

Lab 4. Factory Method

Bài 1. Hãy cài đặt lại đoạn mã sau về dạng factory

<pre>public interface IShape { void draw(); } public class SolidCircle implements IShape { public void draw(){ ... } } public class SolidSquares implements IShape { public void draw(){ ... } } public class SolidRectangle implements IShape { public void draw(){ ... } } public class Circle implements IShape { public void draw(){ ... } } public class Squares implements IShape { public void draw(){ ... } } public class Rectangle implements IShape { public void draw(){ ... } }</pre>	<pre>public class DrawShape { public void draw(int shape, int style) { IShape shape; if(style == 2) { if(shape == 1) shape = new SolidCircle(); else if(shape == 2) shape = new SolidSquares(); else shape = new SolidRectangle(); } else { if(shape == 1) shape = new Circle(); else if(shape == 2) shape = new Squares(); else shape = new Rectangle(); } shape.draw(); } }</pre>
--	---

Bài 2. Hãy cài đặt lại đoạn mã sau dùng pattern phù hợp

<pre>public class SortedList { private ArrayList<int> list = new ArrayList<int>(); public void add(int number) { list.add(number); } public void quickSort() { //sort the list increase } public void insertionSort() { //sort the list decrease } public void selectionSort() { //sort the list decrease } }</pre>	<pre>public class SortApp { public static void main(String[] args) { SortedList test = new SortedList(); test.add(1); test.add(5); test.add(3); test.add(9); test.add(4); test.quickSort(); test.printList(); test.insertionSort(); test.printList(); test.selectionSort(); test.printList(); } }</pre>
---	---

Bài 3. Viết ứng dụng windows form dùng chuột để vẽ hình trong đó áp dụng kết quả của bài 1.

Bài 4. Cho biết việc vẽ hình có thể có nhiều dạng: nét đứt, nét liền, tô kín hay không. Hãy cài lại bài 3 bổ sung pattern phù hợp cho việc vẽ hình ra màn hình.