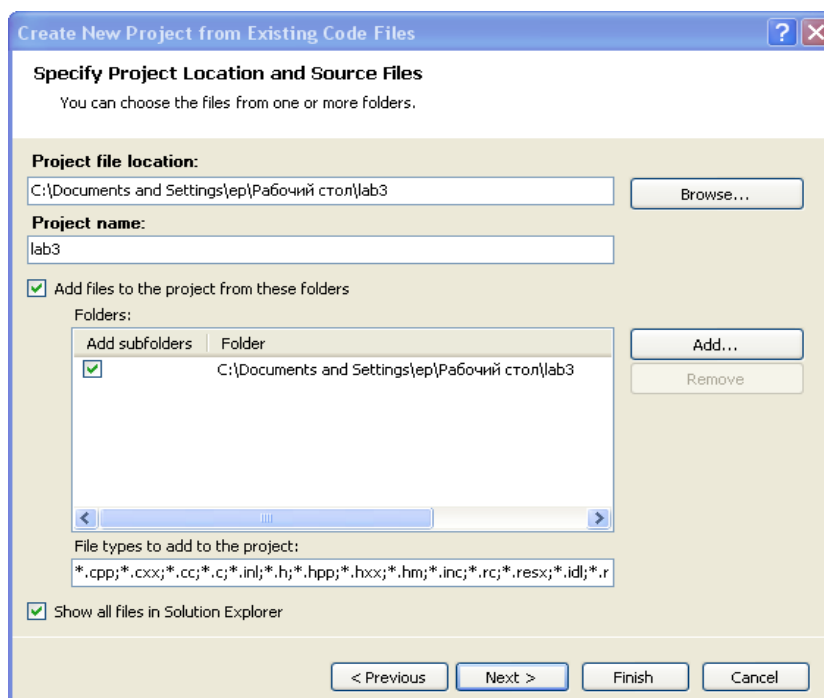


Рисунок 3.3 - Создание диаграммы классов – шаг 2

4) Нажмите кнопку Next (три раза)

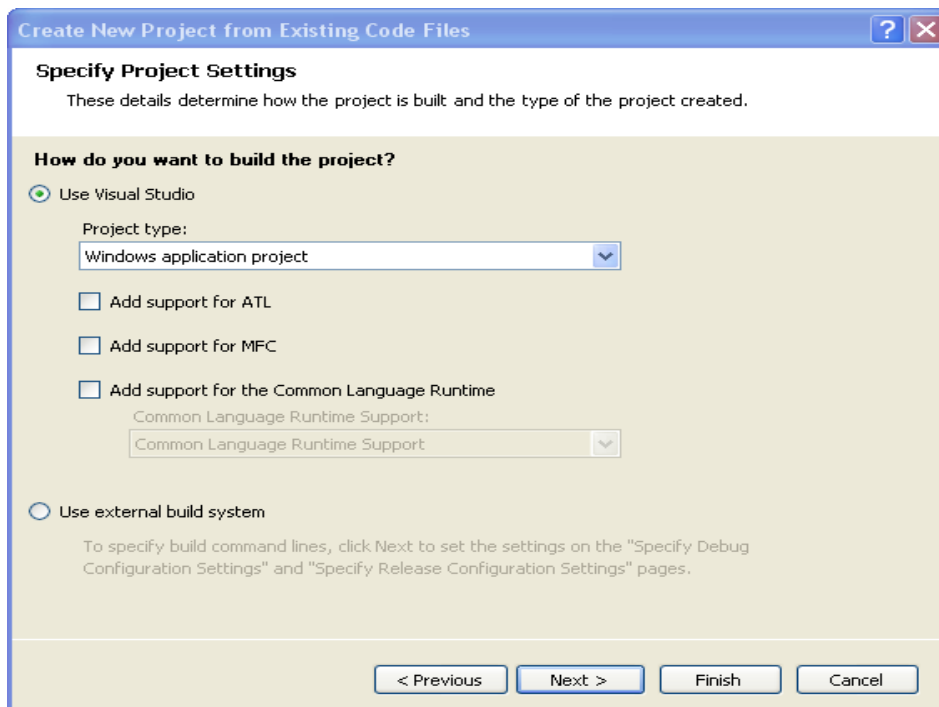


Рисунок 3.4 - Создание диаграммы классов – шаг 3

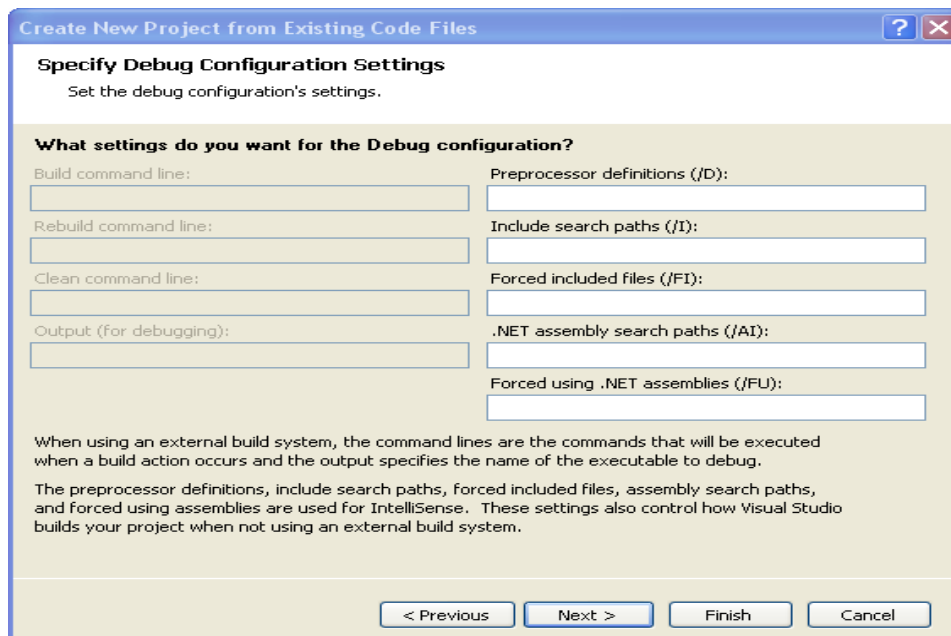


Рисунок 3.5 - Создание диаграммы классов – шаг 4

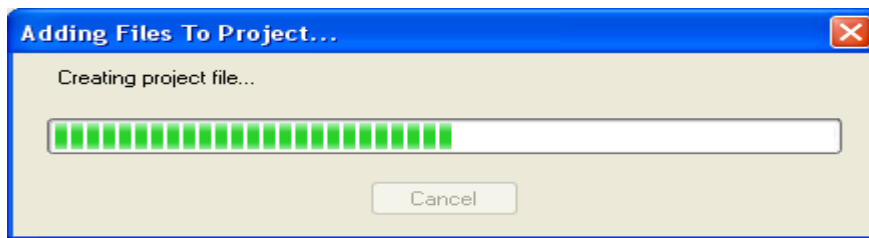


Рисунок 3.6 - Создание диаграммы классов – шаг 5

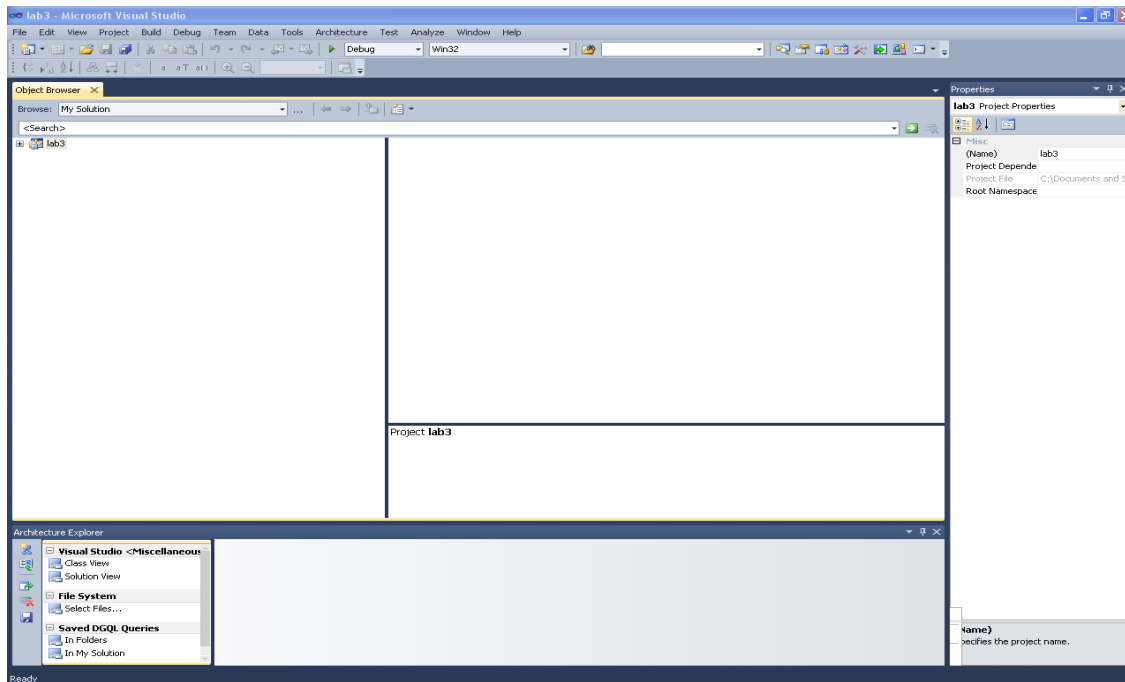


Рисунок 3.7 - Создание диаграммы классов – шаг 6

5) В контекстном меню созданного проекта, выберите ViewClassDiagram, при этом Visual Studio сгенерирует схему классов для существующего проекта:

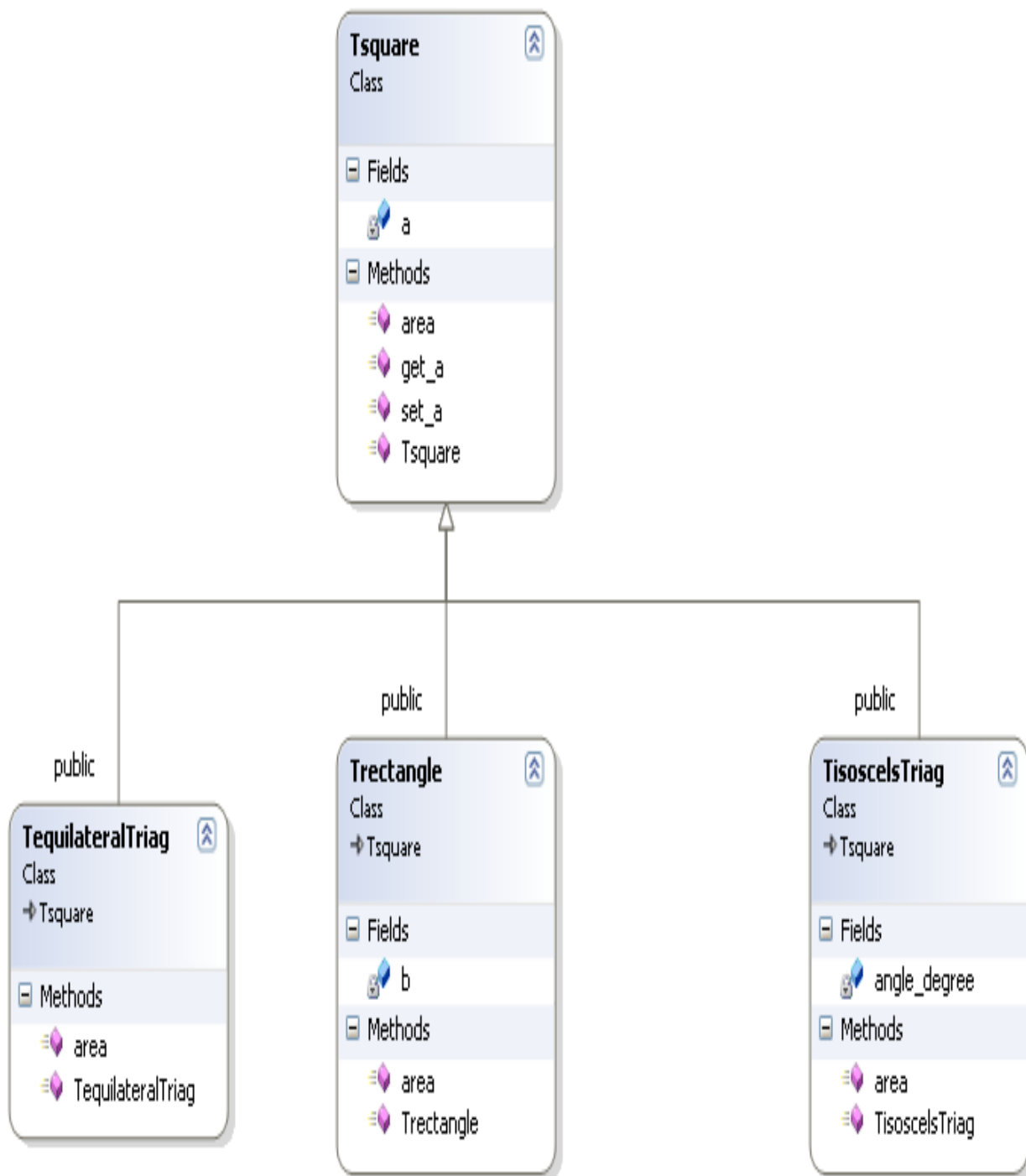


Рисунок 3.8 - Диаграмма классов

3.2 Индивидуальные задания

Общая часть.

Для своего варианта выполнить следующее:

- 1) Создать иерархию классов в соответствии со своим вариантом(см. ниже).

2) Для каждого класса создать конструктор и деструктор, выдающий сообщение о своей работе.

3) Для каждого класса создать внешние функции установки и получения полей данных.

4) Для каждого класса разработать функции, позволяющие представить на экране значения полей данных.

5) Для каждого класса разработать функции, позволяющие вводить с консоли значения полей данных.

6) Написать демонстрационную программу, иллюстрирующую поочередный вызов конструкторов и деструкторов базового и производного классов.

Варианты заданий.

Вариант 1.

Уровень 1. Базовый класс: Персона (Фамилия, год рождения) - производный класс: Студент (курс, группа, средний балл).

Уровень 2. Базовый класс: Студент - производный класс: Дипломник (тема, руководитель).

Вариант 2.

Уровень 1. Базовый класс: Персона (Фамилия) - производный класс: Преподаватель (кафедра, стаж, количество дисциплин).

Уровень 2. Базовый класс: Преподаватель - производный класс: Руководитель дипломной работы (число дипломников, специальность).

Вариант 3.

Уровень 1. Базовый класс: Персона (Фамилия, Имя) - производный класс: Сотрудник (отдел, должность, оклад).

Уровень 2. Базовый класс: Сотрудник - производный класс: Родитель (число детей, семейное положение, материальное положение).

Вариант 4.

Уровень 1. Базовый класс: Персона (Фамилия, Возраст) - производный класс: Школьник (номер школы, класс).

Уровень 2. Базовый класс: Школьник - производный класс: Спортсмен (вид спорта, спортивный разряд).

Вариант 5.

Уровень 1. Базовый класс: Персона (Фамилия, возраст) - производный класс: Военнообязанный (звание, военно-учетная специальность).

Уровень 2. Базовый класс: Военнообязанный - производный класс: Военнослужащий (номер части, фамилия командира).

Вариант 6.

Уровень 1. Базовый класс: Персона (Фамилия, год рождения) - производный класс: Налогоплательщик (РНН, регион).

Уровень 2. Базовый класс: Налогоплательщик - производный класс: Частный Предприниматель (название ЧП, годовой доход).

Вариант 7.

Уровень 1. Базовый класс: Персона (Фамилия, год рождения) - производный класс: Спортсмен (вид спорта, спортивное звание).

Уровень 2. Базовый класс: Спортсмен - производный класс: Победитель (название соревнования, год, месяц).

Вариант 8.

Уровень 1. Базовый класс: Персона (Фамилия) - производный класс: Учитель (стаж, квалификация, номер школы).

Уровень 2. Базовый класс: Учитель - производный класс: Предметник (название предмета, год получения диплома).

Вариант 9.

Уровень 1. Базовый класс: Персона (Фамилия, год рождения) - производный класс: Врач (стаж, специальность, название института).

Уровень 2. Базовый класс: Врач - производный класс: Сотрудник (название клиники, оклад).

Вариант 10.

Уровень 1. Базовый класс: Персона (Фамилия, год рождения) - производный класс: Инженер (стаж, специальность, название института).

Уровень 2. Базовый класс: Инженер - производный класс: Сотрудник (название КБ, оклад).

Вариант 000.

Уровень 1. Базовый класс: Печатное издание (Название, количество страниц, год издания) - производный класс: Книга (автор, издательство).

Уровень 2. Базовый класс: Книга - производный класс: Учебник (предмет, класс).

Пояснение: под разновидностями печатных изданий будем понимать например следующие: книга, журнал, газета, рекламный буклет, статья, ...

Пример выполнения индивидуального задания для варианта 000.

Текст программы

/ лабораторная работа №3, вариант 000
Вариант 000.*

Уровень 1. Базовый класс: Печатное издание (Название, количество страниц, год издания)

- производный класс: Книга (автор, издательство)

Уровень 2. Базовый класс: Книга - производный класс: Учебник (предмет, класс)

```
*/
#define lname 50
#include<iostream>
using namespace std;
//Описание класса Печатное издание
class Publication
{   private: char name[lname]; int num_pages; int year;
protected: static int count;
public: Publication(); ~Publication();
void set_name(const char*); char* get_name();
void set_num_pages(int); int get_num_pages(); void set_year(int);
int get_year(); void show(); void read();
};
int Publication::count = 0; //Инициализация статических элементов
классов
// Реализация методов класса Печатное издание
Publication::Publication()          {count++;          cout<<"Publication
="<<count<<endl;};

Publication::~Publication(){count--;          cout<<"~Publication:
"<<count<<endl;};
void Publication::set_name(const char*r) {if (strlen(r) <= lname)
strcpy(name, r);};
char *Publication::get_name() {return &name[0];};
void Publication::set_num_pages(int r) {num_pages=r;};
int Publication::get_num_pages() {return num_pages;};
void Publication::set_year(int r) {year=r;};
int Publication::get_year() {return year;};
void Publication::show()
{cout<<"Название="<<name<<" Число страниц="
<<num_pages<<" Год="<<year<<endl;};
void Publication::read()
{cout<<"Введите Название книги"<<endl; gets(name); cout<<"Число
страниц"<<endl; cin>>num_pages;
cout<<"Введите год"<<endl; cin>>year;};
//описание класса Книга
class Book: public Publication
{   private: char author [lname]; char publishing_house[lname]; // Автор,
Издательство
```

```

public: Book();
~Book();
void set_author(const char *); char *get_author();
void set_publishing_house(const char *);
char *get_publishing_house();
void show(); void read();
};
//Реализация методов класса Книга
//Book::Book():Publication(){count++; cout<<"Book = "<<count<<endl;};
Book::Book(){cout<<"Book = "<<count<<endl;};
Book::~~Book(){cout<<"~Book: " <<count<<endl;};
void Book::set_author(const char *r) {if (strlen(r) <= lname) strcpy(author,
r);};
char *Book::get_author() {return &author[0];};
void Book::set_publishing_house(const char *r)
{int i; for (i=0; i<=lname-1; i++) publishing_house[i]=r[i];};
char *Book::get_publishing_house() {return &publishing_house[0];};
void Book::read()
{Publication::read(); cout<<"Введите автора"<<endl;
//cin.ignore(); // очистка стандартного входного потока (C++)
rewind(stdin); // сброс стандартного входного потока stdin (C)
gets(author);
cout<<"Введите издательство"<<endl; gets(publishing_house);
//rewind(stdin);
};
void Book::show()
{Publication::show();
cout<<" автор="<<author<<" Издание="<<publishing_house<<endl;
};

//описание класса Учебник
class ClassBook: public Book
{   private: char subject [lname]; int nclass; // учебник, класс
public: ClassBook();
~ClassBook();
void set_subject(const char *); char *get_subject();
void set_nclass(const int); int get_nclass();
void show(); void read();
};

//Реализация методов класса Учебник

```

```

ClassBook::ClassBook(){cout<<"ClassBook = "<<count<<endl;};
ClassBook::~ClassBook(){cout<<"~ClassBook: "<<count<<endl;};
void ClassBook::set_subject(const char *r)
{
if(strlen(r) <= lname) strcpy(subject, r);
}; //предмет, класс
char *ClassBook::get_subject() {return &subject[0];};
void ClassBook::set_nclass(const int r) {nclass=r;};
int ClassBook::get_nclass() {return nclass;};
void ClassBook::show()
{
Book::show(); cout<<" предмет="<<subject<<" Курс="<<nclass<<endl;
};
void ClassBook::read()
{Book::read();
cout<<"Введите дисциплину"<<endl;
gets(subject);
cout<<"Введите курс"<<endl; cin>>nclass;
};

// Головная программа
int main()
{
{ Publication x; x.set_name("Методические рекомендации");
x.set_num_pages(235); x.set_year(1993); //x.show();

Book y; y.set_name("Война и мир"); y.set_num_pages(12000000);
y.set_year(1993);
y.set_author("Толстой"); y.set_publishing_house("Монитор"); //y.show();

ClassBook z; z.set_name("Алгебра"); z.set_num_pages(120);
z.set_year(2005);
z.set_author("Иманбаев"); z.set_publishing_house("Мектеп");
z.set_subject("Математика");
z.set_nclass(8); //z.show();

ClassBook w; w.read(); w.show();
};

cout<<"вызов деструкторов"<<endl;
system("pause");
return 0;
}

```

```
Publication =1
Publication =2
Book =2
Publication =3
Book =3
ClassBook =3
Publication =4
Book =4
ClassBook =4
Введите Название книги
Сборник задач по физике
Число страниц
498
Введите год
1998
Введите автора
Чертов Воробьев
Введите издательство
Высшая школа
Введите дисциплину
Физика
Введите курс
1
Название=Сборник задач по физике Число страниц=498 Год=1998
автор=Чертов Воробьев Издание=Высшая школа
предмет=Физика Курс=1
^ClassBook: 4
^Book: 4
^Publication: 3
^ClassBook: 3
^Book: 3
^Publication: 2
^Book: 2
^Publication: 1
^Publication: 0
вызов деструкторов
Для продолжения нажмите любую клавишу . . .
```

Рисунок 3.9 - Вид экрана

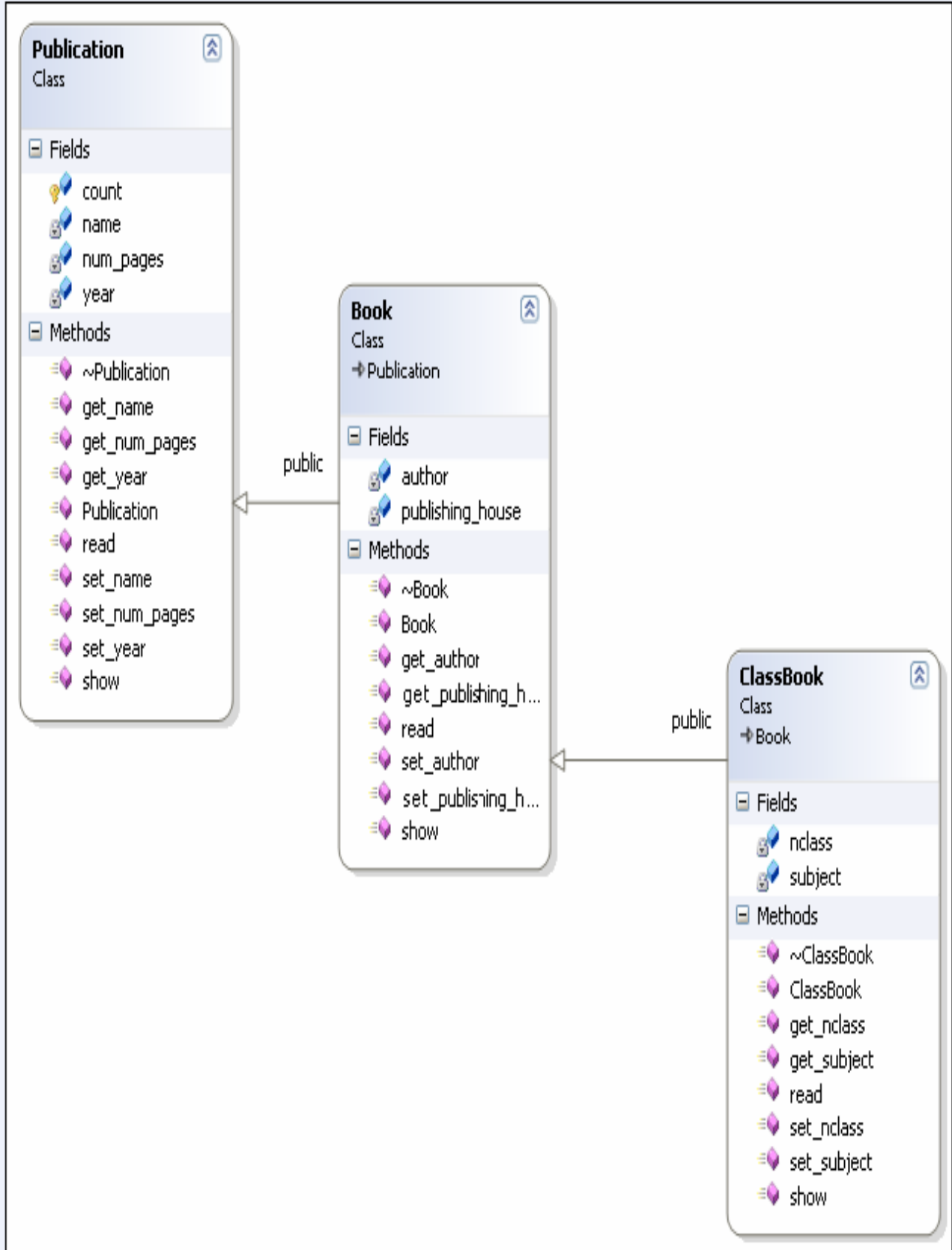


Рисунок 3.10 - Диаграмма классов

3.3 Контрольные вопросы

- 1) Наследование.
- 2) Базовый класс.
- 3) Производный класс.
- 4) Конструкторы базового класса.
- 5) Конструкторы производного класса.
- 6) Организация доступа к членам класса при наследовании.
- 7) Диаграммы UML.
- 8) Диаграмма классов.
- 9) Реализация функций.
- 10) Описание функций.
- 11) Передача параметров по значению.
- 12) Передача параметров по ссылке.

Список литературы

- 1) Шилдт Г. Полный справочник по C++. - М.: Издательский дом «Вильямс», 2006. - 800 с.
- 2) Выгодский М.Я. Справочник по высшей математике. - М.: АСТ Астрель, 2006. - 991 с.
- 3) Выгодский М.Я. Справочник по элементарной математике. - М.: АСТ Астрель, 2006. - 509 с.

Содержание

1 Лабораторная работа №1. Организация консольного ввода/вывода в C++	3
2 Лабораторная работа №2. Классы. Наследование	10
3 Лабораторная работа №3. Классы. Инкапсуляция	16
Список литературы	31

Сейткерим Бимурзаевич Бимурзаев
Ерик Сулейменович Турганбаев

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

Методические указания по выполнению лабораторных работ
для студентов специальности
5В070400 – Вычислительная техника и программное обеспечение

Редактор
Специалист по стандартизации

Подписано в печать __. __. ____.
Тираж 100 экз
Объем __2,0 уч.-изд. л.

Формат 60x84 1/16
Бумага типографская №1
Заказ ____. Цена 1000 т.

Копировально-множительное бюро
некоммерческого акционерного общества
«Алматинский университет энергетики и связи»
050013, Алматы, Байтурсынова, 126