Министерство цифрового развития, связи и массовых коммуникаций
Российской Федерации Сибирский государственный университет телекоммуникаций и информатики

кафедра ТС и ВС

Лабораторная работа 2

по дисциплине

«Объектно-ориентированное программирование»

Выполнил: Демин С. А.
Группа: ИКС-433
Вариант: 5

Проверил: Нейдорф П. Я.

Новосибирск, 2025

Класс-наследник

Цель работы: Освоить основные принципы объектно-ориентированного программирования, изучить понятия базового класса и класса-наследника, научиться применять динамический вызов методов, ознакомиться с принципом полиморфизма.

Задание: Определить суперкласс в соответствии с вариантом задания. Описать поля и методы класса. Создать класс Main с методом main() для проверки работы методов класса. Определить дочерний класс в соответствии с 2 вариантом задания. Определить дополнительные поля и добавить механизм их инициализации в конструкторе. Переопределить метод вывода информации об объекте в дочернем классе. Скомпилировать проект и продемонстрировать работу методов.

Вариант:

№	Суперкласс	Поля
5	Создать класс четырехугольник. Поля класса – координаты 4-х точек. Предусмотреть в классе методы проверки существования четырехугольника, Вычисления длин сторон, периметра и вывода сведений о фигуре.	Параллелограмм. Предусмотреть в классе проверку, является ли фигура параллелограммом.
Написать программу, демонстрирующую работу с подклассом: дано N параллелограммов. Найти среднюю площадь и параллелограмм наименьшей и наибольшей площади.		

Текст основного класса:

```
class Quadrangle {
protected:
   double x1,y1,x2,y2,x3,y3,x4,y4;

public:
   Quadrangle(double X1,double Y1,double X2,double Y2,double X3,double
Y3,double X4,double Y4):
      x1(X1),y1(Y1),x2(X2),y2(Y2),x3(X3),y3(Y3),x4(X4),y4(Y4){}

      double length(double xa, double ya,double xb,double yb)const{
         return sqrt((xb-xa)*(xb-xa)+(yb-ya)*(yb-ya));}

      virtual double perimeter() const {
         return
length(x1,y1,x2,y2)+length(x2,y2,x3,y3)+length(x3,y3,x4,y4)+length(x4,y4,x1,y1)
;}
```

```
    virtual double area() const {
        return abs((x1*y2+x2*y3+x3*y4+x4*y1)-
(y1*x2+y2*x3+y3*x4+y4*x1))/2.0;}

    virtual void show() const {
        cout<< "Координаты: ("<<x1<<","<<y1<<"), ("<<x2<<","<<y2<<"),
("<<x3<<","<<y3<<"), ("<<x4<<","<<y4<<")" << endl;
        cout<< "Периметр: "<< perimeter() <<endl;
        cout<< "Площадь: "<< area()<<endl;}};
```

Текст класса-наследника:

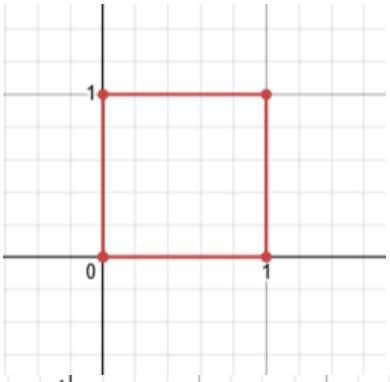
```
class Parallelogram:public Quadrangle {
private:
    string name;

public:
    Parallelogram(double a,double b,double c,double d,double e,double
f,double g,double h,string n=""):Quadrangle(a,b,c,d,e,f,g,h),name(n){}

    string getName()const {return name;}
    void setName(string n){name=n;}

    bool isPara()const{
        double xab=x2-x1,yab=y2-y1;
        double xcd=x4-x3,ycd=y4-y3;
        double xbc=x3-x2,ybc=y3-y2;
        double xda=x1-x4,yda=y1-y4;
        bool ab_parallel_cd =abs(xab*ycd-yab*xcd)<0.0001;
        bool bc_parallel_da =abs(xbc*yda-ybc*xda)<0.0001;

        return ab_parallel_cd && bc_parallel_da;}

    void show() const override {
        cout<<"=== "<<name<<" ==="<<endl;
        bool valid_para=isPara();
        cout<<"Является параллелограммом:
"<<(valid_para?"ДА":"НЕТ")<<endl;

        if (valid_para) {
            Quadrangle::show();
        } else {
            cout<<"Фигура не является параллелограммом"<<endl;
            cout<<"Координаты: ("<<x1<<","<<y1<<"), ("<<x2<<","<<y2<<"),
("<<x3<<","<<y3<<"), ("<<x4<<","<<y4<<")"<< endl;}
        cout << "----------------------" << endl;}};
```

Текст класса main:

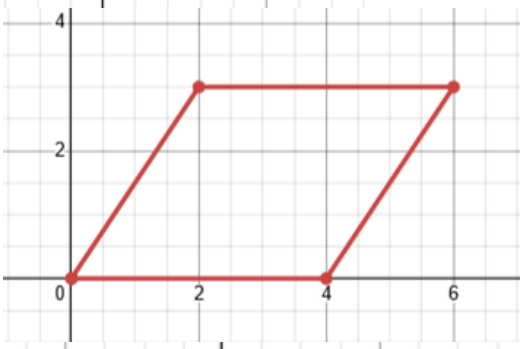
```
class Main {
public:
   static void main(){
      int figureCount;
      cout<<"Введите количество четырехугольников: ";
      cin>>figureCount;
      Quadrangle** figures=new Quadrangle*[figureCount];
      for (int i=0;i<figureCount;i++) {
         double x1,y1,x2,y2,x3,y3,x4,y4;

         cout<< endl<<"Фигура "<<i+1<<":"<<endl;
         cout<<"Введите координаты четырехугольника:" << endl;
         cout<<"Точка 1 (x y): "; cin>>x1>>y1;
         cout<<"Точка 2 (x y): "; cin>>x2>>y2;
         cout<<"Точка 3 (x y): "; cin>>x3>>y3;
         cout<<"Точка 4 (x y): "; cin>>x4>>y4;

         int type;
         cout<< "Выберите тип фигуры (1-четырехугольник,2-
параллелограм): ";
         cin>> type;

         if (type == 2) {
            string name;
            cout << "Введите название параллелограмма: ";
            cin.ignore();
            getline(cin, name);
            figures[i]=new Parallelogram(x1,y1,x2,y2,x3,y3,x4,y4,name);
         } else {
            figures[i]=new Quadrangle(x1,y1,x2,y2,x3,y3,x4,y4);}}

      cout<<endl<<"ВЫВОД ИНФОРМАЦИИ О ФИГУРАХ:"<<endl;
      cout<<"============================"<<endl;

      for (int i = 0; i < figureCount; i++) {
         cout << "Фигура " << i + 1 << ":" << endl;
         figures[i]->show();}}};
```

Фигуры, подающиеся на вход программы:

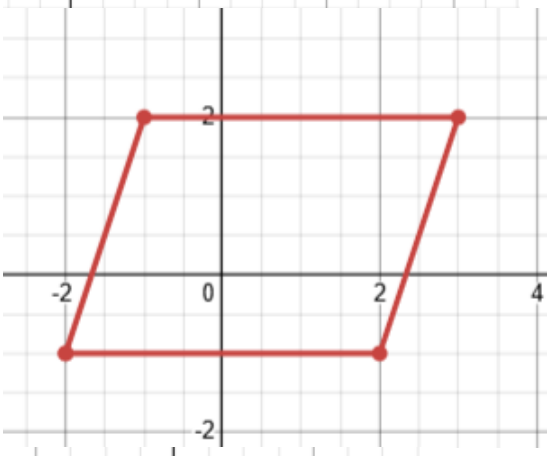
1.



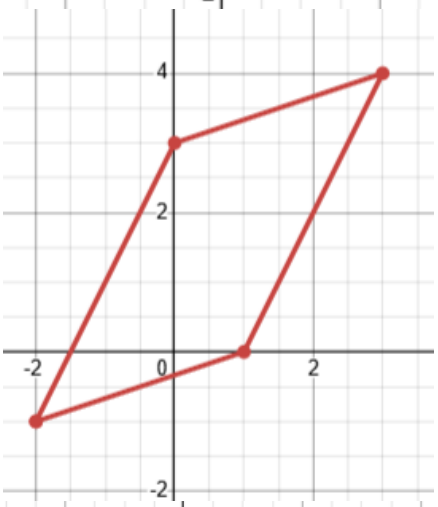
2.



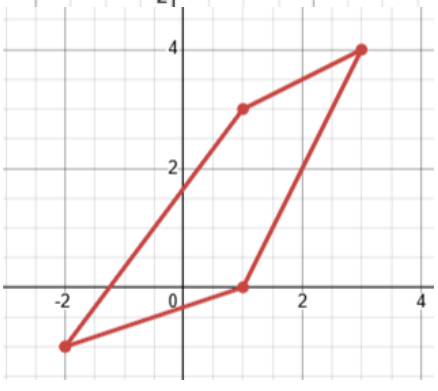
3.



4.



5.



Результаты:

Введите количество четырехугольников: 5

Фигура 1:
Введите координаты четырехугольника:
Точка 1 (x y): 0 0
Точка 2 (x y): 0 1
Точка 3 (x y): 1 1
Точка 4 (x y): 1 0
Выберите тип фигуры (1-четырехугольник,2-парралелограм): 2
Введите название параллелограмма: first

Фигура 2:
Введите координаты четырехугольника:
Точка 1 (x y): 0 0
Точка 2 (x y): 2 3
Точка 3 (x y): 6 3
Точка 4 (x y): 4 0
Выберите тип фигуры (1-четырехугольник,2-парралелограм): 2
Введите название параллелограмма: second

Фигура 3:
Введите координаты четырехугольника:
Точка 1 (x y): -2 -1
Точка 2 (x y): -1 2
Точка 3 (x y): 3 2
Точка 4 (x y): 2 -1
Выберите тип фигуры (1-четырехугольник,2-парралелограм): 2
Введите название параллелограмма: thirt

Фигура 4:
Введите координаты четырехугольника:
Точка 1 (x y): -2 -1
Точка 2 (x y): 0 3
Точка 3 (x y): 3 4
Точка 4 (x y): 1 0
Выберите тип фигуры (1-четырехугольник,2-парралелограм): 2
Введите название параллелограмма: fourth

Фигура 5:
Введите координаты четырехугольника:
Точка 1 (x y): -2 -1
Точка 2 (x y): 1 3
Точка 3 (x y): 3 4
Точка 4 (x y): 1 0
Выберите тип фигуры (1-четырехугольник,2-парралелограм): 2
Введите название параллелограмма: fifth

ВЫВОД ИНФОРМАЦИИ О ФИГУРАХ:
==========
Фигура 1:
=== first ===
Является параллелограммом: ДА
Координаты: (0,0), (0,1), (1,1), (1,0)
Периметр: 4
Площадь: 1

Фигура 2:
=== second ===

Является параллелограммом: ДА
Координаты: (0,0), (2,3), (6,3), (4,0)
Периметр: 15.2111
Площадь: 12

Фигура 3:
=== thirt ===
Является параллелограммом: ДА
Координаты: (-2,-1), (-1,2), (3,2), (2,-1)
Периметр: 14.3246
Площадь: 12

Фигура 4:
=== fourth ===
Является параллелограммом: ДА
Координаты: (-2,-1), (0,3), (3,4), (1,0)
Периметр: 15.2688
Площадь: 10

Фигура 5:
=== fifth ===
Является параллелограммом: НЕТ
Фигура не является параллелограммом
Координаты: (-2,-1), (1,3), (3,4), (1,0)

Описание результатов:

В ходе выполнения лабораторной работы была разработана программа для работы с геометрическими фигурами - четырехугольниками и параллелограммами. Пользователь вводит координаты вершин фигур и выбирает их тип. Программа корректно определяет, является ли фигура параллелограммом, вычисляет периметр и площадь для каждой фигуры. На примере пяти фигур продемонстрирована работа программы: для первых четырех фигур, которые являются параллелограммами, выводятся все характеристики (координаты, периметр, площадь), а для пятой фигуры, не являющейся параллелограммом, выводится соответствующее сообщение. Программа успешно обрабатывает различные конфигурации точек и правильно идентифицирует тип фигуры.

Выводы:

В результате выполнения работы были успешно освоены основные принципы объектно-ориентированного программирования: создан базовый класс Quadrangle и класс-наследник Parallelogram, реализован механизм наследования, продемонстрирован принцип полиморфизма через переопределение метода show() в классе-наследнике. Динамический вызов методов обеспечил правильное отображение информации о фигурах в зависимости от их типа. Работа подтвердила эффективность использования наследования для создания иерархии классов и преимущества полиморфизма для обработки объектов разных типов единообразным способом.

Контрольные вопросы:

1. Охарактеризуйте механизм наследования.

Наследование позволяет создавать новый класс на основе существующего. Новый класс (подкласс) наследует все поля и методы родительского класса (суперкласса) и может добавлять свои собственные. Это как ребенок наследует черты родителей, но может иметь и свои особенности.

2. Что такое суперкласс и подкласс?

- Суперкласс (родительский класс) - это основной класс, от которого наследуются другие классы
- Подкласс (дочерний класс) - это класс, который наследует свойства и методы от суперкласса

3. Объясните, что такое инкапсуляция и полиморфизм.

- Инкапсуляция - это сокрытие внутренней реализации класса от внешнего мира. Как черный ящик - мы знаем, что он делает, но не знаем как именно.
- Полиморфизм - это возможность использовать одинаковые методы для разных классов. Например, метод show() работает по-разному для четырехугольника и параллелограмма.

4. Чем отличаются ранее и позднее связывание?

- Раннее связывание - компилятор заранее знает, какой метод вызывать
- Позднее связывание - решение о вызове метода принимается во время выполнения программы (используется virtual)

Весь код:

```
#include <iostream>
#include <cmath>
#include <string>
using namespace std;

class Quadrangle {
protected:
    double x1,y1,x2,y2,x3,y3,x4,y4;

public:
    Quadrangle(double X1,double Y1,double X2,double Y2,double X3,double
Y3,double X4,double Y4):
    x1(X1),y1(Y1),x2(X2),y2(Y2),x3(X3),y3(Y3),x4(X4),y4(Y4){}

    double length(double xa, double ya,double xb,double yb)const{
        return sqrt((xb-xa)*(xb-xa)+(yb-ya)*(yb-ya));}

    virtual double perimeter() const {
        return
length(x1,y1,x2,y2)+length(x2,y2,x3,y3)+length(x3,y3,x4,y4)+length(x4,y4,x1,y1)
;}

    virtual double area() const {
        return abs((x1*y2+x2*y3+x3*y4+x4*y1)-
(y1*x2+y2*x3+y3*x4+y4*x1))/2.0;}

    virtual void show() const {
        cout<< "Координаты: ("<<x1<<","<<y1<<"), ("<<x2<<","<<y2<<"),
("<<x3<<","<<y3<<"), ("<<x4<<","<<y4<<")" << endl;
        cout<< "Периметр: "<< perimeter() <<endl;
        cout<< "Площадь: "<< area()<<endl;}};

class Parallelogram:public Quadrangle {
private:
    string name;

public:
    Parallelogram(double a,double b,double c,double d,double e,double
f,double g,double h,string n=""):Quadrangle(a,b,c,d,e,f,g,h),name(n){}

    string getName()const {return name;}
    void setName(string n){name=n;}

    bool isPara()const{
        double xab=x2-x1,yab=y2-y1;
        double xcd=x4-x3,ycd=y4-y3;
```

```
            double xbc=x3-x2,ybc=y3-y2;
            double xda=x1-x4,yda=y1-y4;
            bool ab_parallel_cd =abs(xab*ycd-yab*xcd)<0.0001;
            bool bc_parallel_da =abs(xbc*yda-ybc*xda)<0.0001;

            return ab_parallel_cd && bc_parallel_da;}

        void show() const override {
            cout<<"=== "<<name<<" ==="<<endl;
            bool valid_para=isPara();
            cout<<"Является параллелограммом:
"<<(valid_para?"ДА":"НЕТ")<<endl;

            if (valid_para) {
                Quadrangle::show();
            } else {
                cout<<"Фигура не является параллелограммом"<<endl;
                cout<<"Координаты: ("<<x1<<","<<y1<<"), ("<<x2<<","<<y2<<"),
("<<x3<<","<<y3<<"), ("<<x4<<","<<y4<<")"<< endl;}
            cout << "-----------------------" << endl;}};

    class Main {
    public:
        static void main(){
            int figureCount;
            cout<<"Введите количество четырехугольников: ";
            cin>>figureCount;
            Quadrangle** figures=new Quadrangle*[figureCount];
            for (int i=0;i<figureCount;i++) {
                double x1,y1,x2,y2,x3,y3,x4,y4;

                cout<< endl<<"Фигура "<<i+1<<":"<<endl;
                cout<<"Введите координаты четырехугольника:" << endl;
                cout<<"Точка 1 (x y): "; cin>>x1>>y1;
                cout<<"Точка 2 (x y): "; cin>>x2>>y2;
                cout<<"Точка 3 (x y): "; cin>>x3>>y3;
                cout<<"Точка 4 (x y): "; cin>>x4>>y4;

                int type;
                cout<< "Выберите тип фигуры (1-четырехугольник,2-
парралелограм): ";
                cin>> type;

                if (type == 2) {
                    string name;
                    cout << "Введите название параллелограмма: ";
                    cin.ignore();
```

```
            getline(cin, name);
            figures[i]=new Parallelogram(x1,y1,x2,y2,x3,y3,x4,y4,name);
        } else {
            figures[i]=new Quadrangle(x1,y1,x2,y2,x3,y3,x4,y4);}}

    cout<<endl<<"ВЫВОД ИНФОРМАЦИИ О ФИГУРАХ:"<<endl;
    cout<<"============================"<<endl;

    for (int i = 0; i < figureCount; i++) {
        cout << "Фигура " << i + 1 << ":" << endl;
        figures[i]->show();}}};

int main() {
    Main::main();}
```