

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ  
НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ УКРАИНЫ  
«КИЕВСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ»

**НАСЛЕДОВАНИЕ C++**  
**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

**к лабораторной работе № 2**

**по дисциплине «ООП»**

Киев 2016

## СОДЕРЖАНИЕ

1	Цель лабораторной работы.....	3
2	Теоретические положения .....	4
3	Задания.....	8
4	Требования к отчету .....	31
5	Контрольные вопросы.....	32



## **1 ЦЕЛЬ ЛАБОРАТОРНОЙ РАБОТЫ**

Цель работы – изучить основные концепции объектно-ориентированного программирования. Изучить особенности наследования и множественного наследования.



## 2 ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

Наследование – это способность брать существующий (базовый) класс и порождать из него новый класс – потомок, с наследованием всех его атрибутов и поведения.

Синтаксис определения производного класса:

```
class имя_класса: спецификатор_доступа1 базовый_класс1, спецификатор_доступа2
базовый_класс2, {
    //тело класса
};
```

При описании нового класса, производного от какого-то одного или нескольких базовых классов, можно добавлять новые функции-элементы и элементы-данные, сохраняя при этом все элементы родителей, а можно родительские элементы переопределить или перегрузить. В производном классе доступны открытые и защищённые элементы базового класса (прямого или косвенного). Закрытые элементы базового класса в производном классе недоступны.

Производный класс может наследоваться от базового класса как public, protected или private. Защищённое и закрытое наследование встречаются редко и каждое из них нужно использовать с большой осторожностью. Доступ к элементам базового класса из производного класса можно представить следующим образом (таблица 2.1).

Таблица 2.1 – Типы наследования

Спецификатор доступа к элементам в базовом классе	Тип наследования		
	public	protected	private
public	public	protected	private
protected	protected	protected	private
private	-	-	-

При порождении класса как public открытые элементы базового класса становятся открытыми элементами производного класса, а защищённые



элементы базового – защищёнными элементами производного. Закрытые элементы базового класса никогда не доступны производному классу.

При защищённом наследовании открытые и защищённые элементы базового класса становятся защищёнными элементами производного класса. При закрытом наследовании открытые и защищённые элементы базового класса становятся закрытыми элементами производного класса. При закрытом и защищённом наследовании не справедливо отношение, что объект производного класса является объектом базового класса.

Если не указать спецификацию наследования, то наследование будет закрытым.

Если в классе-наследнике переопределить функцию-элемент (внести новую функцию с тем же именем), то для объектов этого класса новая функция отменит родительскую. Если же обращаться к объекту этого класса, то вызываться будет новая функция. Если надо вызывать именно функцию базового класса, то надо использовать операцию разрешения области действия.

### Пример 1

```
#include "stdafx.h"
#include <cstdlib>
#include <iostream>

using namespace std;

class Shape{
public:
    void Draw(){ cout << "Shape - Draw" << endl; };
};
class Circl : public Shape{
public:
    void Draw(){ cout << "Circl - Draw" << endl; };
};
int _tmain(int argc, _TCHAR* argv[])
{
    Shape *PQ1 = new Shape;
    PQ1->Draw();
    Circl *PQ2 = new Circl;
    PQ2->Draw();
    PQ2->Shape::Draw();
    ((Shape *)PQ2)->Draw();
    system("pause");
    return 0;
}
```



Если бы в классе-наследнике `Circl` отсутствовала функция `Draw`, то все приведённые операторы вызывали бы функцию базового класса.

Отдельно выделяют **множественное наследование**, когда один производный класс имеет несколько базовых. Преимущество множественного наследования состоит в том, что в одном производном классе сочетается поведение нескольких базовых. Однако при этом может возникать конфликт имён – несколько базовых классов могут иметь элементы-данные или функции-элементы с одинаковыми именами. **Возникает неоднозначность при использовании таких элементов. Для разрешения конфликта используется оператор разрешения видимости или виртуальное наследование.**

## Пример 2

```
class location{
public:
    int x, y;
    location(int x1, int y1) :x(x1), y(y1){}
};
class circl : public location{
    int rad;
public:
    circl(int x1, int y1, int r1) :location(x1, y1), rad(r1){}
    void show() { cout<<"x-"<<x<<" y-"<<y<<" r-"<<rad<<endl; }
};
class message : public location{
    char *msg;
public:
    message(int x1, int y1, char *text) :location(x1, y1), msg(text){};
    void show() { cout << "x-" << x << " y-" << y << " message-" << msg
<< endl; }
};
class mescir : public circl, public message{
public:
    mescir(int x1, int y1, int r1, char *text) : circl(x1, y1, r1),
message(x1, y1, text){};
    void show();
};
void mescir::show(){
    circl::show();
    message::show();
}
int _tmain(int argc, _TCHAR* argv[])
{
    mescir *obj= new mescir(250, 100, 30, "Hello word");
    obj->show();
    cout << endl;
    obj->circl::show();
    ((circl*)obj)->show();
    system("pause");
    return 0;
}
```



}



### 3 ЗАДАНИЯ

Разработать пошаговый алгоритм решения задачи. Разработать UML диаграмму классов. Выполнить программную реализацию задания согласно варианту. Прототипы классов должны содержаться в отдельном \*.h-файле. В программе обязательно предусмотреть вывод информации об исполнителе работы (ФИО), номере варианта, выбранном уровне сложности и задании. Предусмотреть возможность повторного запуска программы (запрос о желании выйти из программы, или продолжить работу). Если программная реализация предполагает решение нескольких подзадач, разработать меню для демонстрации каждой из них. Ключевые моменты программы обязательно должны содержать комментарии.

#### Уровень А (1 балл)

1. Спроектировать класс «Отрезок», являющийся наследником класса «Точка», с закрытыми элементами  $x1, y1$ . Определить для класса собственные элементы  $x2, y2$  (вторая точка). Найти длину отрезка заданного координатами  $x1, y1$  и  $x2, y2$ .

2. Спроектировать класс «Квадрат», являющийся наследником класса «Точка», с закрытыми элементами  $x1, y1$ . Определить для класса собственные элементы  $x2, y2$  (вторая точка). Две точки образуют диагональ квадрата. Найти площадь заданного квадрата, по длине диагонали.

3. Спроектировать класс «Квадрат», являющийся наследником класса «Точка», с закрытыми элементами  $x1, y1$ . Определить для класса собственные элементы  $x2, y2$  (вторая точка). Две точки образуют диагональ квадрата. Найти периметр заданного квадрата, по длине диагонали.

4. Спроектировать класс «Круг», являющийся наследником класса «Точка», с закрытыми элементами  $x1, y1$ . Определить для класса собственные элементы  $x2, y2$  (образуют радиус с первой точкой). Найти площадь круга.





5. Спроектировать класс «Окружность», являющийся наследником класса «Точка» с закрытыми элементами  $x1$ ,  $y1$ . Определить для класса собственные элементы  $x2$ ,  $y2$  (образуют радиус с первой точкой). Найти длину окружности.

6. Спроектировать класс «Квадрат», являющийся наследником класса «Точка» с закрытыми элементами  $x1$ ,  $y1$ . Определить для класса собственный элемент  $a$  (сторона квадрата). Ввести в базовый класс чисто виртуальную функцию «Площадь фигуры», переопределить её в дочернем классе. Найти площадь квадрата.

7. Спроектировать класс «Квадрат», являющийся наследником класса «Точка». Определить для класса собственный элемент  $a$  (сторона квадрата). Ввести в базовый класс чисто виртуальную функцию «Периметр фигуры», переопределить её в дочернем классе. Найти периметр квадрата.

8. Спроектировать класс «Конус», являющийся наследником класса «Круг». Определить для класса собственный элемент  $a$  (высота конуса). Ввести в базовый класс чисто виртуальную функцию «Площадь фигуры», переопределить её в дочернем классе. Найти площадь поверхности конуса.

9. Спроектировать класс «Конус», являющийся наследником класса «Круг». Определить для класса собственный элемент  $a$  (высота конуса). Ввести в базовый класс чисто виртуальную функцию «Объём фигуры», переопределить её в дочернем классе. Найти объём конуса.

10. Спроектировать класс «Цилиндр», являющийся наследником класса «Окружность». Определить для класса собственный элемент  $a$  (высота цилиндра). Ввести в базовый класс чисто виртуальную функцию «Площадь фигуры», переопределить её в дочернем классе. Найти площадь поверхности цилиндра.

11. Спроектировать класс «Цилиндр», являющийся наследником класса «Окружность». Определить для класса собственный элемент  $a$  (высота



цилиндра). Ввести в базовый класс чисто виртуальную функцию «Объём фигуры», переопределить её в дочернем классе. Найти объём цилиндра.

12.Спроектировать класс «Параллелепипед», являющийся наследником класса «Прямоугольник». Определить для класса собственный элемент  $h$  (высота параллелепипеда). Ввести в базовый класс чисто виртуальную функцию «Площадь фигуры», переопределить её в дочернем классе. Найти площадь поверхности параллелепипеда.

13.Спроектировать класс «Параллелепипед», являющийся наследником класса «Прямоугольник». Определить для класса собственный элемент  $h$  (высота параллелепипеда). Ввести в базовый класс чисто виртуальную функцию «Объём фигуры», переопределить её в дочернем классе. Найти объём параллелепипеда.

14. Определить класс «Строка» с закрытыми элементами  $s$  (указатель на строку) и  $len$ (длина строки). Разработать на основании класса дочерний класс «Форматированная строка», имеющая свои собственные элементы  $f\_len$  (длина форматированной строки) и массив  $f\_s$  (массив указателей на строки длиной  $f\_len$ ). Задать для базового класса функцию вывода строки на экран, переопределить её в дочернем для вывода форматированного текста.

15.Определить класс «Строка» с закрытыми элементами  $s$  (указатель на строку) и  $len$ (длина строки). Разработать на основании класса дочерний класс «Словарь», который будет содержать все слова строки в алфавитном порядке. Задать для базового класса функцию вывода строки на экран, переопределить её в дочернем для вывода словаря.

16.Спроектировать класс «Сообщение», являющийся наследником класса «Строка». Определить для класса собственные элементы  $x$  и  $y$  (координаты точки вывода сообщения на экране). Определить в базовом классе метод вывода строки на экран, переопределить его в дочернем.



17. Спроектировать класс «Пирамида», являющийся наследником класса «Треугольник». Определить для класса собственный элемент  $h$  (высота пирамиды). Ввести в базовый класс чисто виртуальную функцию «Объём фигуры», переопределить её в дочернем классе. Найти объём пирамиды.

18. Спроектировать класс «Пирамида», являющийся наследником класса «Треугольник». Определить для класса собственный элемент  $h$  (высота пирамиды). Ввести в базовый класс чисто виртуальную функцию «Площадь поверхности фигуры», переопределить её в дочернем классе. Найти площадь поверхности пирамиды.

19. Спроектировать класс «Параллелограмм», являющийся наследником класса «Треугольник». Определить для класса собственный элемент  $h$  (высота параллелограмма), значение элемента вычислять по значениям элементов базового класса ( $a$ ,  $b$  – стороны параллелограмма,  $c$  – диагональ). Ввести в базовый класс чисто виртуальную функцию «Периметр фигуры», переопределить её в дочернем классе. Найти периметр параллелограмма.

20. Спроектировать класс «Пирамида», являющийся наследником класса «Квадрат». Определить для класса собственный элемент  $h$  (высота пирамиды). Ввести в базовый класс чисто виртуальную функцию «Объём фигуры», переопределить её в дочернем классе. Найти объём пирамиды.

21. Спроектировать класс «Пирамида», являющийся наследником класса «Квадрат». Определить для класса собственный элемент  $h$  (высота пирамиды). Ввести в базовый класс чисто виртуальную функцию «Площадь поверхности фигуры», переопределить её в дочернем классе. Найти площадь поверхности пирамиды.

22. Спроектировать класс «Упорядоченный массив», являющийся наследником класса «Одномерный массив чисел». Определить в базовом классе чисто виртуальную функцию сортировки данных по возрастанию, переопределить её в дочернем (сортировку осуществлять методом вставок).



Определить в базовом классе метод, определяющий минимальный элемент, переопределить его в дочернем.

23.Спроектировать класс «Упорядоченный массив», являющийся наследником класса «Одномерный массив чисел». Определить в базовом классе чисто виртуальную функцию сортировки данных по убыванию, переопределить её в дочернем (сортировку осуществлять методом выбора). Определить в базовом классе метод, определяющий максимальный элемент, переопределить его в дочернем.

24.Определить класс «Одномерный массив чисел» с закрытыми элементами mas – массив чисел и n–число элементов массива. Задать для класса метод вывода элементов массива на экран. Спроектировать, дочерний класс, имеющий собственный элемент – разделитель чисел. Переопределить метод вывода элементов с использованием разделителя.

25.Определить класс «Одномерный массив чисел» с закрытыми элементами mas – массив чисел и n – число элементов массива. Определить для класса чисто виртуальную функцию вывода элементов на экран. Разработать два дочерних класса, в которых переопределить функцию вывода на экран: реализовать вывод элементов в строку и в столбец.

26.Определить класс «Одномерный массив чисел» с закрытыми элементами mas – массив чисел и n – число элементов массива. Определить для класса виртуальную функцию вывода элементов на экран. Разработать дочерний класс, в котором задать собственный элемент – строку, содержащий все характеристика массива (число элементов, максимальный и минимальный элемент, среднее арифметическое элементов массива). Переопределить функцию вывода элементов (с учётом вывода характеристик).

27.Спроектировать класс «Всплывающее меню», являющийся наследником класса «Меню». Определить для класса собственные элементы x и



у (координаты вывода меню на экран). Определить в базовом классе метод отображения меню на экране, переопределить его в дочернем классе.

28. Определить класс «Множество» с закрытыми элементами `mas` – массив с элементами множества и `n` – число элементов массива. Определить для класса метод отображения на экране элементов множества. Разработать класс-наследник, в котором переопределить метод таким образом, чтобы вывод элементов множества осуществлялся в верхнем регистре.

29. Определить класс «Множество» с закрытыми элементами `mas` – массив с элементами множества и `n` – число элементов массива. Определить для класса метод отображения на экране элементов множества. Разработать класс-наследник, в котором переопределить метод таким образом, чтобы вывод элементов множества осуществлялся в нижнем регистре.

30. Определить класс «Множество» с закрытыми элементами `mas` – массив с элементами множества и `n` – число элементов массива. Определить для класса метод отображения на экране элементов множества. Разработать класс-наследник, имеющий собственный элемент – разделитель. Переопределить метод отображения таким образом, чтобы вывод элементов множества осуществлялся через разделитель.

### **Уровень Б (+0,5 балла)**

Требуется построить иерархию классов заданную в таблице 3.1, и для каждой фигуры найти (если возможно) периметр, площадь и объём.

В скобках указаны элементы-данные классов. Элементы данные должны быть защищенными.



Таблица 3.1 – Варианты заданий

№ варианта	Базовый класс	Наследник базового класса	Наследник наследника
1	Отрезок (длина)	Прямоугольник (вторая сторона)	Параллелепипед (третья сторона)
2	Отрезок (длина)	Треугольник (две стороны)	Призма (высота)
3	Отрезок (длина)	Треугольник (высота)	Призма (высота)
4	Отрезок (длина)	Равнобедренный треугольник	Призма (высота)
5	Отрезок (длина)	Треугольник (второй катет)	Призма (высота)
6	Отрезок (длина)	Прямоугольник (вторая диагональ и угол между ними)	Призма (высота)
7	Отрезок (длина)	Параллелограмм (вторая сторона и угол между ними)	Призма (высота)
8	Отрезок (длина)	Параллелограмм (вторая диагональ и угол между ними)	Призма (высота)



Продолжение таблицы 3.1

№ варианта	Базовый класс	Наследник базового класса	Наследник наследника
9	Отрезок (длина)	Ромб (вторая диагональ)	Призма (высота)
10	Отрезок (длина)	Треугольник (две стороны)	Пирамида (высота)
11	Отрезок (длина)	Прямоугольник (вторая диагональ и угол между ними)	Пирамида (высота)
12	Отрезок (длина)	Параллелограмм (вторая сторона и угол между ними)	Пирамида (высота)
13	Отрезок (длина)	Параллелограмм (вторая диагональ и угол между ними)	Пирамида (высота)
14	Отрезок (длина)	Ромб (вторая диагональ)	Пирамида (высота)
15	Точка ( $x1, y1$ )	Круг ( $x2, y2$ )	Цилиндр (высота)
16	Точка ( $x1, y1$ )	Круг ( $x2, y2$ )	Конус (высота)
17	Точка ( $x1, y1$ )	Круг ( $x2, y2$ )	Шар
18	Точка ( $x1, y1$ )	Круг ( $x2, y2$ )	Эллипс (второй радиус)
19	Точка ( $x1, y1$ )	Круг ( $x2, y2$ )	Тор ( $R$ )



Продолжение таблицы 3.1

№ варианта	Базовый класс	Наследник базового класса	Наследник наследника
20	Отрезок (длина)	Равносторонний треугольник	Тетраэдр
21	Точка ( $x1, y1$ )	Круг ( $x2, y2$ )	Сектор (длина)
22	Точка ( $x1, y1$ )	Круг ( $x2, y2$ )	Сектор (угол в радианах)
23	Точка ( $x1, y1$ )	Круг ( $x2, y2$ )	Сегмент (угол в радианах)
24	Точка ( $x1, y1$ )	Круг ( $x2, y2$ )	Сегмент (высота)
25	Точка ( $x1, y1$ )	Круг ( $x2, y2$ )	Сегмент (разность между радиусом и высотой)
26	Точка ( $x1, y1$ )	Круг ( $x2, y2$ )	Сегмент (длина хорды)
27	Круг (радиус)	Шар	Шаровой сегмент (высота)
28	Круг (радиус)	Шар	Шаровой слой (две высоты)
29	Круг (радиус)	Эллипс (второй радиус)	Эллипсоид (третий радиус)
30	Круг (радиус)	Эллипс (второй радиус)	Цилиндр (высота)





### Уровень В (+1 балл)

1. Определить базовый класс «Список», и дочерние классы «Стек» и «Очередь». В базовом классе определить методы добавления и удаления элементов в список, а в дочерних – переопределить их соответствующим образом.
2. Определить следующую иерархию классов (рисунок 3.1).

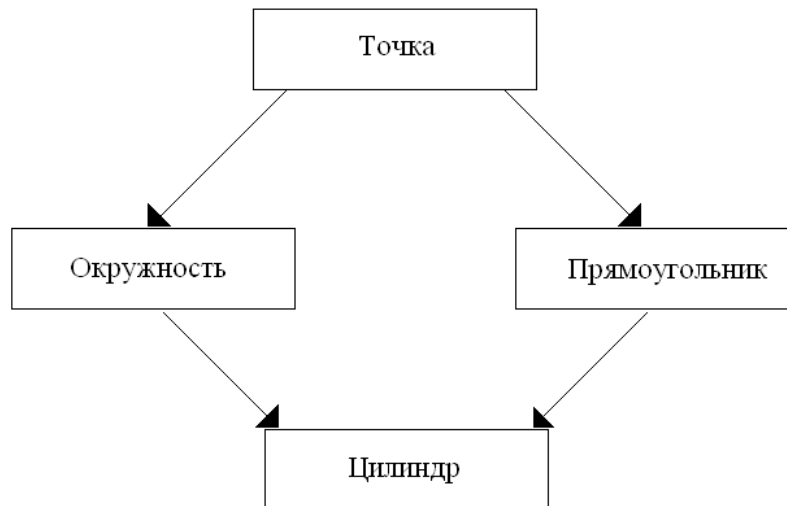


Рисунок 3.1

Определить площадь каждой фигуры (для цилиндра – площадь боковой поверхности и объём). Инициализация данных производится пользователем.

3. Определить следующую иерархию классов (рисунок 3.2).

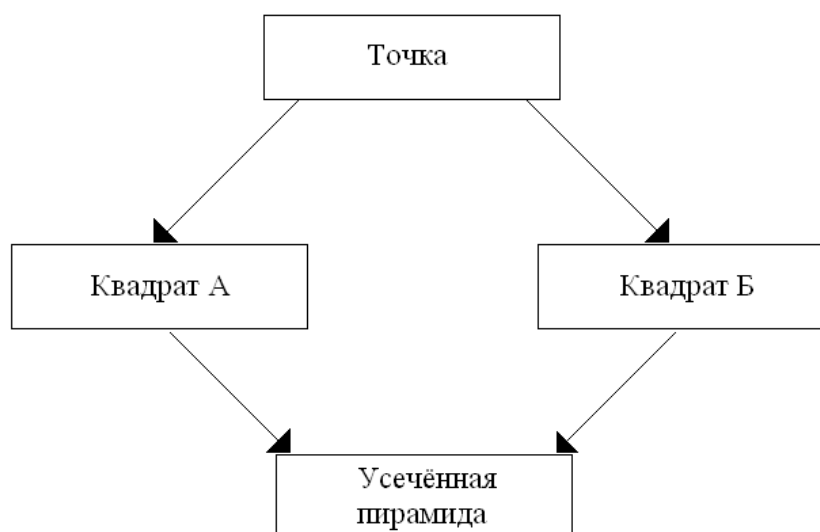


Рисунок 3.2



Определить площадь каждой фигуры (для усечённой пирамиды – площадь боковой поверхности и объём). Инициализация данных производится пользователем.

4. Определить следующую иерархию классов (рисунок 3.3).

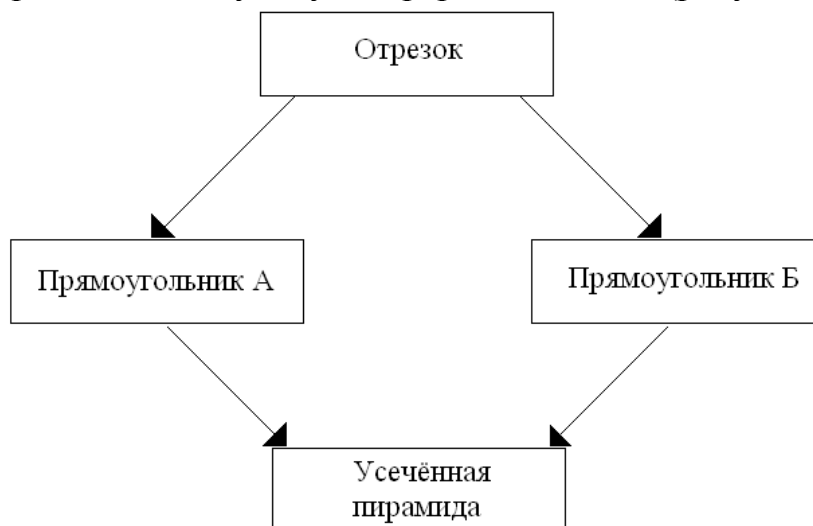


Рисунок 3.3

Определить площадь каждой фигуры (для усечённой пирамиды – площадь боковой поверхности и объём). Инициализация данных производится пользователем.

5. Определить следующую иерархию классов (рисунок 3.4).

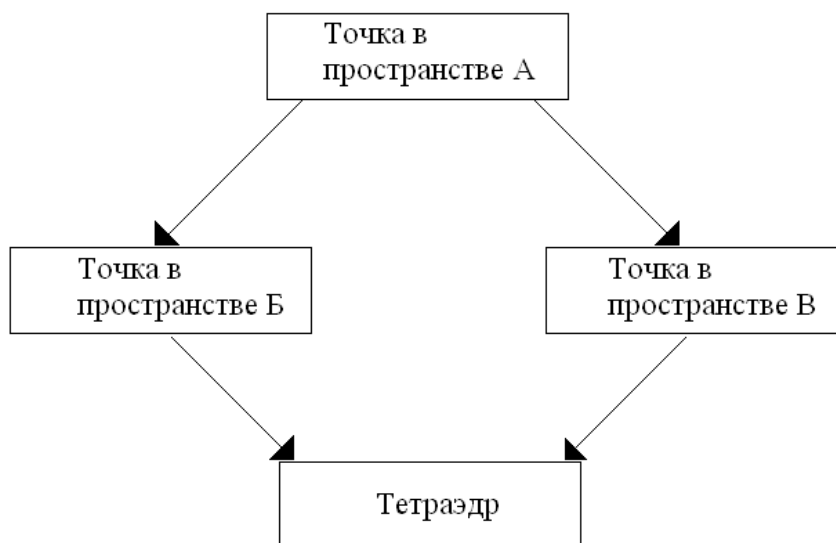


Рисунок 3.4



Определить площадь боковой поверхности и объём тетраэдра.  
Инициализация данных производится пользователем.

6. Определить следующую иерархию классов (рисунок 3.5).

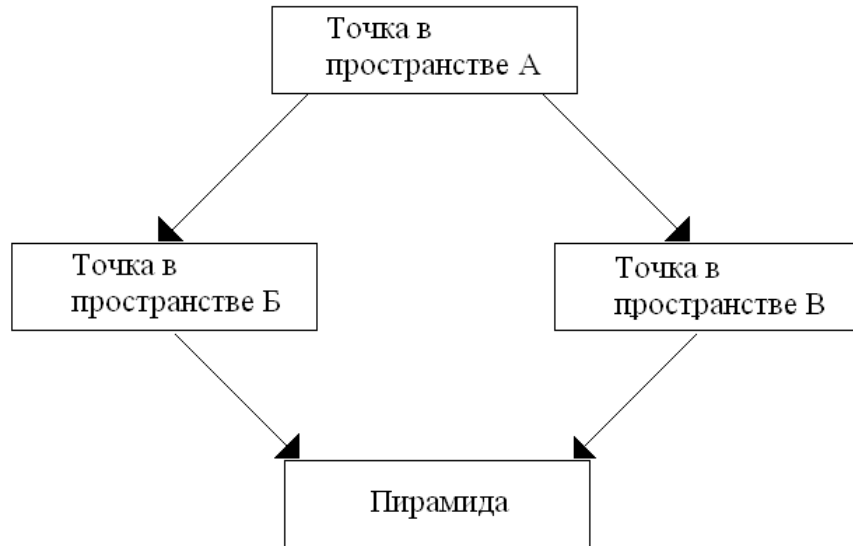


Рисунок 3.5

Найти уравнение плоскости, проходящей через точки А, Б, В и высоту пирамиды.

7. Определить следующую иерархию классов (рисунок 3.6).

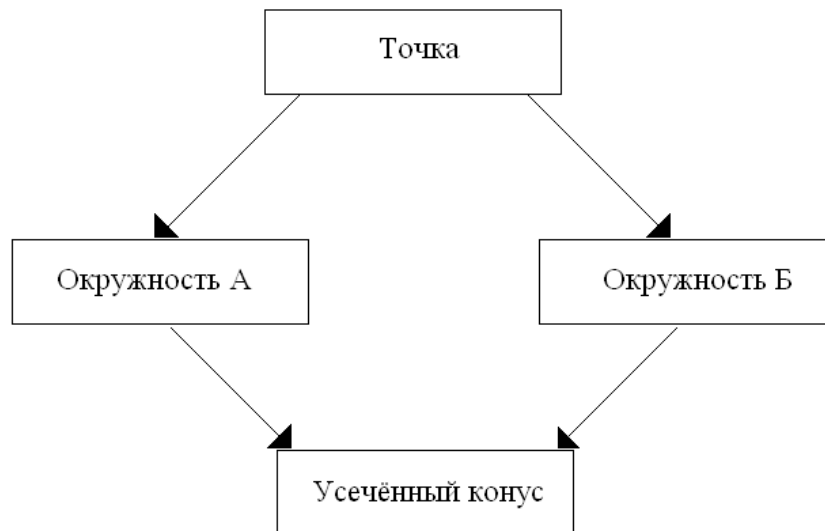


Рисунок 3.6



Определить площадь каждой фигуры (для усечённого конуса – площадь боковой поверхности и объём). Инициализация данных производится пользователем.

8. Определить следующую иерархию классов (рисунок 3.7).

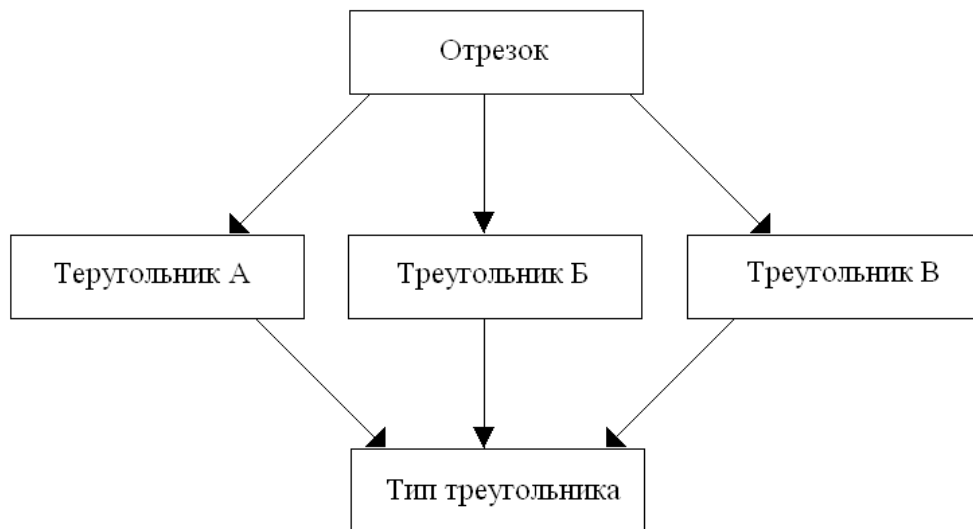


Рисунок 3.7

В классе «Тип треугольника», определить тип каждого треугольника и найти его площадь. Инициализация данных производится пользователем.

9. Определить следующую иерархию классов (рисунок 3.8).

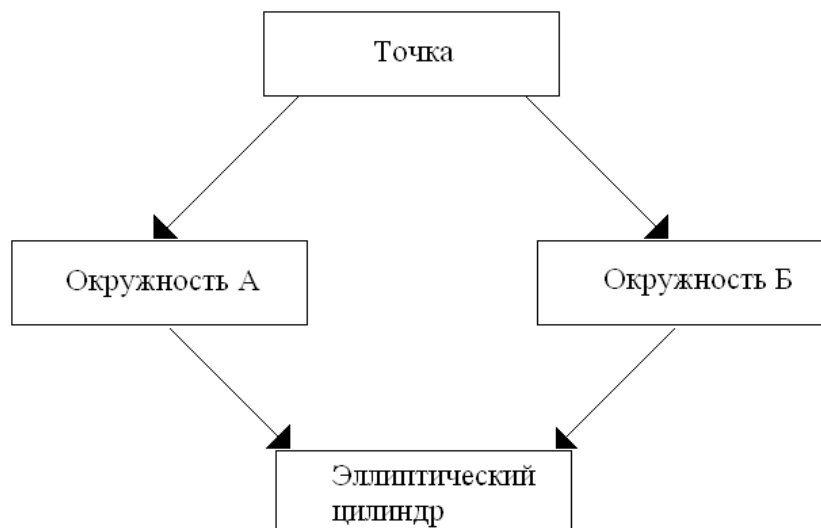


Рисунок 3.8



Определить площадь каждой фигуры (для эллиптического цилиндра – площадь боковой поверхности и объём). Инициализация данных производится пользователем.

10. Определить следующую иерархию классов (рисунок 3.9).

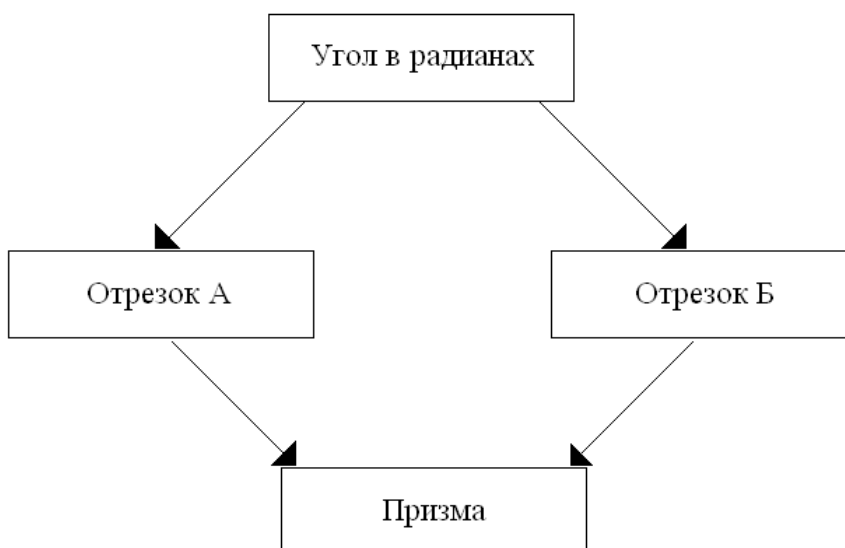


Рисунок 3.9

Определить площадь боковой поверхности и объём призмы. Инициализация данных производится пользователем.

11. Определить следующую иерархию классов (рисунок 3.10).

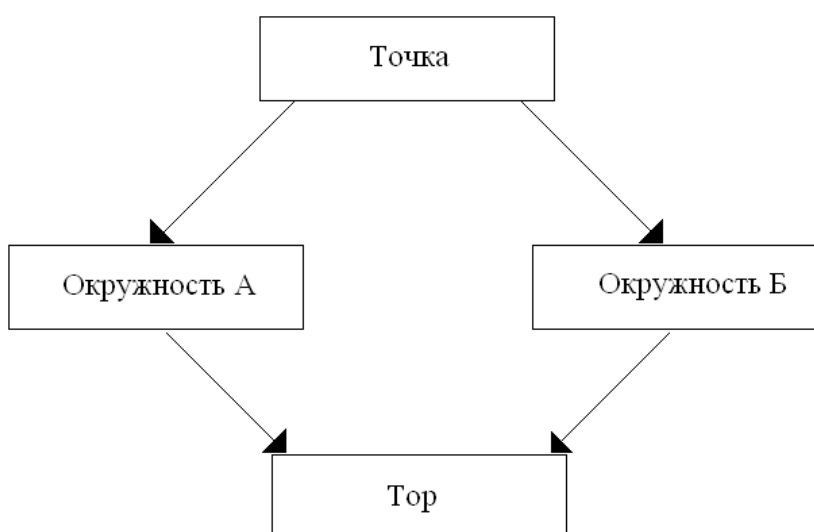


Рисунок 3.10



Определить площадь каждой фигуры (для тора – площадь боковой поверхности и объём). Инициализация данных производится пользователем.

12. Определить следующую иерархию классов (рисунок 3.11).

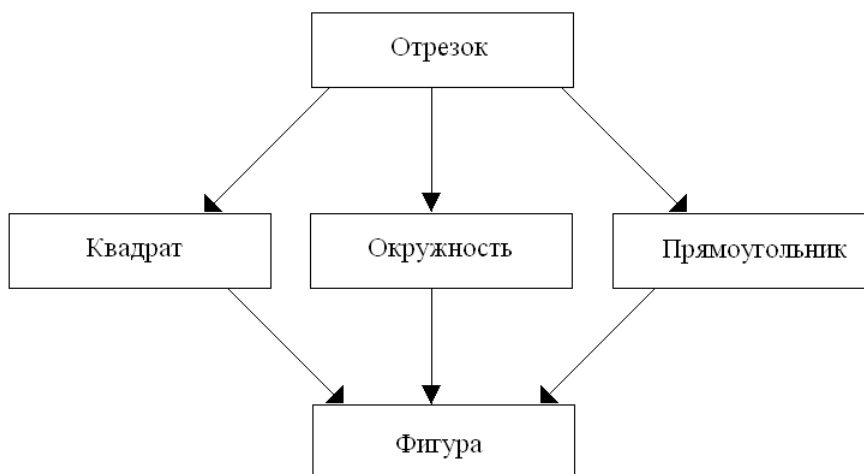


Рисунок 3.11

Определить площадь каждой фигуры. В классе «Фигура» определить можно ли в одну из фигур, вписать две остальные. Инициализация данных производится пользователем.

13. Определить следующую иерархию классов (рисунок 3.12).

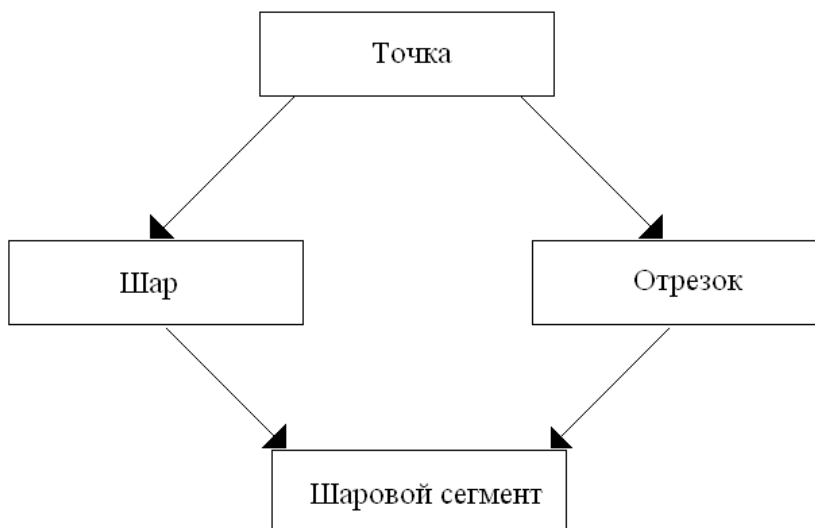


Рисунок 3.12



Определить площадь каждой фигуры (для шара и шарового сегмента – площадь боковой поверхности и объём). Инициализация данных производится пользователем.

14. Определить следующую иерархию классов (рисунок 3.13).

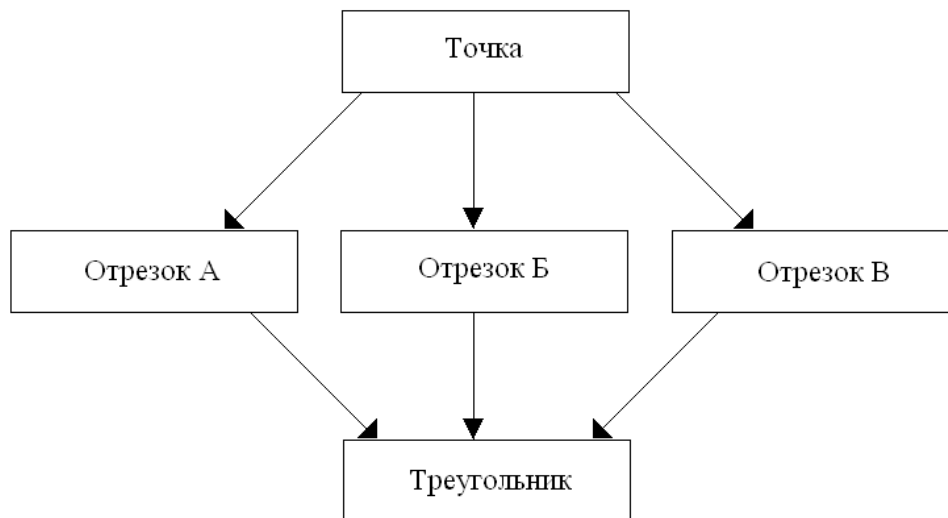


Рисунок 3.13

Определить в классе, образуют ли 3 отрезка треугольник, если треугольник существует найти его площадь. Инициализация данных производится пользователем.

15. Определить следующую иерархию классов (рисунок 3.14).

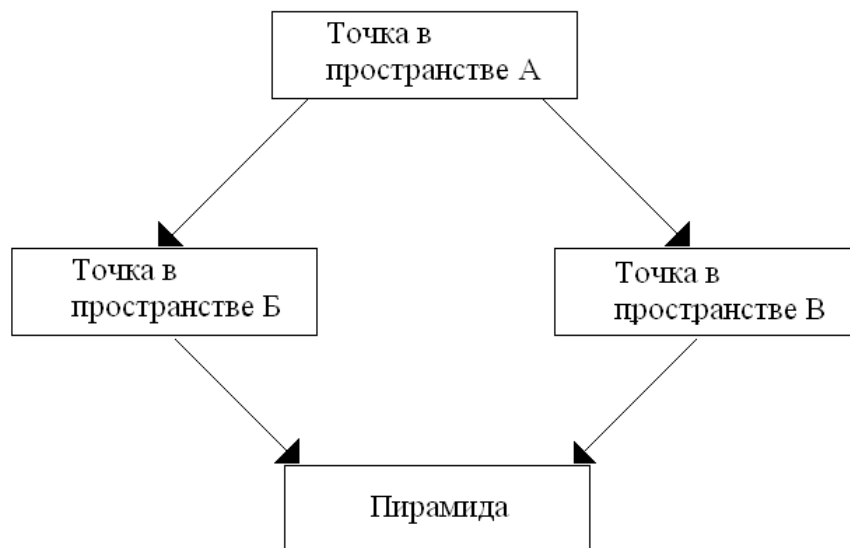


Рисунок 3.14

Определить длину ребер основания и уравнение плоскости основания пирамиды. Инициализация данных производится пользователем.

16. Определить следующую иерархию классов (рисунок 3.15).

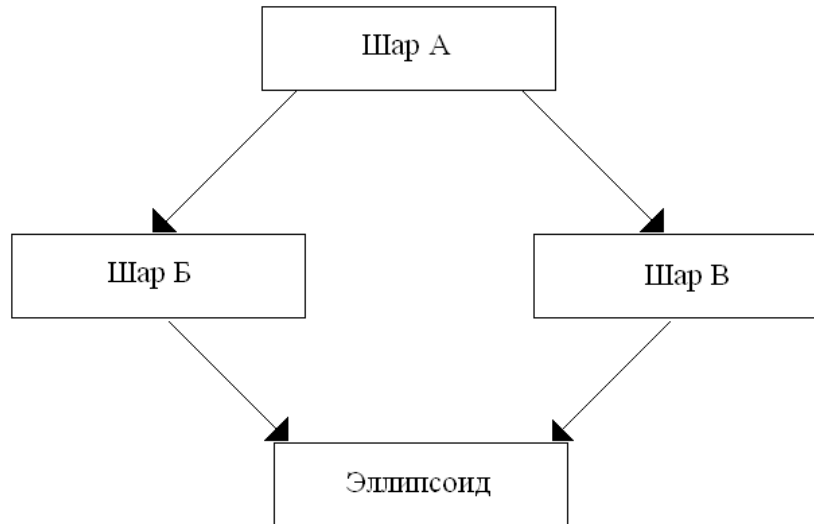


Рисунок 3.15

Определить площадь боковой поверхности и объем каждой фигуры. Инициализация данных производится пользователем.

17. Определить следующую иерархию классов (рисунок 3.16).

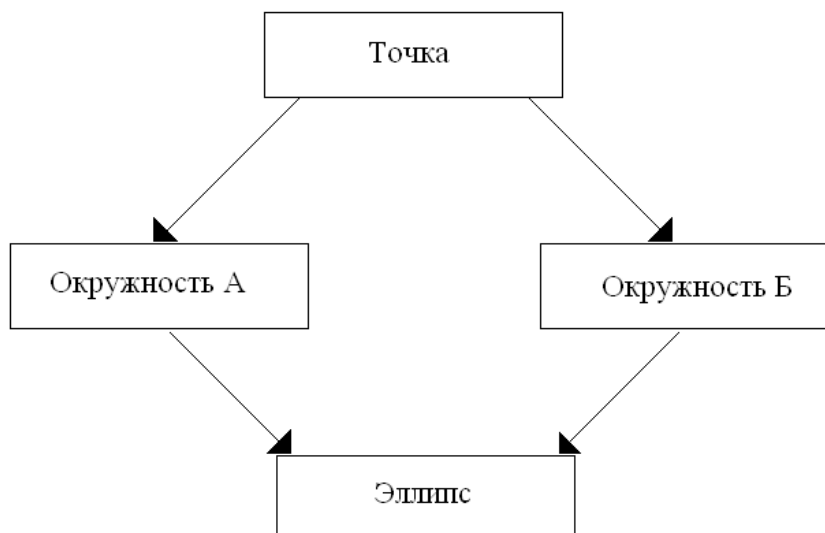


Рисунок 3.16



Определить площадь каждой фигуры. Инициализация данных производится пользователем.

18. Определить следующую иерархию классов (рисунок 3.17).

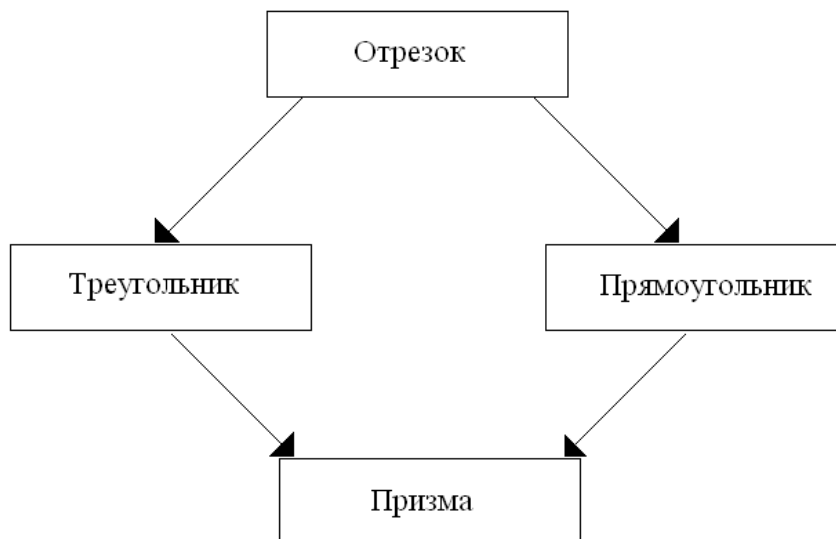


Рисунок 3.17

Определить площадь каждой фигуры (для призмы – площадь боковой поверхности и объём). Треугольники выступают основаниями призмы, а прямоугольники – сторонами. Инициализация данных производится пользователем.

19. Определить следующую иерархию классов (рисунок 3.18).

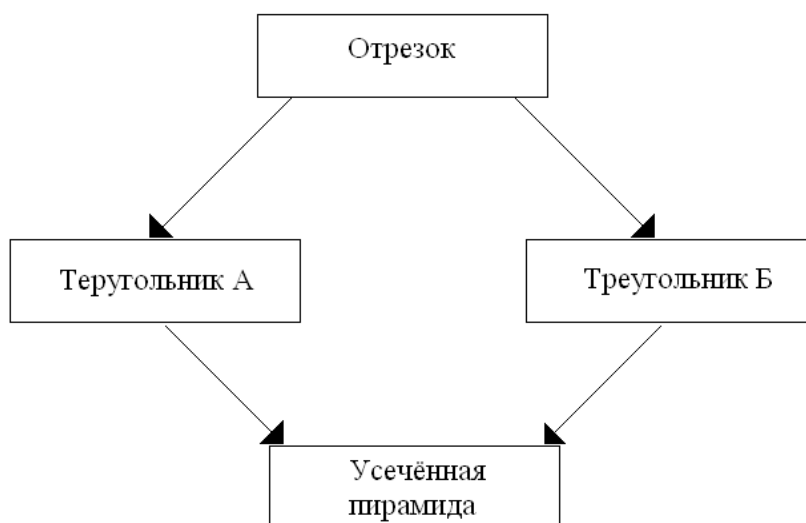


Рисунок 3.18



Определить площадь каждой фигуры (для усечённой пирамиды – площадь боковой поверхности и объём). Инициализация данных производится пользователем.

20. Определить базовый класс «Кости домино», в котором хранится некоторое количество костей в виде « $a/b$ ». В дочернем классе определить метод для правильного упорядочивания костей, или вывести сообщение о невозможности выполнения такого упорядочивания.

21. Определить следующую иерархию классов (рисунок 3.19).

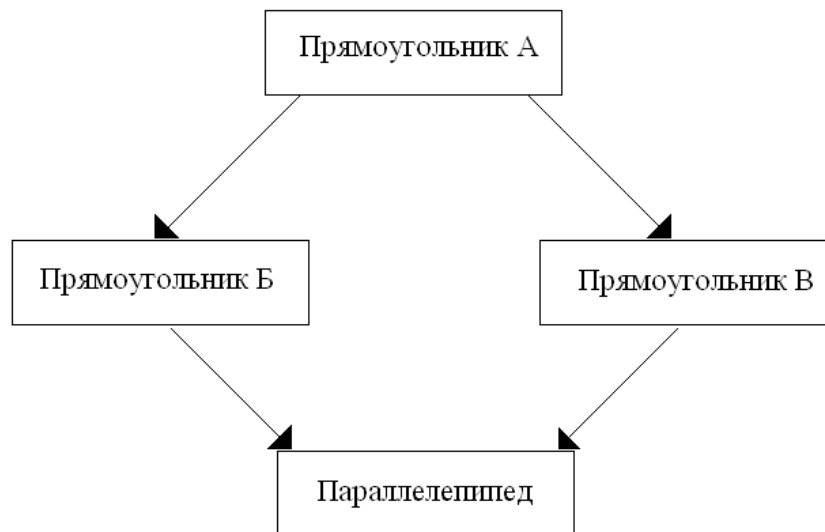


Рисунок 3.19

Определить площадь каждой фигуры (для параллелепипеда – площадь боковой поверхности и объём). Инициализация данных производится пользователем.



22. Определить следующую иерархию классов (рисунок 3.20).

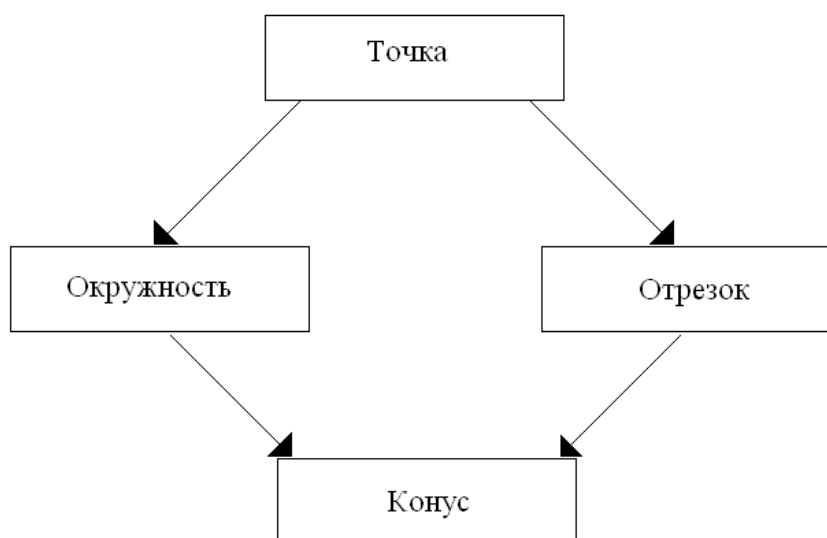


Рисунок 3.20

Определить площадь каждой фигуры (для конуса – площадь боковой поверхности и объём). Инициализация данных производится пользователем.

23. Определить базовый класс «Список», и дочерние классы «Двусвязный список» и «Циклический список». В базовом классе определить методы добавления и удаления элементов в список, а в дочерних – переопределить их соответствующим образом.

24. Определить базовый класс «Прогрессия», в котором хранится первый элемент прогрессии, и дочерние классы «Арифметическая прогрессия» и «Геометрическая прогрессия». На основе классов «Арифметическая прогрессия» и «Геометрическая прогрессия» создать новый класс – «Арифметико-геометрическая прогрессия». В базовом классе определить чисто виртуальную функцию нахождения  $n$ -го элемента прогрессии и переопределить ее в производных.



25. Определить следующую иерархию классов (рисунок 3.21).

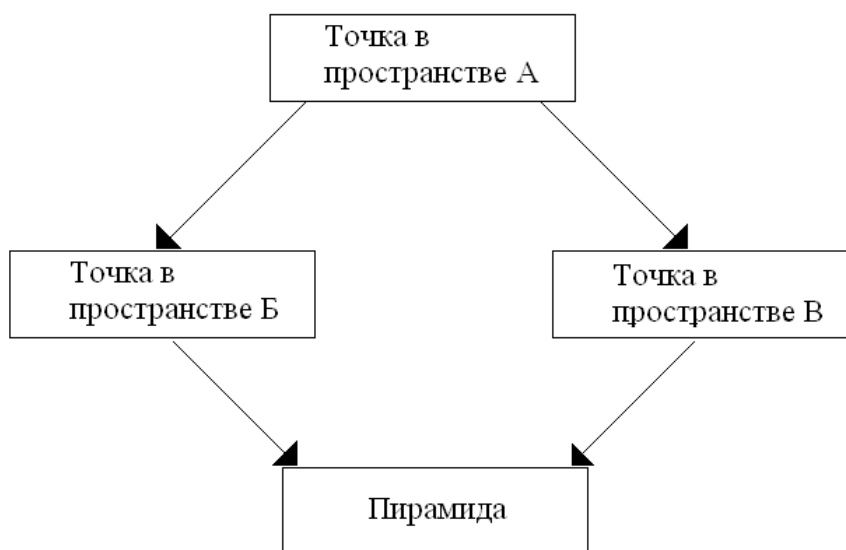


Рисунок 3.21

Определить площадь боковой поверхности и объём пирамиды. Инициализация данных производится пользователем.

26. Определить следующую иерархию классов (рисунок 3.22).

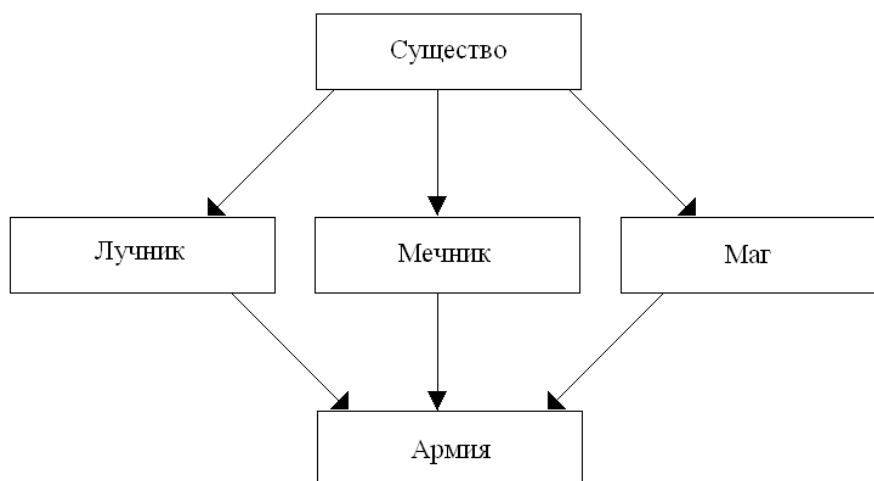


Рисунок 3.22

Определить в наследниках базового класса собственные характеристики каждого существа (здоровье, урон, защиту). В классе «Армия» определить методы для комплектации армии разными существами (до 50 существ) и методы имитации боя. Правила боя разработать самостоятельно. Определить лучшую комплектацию армии.



27. Определить следующую иерархию классов (рисунок 3.23).

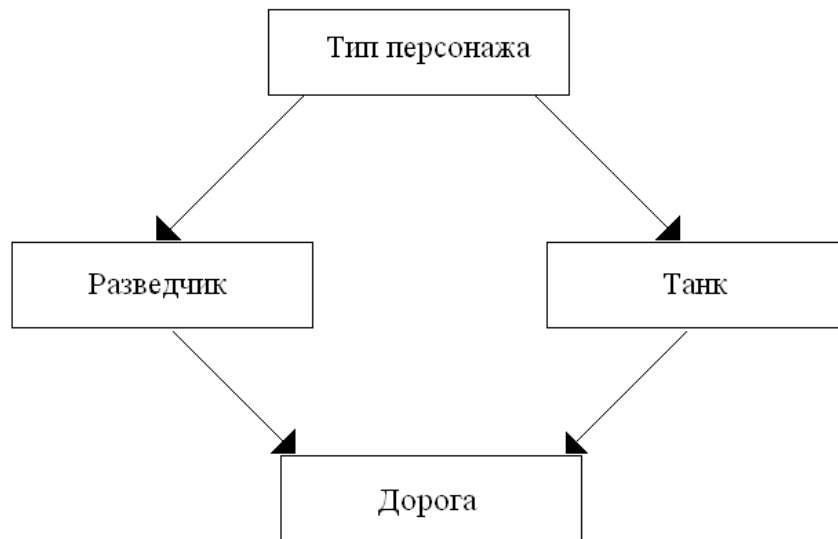


Рисунок 3.23

Класс «Дорога» содержит дорогу с препятствиями, по которой должны пройти два персонажа. Длина дороги минимум 100 ед. Количество ходов каждого персонажа определяется подбрасыванием игрального кубика. При этом разведчик ходит удвоенное количество ходов, но останавливается, достигнув препятствия. Танк ходит обычное число ходов, но игнорирует препятствия. Персонажи ходят по очереди. Разведчик всегда ходит первым. Промоделировать несколько полных прохождений дороги и определить процент побед каждого персонажа.

28. Определить базовый класс «Обход графа», и дочерние классы «Обход в глубину» и «Обход в ширину». В базовом классе определить граф одним из способов. В дочерних классах определить методы обхода графа в глубину и в ширину.



29. Определить следующую иерархию классов (рисунок 3.24).

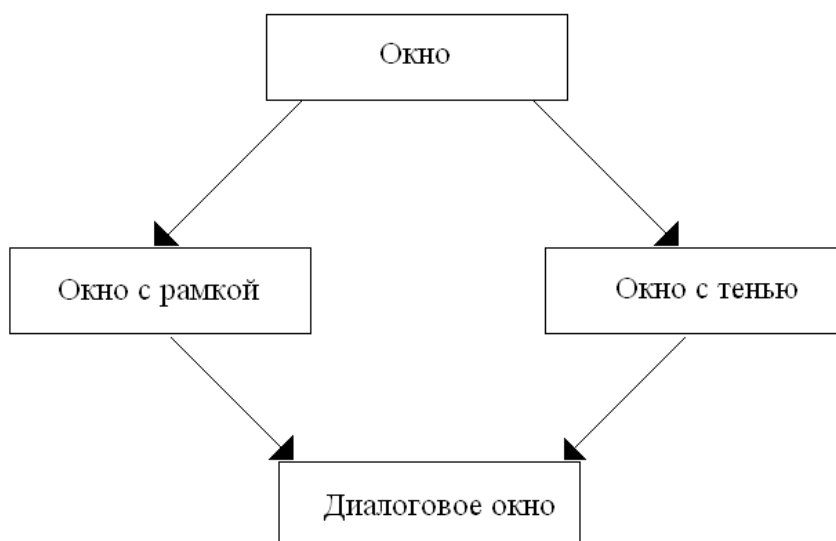


Рисунок 3.24

Для класса «Окно с рамкой» определить цвет рамки. Для класса «Окно с тенью» определить цвет тени. Для класса «Диалоговое окно» определить текст диалогового сообщения и реакцию на сообщение (в виде кнопок).

30. Определить следующую иерархию классов (рисунок 3.25).

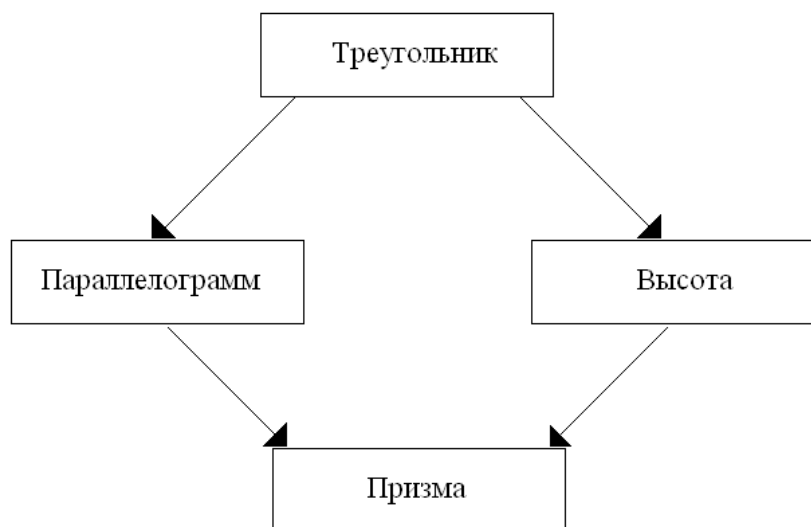


Рисунок 3.25

Определить площадь каждой фигуры (для призмы – площадь боковой поверхности и объём). Параллелограммы выступают основаниями призмы. Инициализация данных производится пользователем.



## 4 ТРЕБОВАНИЯ К ОТЧЕТУ

Отчет подается после полной сдачи и защиты лабораторной работы в электронном виде (документ Word).

Отчет должен быть оформлен согласно ДСТУ 3008-95.

В отчет должен содержать следующие пункты:

- титульный лист;
- содержание;
- цель работы;
- постановка задачи;
- аналитические выкладки, если необходимо;
- пошаговый алгоритм решения задачи;
- UML-диаграмму классов;
- исходный код программы с комментариями;
- примеры работы программы;
- выводы.

**Отчет оценивается максимально в 0,5 балла.**



## 5 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое наследование?
2. Что такое множественное наследование?
3. Что такое виртуальная функция?
4. В чём разница между виртуальной и чисто виртуальной функциями?
5. Что такое абстрактный класс?

