

실습 1:

-실습코드

```
public class Crayon{
    private String CrayonColor;
    private int CrayonSize;

    public Crayon() {
        setColor("white");
        setSize(1);
    }

    public Crayon(int size, String color) {
        setColor(color);
        setSize(size);
    }

    public void setColor(String color) {
        CrayonColor = color;
    }

    public void setSize(int thickness) {
        CrayonSize = thickness;
    }

    public void draw() {
        System.out.println("Crayon drawing with a " + CrayonColor + " color and " + CrayonSize +
            " size.");
    }
}
```

```
public class Brush {

    private String brushColor;
    private int brushSize;

    public Brush(){
        setColor("white");
        setSize(1);
    }

    public Brush(int size, String color){
        setColor(color);
        setSize(size);
    }

    public void setColor(String color) {
        brushColor = color;
    }

    public void setSize(int thickness) {
        brushSize = thickness;
    }

    public void draw() {
        System.out.println("brush drawing with a " + brushColor+" color and "+brushSize+" size.");
    }
}
```

```

public class Pencil {

    private String pencilColor;
    private int pencilSize;

    public Pencil(){
        setColor("white");
        setSize(1);
    }

    public Pencil(int size, String color){
        setColor(color);
        setSize(size);
    }

    public void setColor(String color) {
        pencilColor = color;
    }

    public void setSize(int thickness) {
        pencilSize = thickness;
    }

    public void draw() {
        System.out.println("pencil drawing with a |" + pencilColor+" color and "+pencilSize+" size.");
    }
}

```

```

1 package Drawing;
2 import java.util.ArrayList;
3
4 public class DrawingFrame {
5
6     public void drawingBrush(Brush b) {
7         b.draw();
8     }
9
10    public void drawingPencil(Pencil p) {
11        p.draw();
12    }
13
14    public void drawingCrayon(Crayon c) {
15        c.draw();
16    }
17
18    public static void main(String[] args) {
19        // TODO Auto-generated method stub
20
21        DrawingFrame f = new DrawingFrame();
22
23        Brush b1 = new Brush();
24        Brush b2 = new Brush(3, "red");
25

```

```
Pencil p1 = new Pencil();
Pencil p2 = new Pencil(5, "black");

Crayon c1 = new Crayon();
Crayon c2 = new Crayon(7, "yellow");

ArrayList<Brush> brushList = new ArrayList<>();
brushList.add(b1);
brushList.add(b2);

ArrayList<Pencil> pencilList = new ArrayList<>();
pencilList.add(p1);
pencilList.add(p2);

ArrayList<Crayon> crayonList = new ArrayList<>();
crayonList.add(c1);
crayonList.add(c2);
```

```
    for (int i = 0; i < brushList.size(); i++) {
        f.drawingBrush(brushList.get(i));
    }

    for (int i = 0; i < pencilList.size(); i++) {
        f.drawingPencil(pencilList.get(i));
    }

    for (int i = 0; i < crayonList.size(); i++) {
        f.drawingCrayon(crayonList.get(i));
    }
}
```

-결과출력

```
Problems Servers Terminal Data Source Explorer Properties Console × History
<terminated> DrawingFrame [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (2024. 11. 17. 오후 7:43:05 -
brush drawing with a white color and 1 size.
brush drawing with a red color and 3 size.
pencil drawing with a white color and 1 size.
pencil drawing with a black color and 5 size.
Crayon drawing with a white color and 1 size.
Crayon drawing with a yellow color and 7 size.
```

-코드관련 설명

#Pencil , brush 클래스와 같은 형태로 Crayon 클래스를 만듭니다.

#DrawingFrame 클래스 안에 drawingCrayon 메소드를 작성하여 Crayon 객체 c 를 받아 draw(); 메소드가 작동하게 합니다.

Crayon 객체 c1,c2을 만듭니다. C1은 기본 생성자를 호출하고, c2는 7,"yellow"를 파라미터로 합니다.

#ArrayList<Crayon> crayonList를 만든 후 , crayonList에 Crayon 객체 c1,c2를 추가합니다.

#이후 for 문을 통해 DrawingFrame 객체 f의 메소드 drawingCrayon을 실행합니다.

실습 2:

1) 실습1번 코드의 문제점:

실습1번 코드에는 Brush와 Pencil, Crayon이라는 별도의 리스트가 있어 drawingBrush()와 drawingPencil() drawingCrayon() 메서드에서 각각 호출되고 있습니다. 이처럼 도구의 종류마다 별도의 메서드와 리스트를 사용하는 것은 다형성을 충분히 활용하지 않는 비효율적인 구조입니다.

또한 Open-Closed Principle에 위배됩니다.

현재 구조에서는 새로운 도구 타입(예: Crayon 클래스)이 추가될 때마다 DrawingFrame에 새로운 메서드나 조건을 추가할 가능성이 큼니다. 이는 코드 수정이 필요하다는 의미로, OCP에 위배됩니다

마지막으로 Dependency Inversion Principle에 위배됩니다.

DrawingFrame 클래스는 특정 구현(Brush, Pencil)에 직접 의존하고 있습니다. DIP에 따르면, DrawingFrame은 Tool이라는 추상화에 의존해야 하며, 구체적인 구현에는 의존하지 않아야 합니다.

-실습 코드

```
1 package Drawing;
2
3 abstract class Tool {
4     abstract void draw();
5
6 }
7
```

```

public class Crayon extends Tool{
    private String CrayonColor;
    private int CrayonSize;

    public Crayon() {
        setColor("white");
        setSize(1);
    }

    public Crayon(int size, String color) {
        setColor(color);
        setSize(size);
    }

    public void setColor(String color) {
        CrayonColor = color;
    }

    public void setSize(int thickness) {
        CrayonSize = thickness;
    }

    public void draw() {
        System.out.println("Crayon drawing with a " + CrayonColor + " color and " + CrayonSize +
            " size.");
    }
}

```

```

1 package Drawing;
2
3 public class Brush extends Tool {
4     private String brushColor;
5     private int brushSize;
6
7     public Brush() {
8         setColor("white");
9         setSize(1);
10    }
11
12    public Brush(int size, String color) {
13        setColor(color);
14        setSize(size);
15    }
16
17    public void setColor(String color) {
18        brushColor = color;
19    }
20
21    public void setSize(int thickness) {
22        brushSize = thickness;
23    }
24
25    public void draw() {
26        System.out.println("brush drawing with a " + brushColor + " color and " + brushSize + " size.")
27    }
28 }

```

```

package Drawing;

public class Pencil extends Tool{
    private String pencilColor;
    private int pencilSize;

    public Pencil() {
        setColor("white");
        setSize(1);
    }

    public Pencil(int size, String color) {
        setColor(color);
        setSize(size);
    }

    public void setColor(String color) {
        pencilColor = color;
    }

    public void setSize(int thickness) {
        pencilSize = thickness;
    }

    public void draw() {
        System.out.println("pencil drawing with a " + pencilColor + " color and " + pencilSize + " size
    }
}

```

```

1 package Drawing;
2 import java.util.ArrayList;
3
4 public class DrawingFrame {
5
6
7     public void drawing(Tool tool) {
8         tool.draw();
9     }
10
11
12
13     public static void main(String[] args) {
14         // TODO Auto-generated method stub
15
16         DrawingFrame f = new DrawingFrame();
17
18         Brush b1 = new Brush();
19         Brush b2 = new Brush(3, "red");
20
21         Pencil p1 = new Pencil();
22         Pencil p2 = new Pencil(5, "black");
23
24         Crayon c1 = new Crayon();
25         Crayon c2 = new Crayon(7, "yellow");
26

```

```

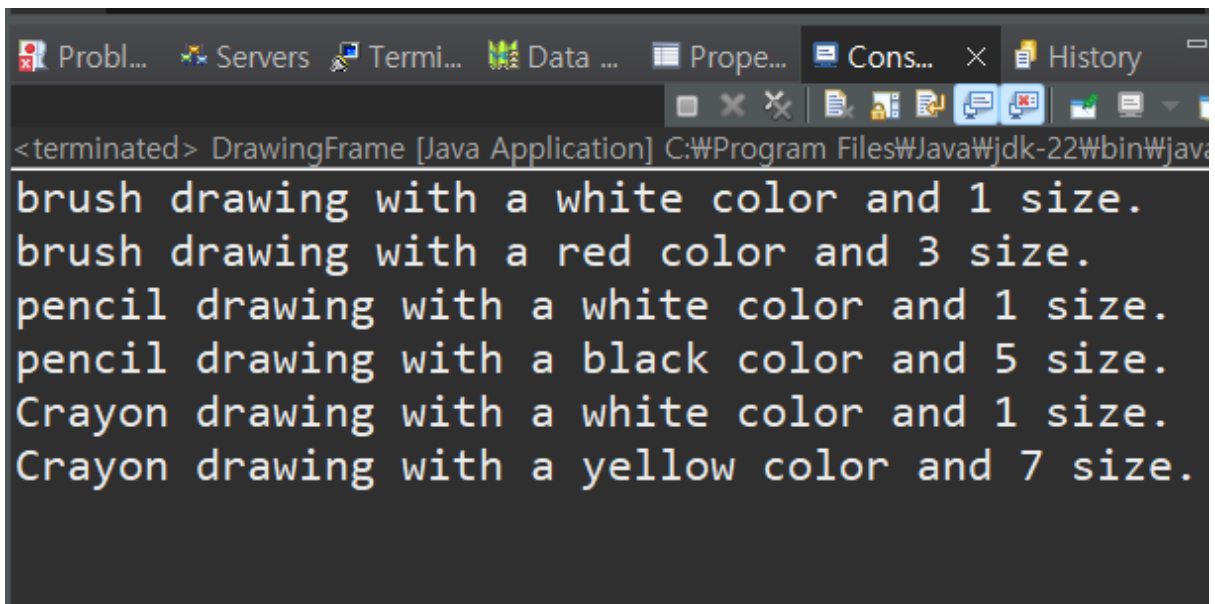
ArrayList<Tool> toolList = new ArrayList<>();
toolList.add(b1);
toolList.add(b2);
toolList.add(p1);
toolList.add(p2);
toolList.add(c1);
toolList.add(c2);

for (Tool tool : toolList) {
    f.drawing(tool);
}

}

```

-결과 출력



```

<terminated> DrawingFrame [Java Application] C:\Program Files\Java\jdk-22\bin\java
brush drawing with a white color and 1 size.
brush drawing with a red color and 3 size.
pencil drawing with a white color and 1 size.
pencil drawing with a black color and 5 size.
Crayon drawing with a white color and 1 size.
Crayon drawing with a yellow color and 7 size.

```

-코드관련 설명

#추상클래스 Tool을 만듭니다. 추상메소드 draw() 를 만들고,

#Tool 클래스를 pencil, brush, Crayon 클래스에 상속시킵니다. 따라서 추상클래스 Tool의 추상메소드 draw()를 , pencil, brush, Crayon 클래스에서 다 구현할 수 있습니다.

이후 DrawingFrame 클래스 안에

```

public void drawing(Tool tool) {

    tool.draw();    }

```

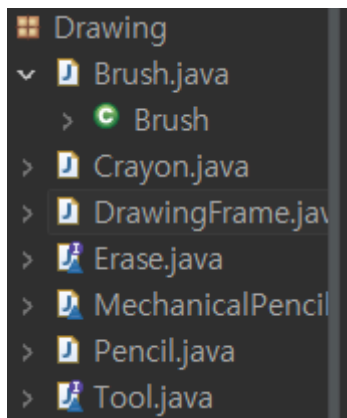

메소드를 만들고, 다형성을 구현합니다.

#`ArrayList<Tool>` toolList를 만들고 그곳에 객체 b1,b2,p1,p2,c1,c2를 저장합니다.

#for 문을 통해 toolist 속의 모든 객체들을 tool 객체로 업캐스팅 하고, (ex, Tool tool=Pencil p1) 이 때의 tool을 파라미터로 객체 f의 메소드 drawing을 작동시킵니다. 그러면 각각의 tool 객체마다 draw() 메소드가 발동되는데, 이때 메소드 오버라이딩을 통해 tool 객체의 하위 클래스인 Brush, Pencil, Crayon의 draw()메소드가 발동됩니다.

실습 3:

-실습 코드:



```
1 package Drawing;
2
3 interface Tool {
4     void draw();
5
6 }
7
```

```
package Drawing;

public class Pencil implements Tool{
    private String pencilColor;
    private int pencilSize;

    public Pencil() {
        setColor("white");
        setSize(1);
    }

    public Pencil(int size, String color) {
        setColor(color);
        setSize(size);
    }

    public void setColor(String color) {
        pencilColor = color;
    }

    public void setSize(int thickness) {
        pencilSize = thickness;
    }

    public void draw() {
        System.out.println("pencil drawing with a " + pencilColor + " color and " +
        pencilSize + " size.");
    }
}
```

```

1 package Drawing;
2
3 public class Brush implements Tool {
4     private String brushColor;
5     private int brushSize;
6
7     public Brush() {
8         setColor("white");
9         setSize(1);
10    }
11
12    public Brush(int size, String color) {
13        setColor(color);
14        setSize(size);
15    }
16
17    public void setColor(String color) {
18        brushColor = color;
19    }
20
21    public void setSize(int thickness) {
22        brushSize = thickness;
23    }
24    public void draw() {
25        System.out.println("brush drawing with a " + brushColor
26            + " color and " + brushSize + " size.");
27    }
28 }

```

```

1 package Drawing;
2
3 public class Crayon implements Tool{
4     private String CrayonColor;
5     private int CrayonSize;
6
7     public Crayon() {
8         setColor("white");
9         setSize(1);
10    }
11
12    public Crayon(int size, String color) {
13        setColor(color);
14        setSize(size);
15    }
16    public void setColor(String color) {
17        CrayonColor = color;
18    }
19
20    public void setSize(int thickness) {
21        CrayonSize = thickness;
22    }
23
24    public void draw() {
25        System.out.println("Crayon drawing with a " + CrayonColor + " color and " + CrayonSize +
26            " size.");
27    }
28 }

```

```

1 package Drawing;
2
3 interface Erase {
4     void erase();
5
6 }
7

```

```

1 package Drawing;
2
3 class MechanicalPencil implements Erase, Tool{
4     private String MechanicalPencilcolor;
5     private int MechanicalPencilsize;
6
7     public MechanicalPencil() {
8         setColor("white");
9         setSize(1);
10    }
11    public MechanicalPencil(int size, String color) {
12        setColor(color);
13        setSize(size);
14    }
15    public void setColor(String color) {
16        MechanicalPencilcolor = color;
17    }
18    public void setSize(int thickness) {
19        MechanicalPencilsize = thickness;
20    }
21    public void draw() {
22        System.out.println("mechanicalPencil drawing with a " +
23        MechanicalPencilcolor + " color and " + MechanicalPencilsize + " size.");
24    }
25    public void erase() {
26        System.out.println("mechanicalPencil erasing.");
27    }
28 }

```

```

1 package Drawing;
2 import java.util.ArrayList;
3
4 public class DrawingFrame {
5
6
7
8     public void drawing(Tool tool) {
9         tool.draw();
10    }
11
12    public static void main(String[] args) {
13        // TODO Auto-generated method stub
14
15        DrawingFrame f = new DrawingFrame();
16
17        Brush b1 = new Brush();
18        Brush b2 = new Brush(3, "red");
19
20        Pencil p1 = new Pencil();
21        Pencil p2 = new Pencil(5, "black");
22
23        Crayon c1 = new Crayon();
24        Crayon c2 = new Crayon(7, "yellow");
25
26        MechanicalPencil m1 = new MechanicalPencil();
27        MechanicalPencil m2 = new MechanicalPencil(9, "gray");
28
29

```

```

        ArrayList<Tool> toolList = new ArrayList<>();
        toolList.add(b1);
        toolList.add(b2);
        toolList.add(p1);
        toolList.add(p2);
        toolList.add(c1);
        toolList.add(c2);
        toolList.add(m1);
        toolList.add(m2);

        for (Tool tool : toolList) {
            f.drawing(tool);
        }
    }
}

```

-결과 출력:

```
Problems Servers Terminal Data Source Explorer Properties Console X History
<terminated> DrawingFrame [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (2024. 11. 17. 오후 9:41:28 - 오후 9:41:29) [pic
brush drawing with a white color and 1 size.
brush drawing with a red color and 3 size.
pencil drawing with a white color and 1 size.
pencil drawing with a black color and 5 size.
Crayon drawing with a white color and 1 size.
Crayon drawing with a yellow color and 7 size.
mechanicalPencil drawing with a white color and 1 size.
mechanicalPencil drawing with a gray color and 9 size.
```

-코드 설명:

#Tool을 인터페이스로 정의하여 draw 메서드를 구현하도록 했고,

#이후 Erase 인터페이스를 새로 정의하고 erase 메서드를 추가했습니다.

#Tool과 Erase를 동시에 구현하는 MechanicalPencil 클래스를 작성하였고

(class MechanicalPencil implements Erase, Tool)

setColor(),setSize() 메소드, 기본생성자, 생성자들의 형태는 crayon, pencil, brush 클래스와 같은형태로 만들었습니다.

또한 이 클래스는 draw와 erase 메서드를 모두 구현합니다.

drawing 메서드를 통해 Tool 인터페이스를 사용하는 객체를 그릴 수 있고,

MechanicalPencil m1,MechanicalPencil m2 객체를 생성합니다. 이후 ArrayList<Tool> toolList에 toolList.add(m1); toolList.add(m2); 메소드를 통해 m1,m2 객체를 넣은 후

For 문을 통해 각 객체별로 drawing 메소드를 적용해 메소드 오버라이딩을 통해 draw() 메소드를 적용합니다.