

Задание 1

Circle

package pract41;  
public class Circle extends Shape  
{  
 int radius;  
 public void setRadius(int rad)  
 {  
 radius=rad;  
 }  
 public void chett()  
 {  
 area=Math.*PI* \* radius \* radius;  
 perim=2\*Math.*PI* \* radius;  
 }  
 public int getRadius()  
 {  
 return radius;  
 }  
 public double getArea()  
 {  
 return area;  
 }  
 public double getPerimeter()  
 {  
 return perim;  
 }  
 public String toString()  
 {  
 return "Радиус круга " + getRadius() + "\nПлощадь круга "+ getArea() + "\nПериметр круга " + getPerimeter();  
 }  
}

Rectangle

package pract41;  
  
public class Rectangle extends Shape  
{  
 int a, b;  
 public void setA(int a)  
 {  
 this.a=a;  
 }  
 public void setB(int b)  
 {  
 this.b=b;  
 }  
 public int getA()  
 {  
 return a;  
 }  
 public int getB()  
 {  
 return b;  
 }  
 public void chett()  
 {  
 area = a \* b;  
 perim = 2 \* (a + b);  
 }  
 public double getArea()  
 {  
 return area;  
 }  
 public double getPerimeter()  
 {  
 return perim;  
 }  
 public String toString()  
 {  
 return "Сторона прямоугольника a " + getA() + " b " + getB() + "\nПериметр прямоугольника " + getPerimeter() + "\nПлощадь прямоугольника " + getArea();  
 }  
  
}

Shape

package pract41;  
public class Shape  
{  
 protected String type;  
 protected double area, perim;  
 public String getType()  
 {  
 return "";  
 }  
 public double getArea()  
 {  
 return 0;  
 }  
 public double getPerimeter()  
 {  
 return 0;  
 }  
 public String toString()  
 {  
 return "";  
 }  
 public void chett()  
 {  
  
 }  
}

Square

package pract41;  
  
public class Square extends Shape  
{  
 int a;  
 public void setA(int a)  
 {  
 this.a=a;  
 }  
 public int getA()  
 {  
 return a;  
 }  
 public void chett()  
 {  
 area = a \* a ;  
 perim = 4 \* a;  
 }  
 public double getArea()  
 {  
 return area;  
 }  
 public double getPerimeter()  
 {  
 return perim;  
 }  
 public String toString()  
 {  
 return "Сторона квадрата a " + getA() + "\n Площадь квадрата " + getArea() + "\n Периметр квадрата " + getPerimeter();  
 }  
}

Test

package pract41;  
import java.util.Scanner;  
public class Test  
{  
 public static void main(String[] args)  
 {  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.println("Введите радиус для круга");  
 Circle c = new Circle();  
 Shape obj = c; //родительская ссылка  
 c.setRadius(in.nextInt());  
 obj.chett();  
 System.*out*.print(c);  
  
 Rectangle r = new Rectangle();  
 System.*out*.println("\nВведите стороны прямоугольника ");  
 r.setA(in.nextInt());  
 r.setB(in.nextInt());  
 obj=r;  
 obj.chett();  
 System.*out*.print(r);  
  
  
 Square s = new Square();  
 obj = s;  
 System.*out*.println("\nВведите сторону квадрата");  
 s.setA(in.nextInt());  
 obj.chett();  
 System.*out*.print(s);  
 }  
}

Задание 2

Phone.java

package pract41\_2;  
  
public class Phone  
{  
 String number, model, weight;  
 Phone(String model, String weight, String number)  
 {  
 this(model, number);  
 this.weight=weight;  
 }  
 Phone(String model, String number)  
 {  
 this.model=model;  
 this.number=number;  
 }  
 Phone()  
 {  
  
 }  
  
 public void setModel(String model) {  
 this.model = model;  
 }  
  
 public String getModel() {  
 return model;  
 }  
  
 public void setWeight(String weight) {  
 this.weight = weight;  
 }  
  
 public String getWeight() {  
 return weight;  
 }  
  
 public void setNumber(String number) {  
 this.number = number;  
 }  
  
 public String getNumber() {  
 return number;  
 }  
 public String toString()  
 {  
 return "Модель телефона " + getModel() + " Вес телефона " + getWeight() + " Номер телефона " + getNumber();  
 }  
  
 public void receiveCall(String name\_caller)  
 {  
 System.*out*.println("Звонит " + name\_caller);  
 }  
 public void receiveCall(String name\_caller, String number\_caller)  
 {  
 System.*out*.println("Звонит " + name\_caller + ", номер " + number\_caller);  
 }  
 public void sendMessage(String message, String ... num)  
 {  
 System.*out*.println("Сообщение: {" + message + "} будет отправлено этим номерам:");  
 for (String x : num)  
 System.*out*.println(x);  
 }  
}

Test java

package pract41\_2;  
import java.util.Scanner;  
public class Test  
{  
 public static void main(String[] args)  
 {  
 Scanner in = new Scanner(System.*in*);  
 String number, weight, model;  
 System.*out*.println("Введите характеристики 1 телефона");  
 System.*out*.println("Введите модель телефона");  
 model= in.next();  
 System.*out*.println("Введите вес телефона");  
 weight= in.next();  
 System.*out*.println("Введите номер телефона");  
 number= in.next();  
 Phone obj1 = new Phone(model, weight, number);  
  
 System.*out*.println("Введите характеристики 2 телефона");  
 System.*out*.println("Введите модель телефона");  
 model= in.next();  
 System.*out*.println("Введите номер телефона");  
 number= in.next();  
 Phone obj2 = new Phone(model, number);  
 System.*out*.println("Введите вес телефона");  
 weight=in.next();  
 obj2.setWeight(weight);  
  
  
 System.*out*.println("Введите характеристики 3 телефона");  
 System.*out*.println("Введите модель телефона");  
 model= in.next();  
 System.*out*.println("Введите вес телефона");  
 weight= in.next();  
 System.*out*.println("Введите номер телефона");  
 number= in.next();  
 Phone obj3 = new Phone(model, weight, number);  
  
 System.*out*.println(obj1);  
 System.*out*.println(obj2);  
 System.*out*.println(obj3);  
 String p, d;  
 System.*out*.println("Введите имя звонящего");  
 p=in.next();  
 obj1.receiveCall(p);  
  
 System.*out*.println("Введите имя звонящего");  
 p=in.next();  
 System.*out*.println("Введите телефон звонящего");  
 d=in.next();  
 obj1.receiveCall(p, d);  
 System.*out*.println("Введите cообщение ");  
 p=in.next();  
 obj1.sendMessage(p, d, obj2.getNumber(), obj3.getNumber());  
 }  
}

Задание 8

Circle

package pract41\_8;  
  
public class Circle extends Shape  
{  
 double radius;  
 Circle()  
 {  
  
 }  
 Circle(double \_radius)  
 {  
 radius = \_radius;  
  
 }  
 Circle(double \_radius, String \_color, boolean \_filled)  
 {  
 this(\_radius);  
 color=\_color;  
 filled=\_filled;  
 }  
  
 public void setRadius(double radius) {  
 this.radius = radius;  
 }  
 public double getRadius() {  
 return radius;  
 }  
  
 @Override  
 public double getArea()  
 {  
 return Math.*PI* \* radius \* radius;  
 }  
  
 @Override  
 public double getPerimeter()  
 {  
 return 2\*Math.*PI* \* radius;  
 }  
  
 @Override  
 public String toString()  
 {  
 return "Радиус: " + getRadius()+ ", color: " + getColor() + ", filled: " + isFilled() + ", Периметр: " + getPerimeter() + ", Площадь: " + getArea();  
 }  
}

Rectangle

package pract41\_8;  
  
public class Rectangle extends Shape  
{  
 double width, length;  
 Rectangle()  
 {  
  
 }  
 Rectangle(double \_width, double \_length)  
 {  
 width=\_width;  
 length=\_length;  
  
 }  
 Rectangle(double \_width, double \_length, String \_color, boolean \_filled)  
 {  
 this(\_width,\_length);  
 color=\_color;  
 filled=\_filled;  
 }  
  
 public void setLength(double \_length) {  
 length = \_length;  
 }  
  
 public void setWidth(double \_width) {  
 width = \_width;  
 }  
  
 public double getLength() {  
 return length;  
 }  
  
 public double getWidth() {  
 return width;  
 }  
 @Override  
 public double getArea()  
 {  
 return length \* width;  
 }  
  
 @Override  
 public double getPerimeter()  
 {  
 return 2 \* (length + width);  
 }  
  
 @Override  
 public String toString()  
 {  
 return "ширина: " + getWidth() + ", длина " + getLength() + ", color: " + getColor() + ", filled: " + isFilled() + ", Периметр: " + getPerimeter() + ", Площадь: " + getArea();  
 }  
}

Shape

package pract41\_8;  
  
public abstract class Shape  
{  
 protected String color = "нет данных";  
 protected boolean filled = false;  
 Shape()  
 {  
  
 }  
 Shape(String \_color, boolean \_filled)  
 {  
 color=\_color;  
 filled=\_filled;  
 }  
  
 public void setColor(String color)  
 {  
 this.color = color;  
 }  
  
 public void setFilled(boolean filled)  
 {  
 this.filled = filled;  
 }  
  
 public String getColor()  
 {  
 return color;  
 }  
  
 public boolean isFilled()  
 {  
 return filled;  
 }  
 public abstract double getArea();  
 public abstract double getPerimeter();  
 public abstract String toString();  
}

Square

package pract41\_8;  
  
public class Square extends Rectangle  
{  
 double side;  
 Square()  
 {  
  
 }  
 Square(double \_side)  
 {  
 side=\_side;  
 }  
 Square(double \_side, String \_color, boolean \_filled)  
 {  
 super(\_side,\_side, \_color, \_filled);  
 side=\_side;  
 color=\_color;  
 filled=\_filled;  
 }  
  
 public void setSide(double side) {  
 this.side = side;  
 }  
  
 public double getSide() {  
 return side;  
 }  
  
 @Override  
 public void setWidth(double \_side) {  
 side=\_side;  
 super.setWidth(side);  
 }  
  
 @Override  
 public void setLength(double \_side) {  
 side=\_side;  
 super.setLength(side);  
 }  
  
 @Override  
 public String toString() {  
 return "side: " + getSide() + ", color: " + getColor() + ", filled: " + isFilled() +", Периметр: " + super.getPerimeter() + ", Площадь: " + super.getArea();  
 }  
}

Test

package pract41\_8;  
import java.util.Scanner;  
public class Test  
{  
 public static void main(String[] args)  
 {  
 Scanner in = new Scanner(System.*in*);  
 Circle obj1 = new Circle(5,"red", true);  
 Rectangle obj2 = new Rectangle(5, 6,"red", true);  
 Square obj3 = new Square();  
 System.*out*.println("Круг \n" + obj1);  
 System.*out*.println("Прямоугольник \n" + obj2);  
 obj3.setLength(25);  
 obj3.setWidth(25);  
 System.*out*.println("Квадрат \n" + obj3);  
 }  
  
}

Упражнение 1

Test\_8\_zad

package pract41\_8;  
import java.util.Scanner;  
public class Test\_8\_zad {  
 public static void main(String[] args)  
 {  
 Shape s1 = new Circle(5.5, "RED", false); // родительская ссылка  
 System.*out*.println(s1);  
 System.*out*.println(s1.getArea());  
 System.*out*.println(s1.getPerimeter());  
 System.*out*.println(s1.getColor());  
 System.*out*.println(s1.isFilled());  
 //System.out.println(s1.getRadius()); //нельзя использовать метод который не наследуется из Shape  
 Circle c1 = (Circle)s1;  
 System.*out*.println(c1);  
 System.*out*.println(c1.getArea());  
 System.*out*.println(c1.getPerimeter());  
 System.*out*.println(c1.getColor());  
 System.*out*.println(c1.isFilled()); //  
 System.*out*.println(c1.getRadius());  
 //Shape s2 = new Shape(); //абстрактный класс нельзя инициализировать  
 Shape s3 = new Rectangle(1.0, 2.0, "RED", false);//  
 System.*out*.println(s3);  
 System.*out*.println(s3.getArea());  
 System.*out*.println(s3.getPerimeter());  
 System.*out*.println(s3.getColor());  
 //System.out.println(s3.getLength()); //нельзя использовать метод который не наследуется из Shape  
 Rectangle r1 = (Rectangle)s3;  
 System.*out*.println(r1);  
 System.*out*.println(r1.getArea());  
 System.*out*.println(r1.getColor());  
 System.*out*.println(r1.getLength());  
 Shape s4 = new Square(6.6);  
 System.*out*.println(s4);  
 System.*out*.println(s4.getArea());  
 System.*out*.println(s4.getColor());  
 //System.out.println(s4.getSide()); //не наследуемый метод  
  
/\*обратите внимание, что выполняем downcast Shape s4 к  
Rectangle, который является суперклассом  
Square(родителем), вместо Square \*/  
 Rectangle r2 = (Rectangle)s4;  
 System.*out*.println(r2); System.*out*.println(r2.getArea());  
 System.*out*.println(r2.getColor());  
 //System.out.println(r2.getSide()); //нельзя вызвать метод потомка из родителя  
 System.*out*.println(r2.getLength());  
// Downcast Rectangle r2 к Square  
 Square sq1 = (Square)r2;  
 System.*out*.println(sq1);  
 System.*out*.println(sq1.getArea());  
 System.*out*.println(sq1.getColor());  
 System.*out*.println(sq1.getSide());  
 System.*out*.println(sq1.getLength());  
  
 }  
}

До изменений

package pract41\_8;  
import java.util.Scanner;  
public class Test\_8\_zad {  
 public static void main(String[] args)  
 {  
 Shape s1 = new Circle(5.5, “RED”, false); // Upcast   
 Circle to Shape System.out.println(s1);//which version?  
 System.out.println(s1.getArea());// which version?  
 System.out.println(s1.getPerimeter());//which version?  
 System.out.println(s1.getColor());  
 System.out.println(s1.isFilled());  
 System.out.println(s1.getRadius());  
 Circle c1 = (Circle)s1;// Downcast back to Circle  
 System.out.println(c1);  
 System.out.println(c1.getArea());  
 System.out.println(c1.getPerimeter());  
 System.out.println(c1.getColor());  
 System.out.println(c1.isFilled());  
 System.out.println(c1.getRadius());  
 Shape s2 = new Shape();  
 Shape s3 = new Rectangle(1.0, 2.0, "RED", false);//   
 Upcast System.out.println(s3);  
 System.out.println(s3.getArea());  
 System.out.println(s3.getPerimeter());  
 System.out.println(s3.getColor());  
 System.out.println(s3.getLength());  
 Rectangle r1 = (Rectangle)s3; //downcast  
 System.out.println(r1);  
 System.out.println(r1.getArea());  
 System.out.println(r1.getColor());  
 System.out.println(r1.getLength());  
 Shape s4 = new Square(6.6);//Upcast  
 System.out.println(s4);  
 System.out.println(s4.getArea());  
 System.out.println(s4.getColor());  
 System.out.println(s4.getSide());  
/\*обратите внимание, что выполняем downcast Shape s4 к  
Rectangle, который является суперклассом   
Square(родителем), вместо Square \*//  
 Rectangle r2 = (Rectangle)s4;  
 System.out.println(r2); System.out.println(r2.getArea());  
 System.out.println(r2.getColor());  
 System.out.println(r2.getSide());  
 System.out.println(r2.getLength());  
// Downcast Rectangle r2 к Square  
 Square sq1 = (Square)r2;  
 System.out.println(sq1);  
 System.out.println(sq1.getArea());  
 System.out.println(sq1.getColor());  
 System.out.println(sq1.getSide());  
 System.out.println(sq1.getLength());  
  
  
 }  
}