Chương 2 ĐỐI TƯỢNG VÀ LỚP TRONG JAVA

2.1 Xây dựng lớp Rectangle kế thừa lớp Square

Chương trình chính sử dụng lớp Square.

1. Tập tin Shapes.java

```
class Point {
  private double x, y;
  Point (double x, double y)  {
    this.x = x;
    this.y = y;
  }
  double getX ()  {
    return x;
  }
  double getY ()  {
    return y;
  }
}

class Square {
  private double width;
  Square (double width)  {
    this.width = width;
  }
  double getWidth ()  {
    return width;
  }
}

class Rectangle extends Square {
  private double length;
  Rectangle (double width, double length)  {
    // Truyền giá trị tham số width parameter vào lớp Square để khởi tạo đối tượng
    super (width);
    this.length = length;
  }
  double getLength ()  {
    return length;
  }
}
class Shapes {
  public static void main (String [] args)  {
    Square s = new Square (100);
    System.out.println ("s.width = " + s.getWidth ());
    Rectangle r = new Rectangle (50, 25);
    System.out.println ("r.width = " + r.getWidth ());
    System.out.println ("r.length = " + r.getLength ());
  }
}
```

2.2 Chương trình tạo đối tượng

Chương trình tạo đối tượng từ lớp Rectangle mô tả đối tượng hình chữ nhật. Rectangle.

2. Tập tin CreateObjectDemo.java

```
public class Rectangle {
   public int width = 0;
   public int height = 0;
   public Point origin;

   // four constructors
   public Rectangle() {
         origin = new Point(0, 0);
   }
   public Rectangle(Point p) {
         origin = p;
   }
   public Rectangle(int w, int h) {
         this(new Point(0, 0), w, h);
   }
   public Rectangle(Point p, int w, int h) {
         origin = p;
         width = w;
         height = h;
   }

   // a method for moving the rectangle
   public void move(int x, int y) {
         origin.x = x;
         origin.y = y;
   }
   // a method for computing the area of the rectangle
   public int area() {
         return width * height;
   }
}

public class CreateObjectDemo {
   public static void main(String[] args) {
      // create a point object and two rectangle objects
      Point origin_one = new Point(23, 94);
      Rectangle rect_one = new Rectangle(origin_one, 100, 200);
      Rectangle rect_two = new Rectangle(50, 100);
      // display rect_one's width, height, and area
      System.out.println("Width of rect_one: " + rect_one.width);
      System.out.println("Height of rect_one: " + rect_one.height);
      System.out.println("Area of rect_one: " + rect_one.area());
```

```
        // set rect_two's position
        rect_two.origin = origin_one;

        // display rect_two's position
        System.out.println("X Position of rect_two: " + rect_two.origin.x);
        System.out.println("Y Position of rect_two: " + rect_two.origin.y);
        // move rect_two and display its new position
        rect_two.move(40, 72);
        System.out.println("X Position of rect_two: " + rect_two.origin.x);
        System.out.println("Y Position of rect_two: " + rect_two.origin.y);
    }
}
```

2.3 Chương trình xử lý tính toán trên số phức

Xây dựng lớp Complex mô tả số phức và các phương thức cộng, trừ, in số phức. Chương trình chính thao tác trên đối tượng được tạo ra từ lớp này.

3. Tập tin UseComplex.java

```
class UseComplex {
  public static void main (String [] args)  {
    Complex c1 = new Complex (2.0, 5.0); // 2.0 + 5.0i
    Complex c2 = new Complex (-3.1, -6.3); // -3.1 - 6.3i
    c1.add (c2); // c1 is now -1.1 - 1.3i
    c1.print ();
  }
}

class Complex {
  private double re, im;
 Complex (double real, double imag)  {
    re = real;
    im = imag;
  }

  public double getRe ()  {
    return re;
  }

  public double getIm ()  {
    return im;
  }

  public void add (Complex c) {
    re += c.getRe ();
    im += c.getIm ();
  }

  public void subtract (Complex c)  {
    re -= c.getRe ();
    im -= c.getIm ();
  }
```

```
  public void print ()   {
    System.out.println ("(" + re + "," + im + ")");
  }
}
```

2.4 Minh họa tính đa hình của Java

Cách dùng từ khóa this, sử dụng đối tượng làm tham số của phương thức.

4. Tập tin JavaExample02.java

```
class Box {
        int width, height, depth;
        Box () {
                width  = 0;
                height = 0;
                depth  = 0;
        }

        Box (int width, int height, int depth) {
                this.width  = width;
                this.height = height;
                this.depth  = depth;
        }
        Box (int a) {
                width  = height = depth  = a;
        }

        Box (Box obj) {
                width  = obj.width;
                height = obj.height;
                depth  = obj.depth;
        }
        public int volumeBox() {
                return width * height * depth;
        }
}//end of class

//---------------------------------------------------------------

class JavaExample02 {
        public static void main (String args[]) {
                Box obj1 = new Box();
                Box obj2 = new Box(3,4,5);
                Box obj3 = new Box(3);
                Box obj4 = new Box(obj2);
                System.out.println(">> The tich 1 = " + obj1.volumeBox());
                System.out.println(">> The tich 2 = " + obj2.volumeBox());
                System.out.println(">> The tich 3 = " + obj3.volumeBox());
```

```
                System.out.println(">> The tich 4 = " + obj4.volumeBox());
        }
}//class
```

2.5 Minh họa tính kế thừa của Java

5. Tập tin JavaExample03.java

```
class Box {
        int width, height, depth;
        Box () {
                width  = 0;
                height = 0;
                depth  = 0;
        }
        Box (int width, int height, int depth) {
                this.width  = width;
                this.height = height;
                this.depth  = depth;
        }
        public int volumeBox() {
                return width * height * depth;
        }
}//end of class
//---------------------------------------------------------------
class SubBox extends Box {
//SubBox ke thua cac dac tinh cua Box va co them weight
//SubBox khong can phai tao lai cac dac diem da co trong Box
//Tinh ke thua cho phep co the tao cac lop con rieng biet tu lop Box
        int weight;
        SubBox (int width, int height, int depth, int weight) {
        /* Cach 1
                this.width  = width;
                this.height = height;
                this.depth  = depth;
                this.weight = weight;
        */
        /* Cach 2 */
                super(width, height, depth);
                this.weight = weight;
        }
        public int volumeBox() {
                return width * height * depth;
        }
}//end of class
//---------------------------------------------------------------
class JavaExample03 {
        public static void main (String args[]) {
```

```
                SubBox obj1 = new SubBox(2,3,4,5);
                System.out.println(">> The tich 1 = " + obj1.volumeBox());
                System.out.println(">> Trong luong = " + obj1.weight);
        }
}
```

2.6 Minh họa cách sử dụng lớp abstract

6. Tập tin Circle.java

```
class Point {
// This is a mere sketch
  int   _x;
  int   _y;
  public Point(int x, int y)  {
        _x = x; _y = y;
 }
  public String toString()  {
        return "(" + _x + "," + _y + ")" ;
  } // sample output: "(2,3)"
}

abstract class Shape {
   private Point _origin;
   public Point getOrigin() { return _origin; }
   public Shape() { _origin = new Point(0,0); }
   public Shape(int x, int y) { origin = new Point(x,y); }
   public Shape(Point o) { _origin = o; }
   abstract public void draw(); // deliberately unimplemented
}
class Circle extends Shape {
  double radius;
  public Circle(double rad) {
        super(); radius = rad;
  }
  public Circle(int x, int y, double rad)   {
    super(x,y);
    radius = rad;
  }
  public Circle(Point o, double rad) {
        super(o); radius = rad;
 }
  public void draw()  {
    System.out.println("Circle@" + getOrigin().toString() + ", rad = " + radius);
  }
  static public void main(String argv[])  {
    Circle circle = new Circle(1.0);
```

```
    circle.draw();
  }
}
// Output: Circle@(0,0), rad = 1.0
```

2.7 Minh họa cách sử dụng lớp interface

7. Tập tin : Circle.java

```
interface Figure {
  public void move(Point p) throws Exception;
  public void draw() throws Exception;
  public double area() throws Exception;
  public double PI = 3.14159;
}
abstract class Shape implements Figure {
   private Point origin;
   public Point getOrigin() { return origin; }
   public Shape() { origin = new Point(0,0); }
   public Shape(int x, int y) { origin = new Point(x,y); }
   public Shape(Point o) { origin = o; }
  public void move(Point p) { origin = p; }
}

//Lớp Circe kế thừa từ lớp Shape.
public class Circle extends Shape {
  double radius;
  public Circle(double rad) {
          super(); radius = rad;
  }
  public Circle(int x, int y, double rad)  {
          super(x,y);
          radius = rad;
  }
  public Circle(Point o, double rad) {
          super(o); radius = rad;
 }
  public double area() {
          return PI * radius * radius;
 }
  public void draw()  {
          System.out.println("Circle@" + getOrigin().toString() + ", rad = " + radius);
  }

static public void main(String argv[]) {
Circle circle = new Circle(2.0);
circle.draw();
```

```
circle.move(new Point(1,1));
circle.draw();
System.out.println("circle area = " + circle.area());
  }
}
// Output:
// Circle@(0,0), rad = 2.0
// Circle@(1,1), rad = 2.0
// circle area = 12.56636
```