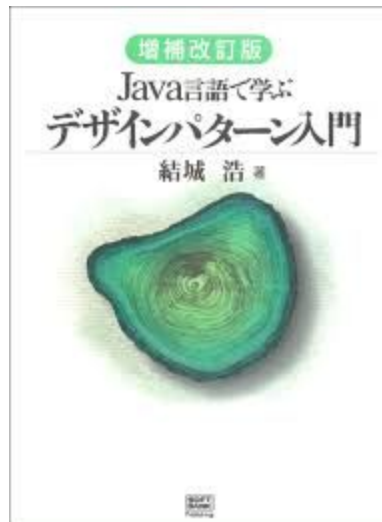


Java言語で学ぶデザインパターン入門

Prototype コピーしてインスタンスを作る



Prototypeパターン

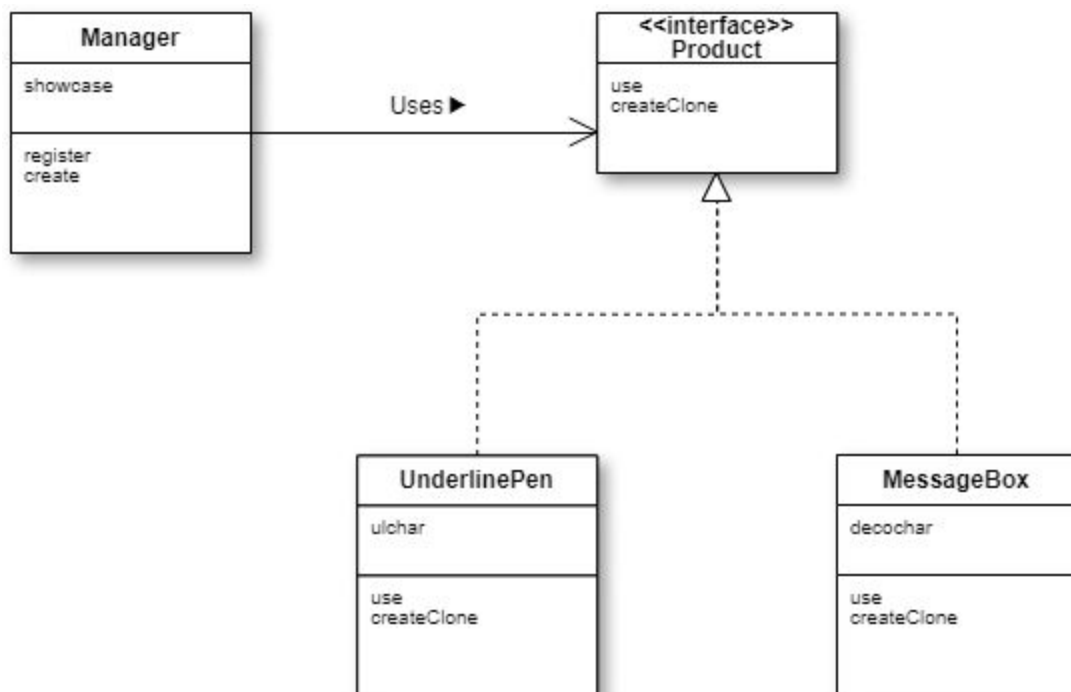
クラスからインスタンスを生成するのではなく、インスタンスから別のインスタンスを作り出す。

サンプルプログラム

パッケージ	名前	解説
framework	Product	抽象メソッドuseとcreateCloneが宣言されているinterface
framework	Manager	createCloneを使ってインスタンスを複製するクラス
無名	MessageBox	文字列を枠線で囲って表示するクラス。useとcreateCloneを実装している。

無名	UnderlinePen	文字列に下線を引いて表示するクラス。useとcreateCloneを実装している。
無名	Main	動作テスト用のクラス

クラス図



Productインタフェース

```

package framework;

public interface Product extends Cloneable{
    public abstract void use(String s);
    public abstract Product createClone();
}
  
```

Managerクラス

```
package framework;

import java.util.HashMap;

public class Manager {
    private HashMap showcase = new HashMap();
    public void register(String name, Product product) {
        showcase.put(name, product);
    }
    public Product create(String protoname) {
        Product product = (Product)showcase.get(protoname);
        return product.createClone();
    }
}
```

Managerクラス

MessageBoxクラス

```
import framework.Product;

public class MessageBox implements Product{
    private char decochar;
    public MessageBox(char decochar) {
        this.decochar = decochar;
    }
    public void use(String s) {
        int length = s.getBytes().length;
        for(int i=0; i<length + 4; i++) {
            System.out.print(decochar);
        }
        System.out.println("");
        System.out.println(decochar + " " + s + " " + decochar);
        for(int i=0; i<length + 4; i++) {
            System.out.print(decochar);
        }
    }
}
```

```

    }
    System.out.println("");
}
public Product createClone() {
    Product product = null;
    try {
        product = (Product)clone();
    } catch (CloneNotSupportedException e) {
        e.printStackTrace();
    }
    return product;
}
}

```

MessageBoxクラス

cloneメソッドは自分自身の複製を作成するメソッド。複製を作成するとき、インスタンスが持っているフィールドの値もそのまま新しいインスタンスにコピーされる。cloneメソッドで複製を作成できるのは、java.lang.Cloneableインタフェースを実装しているクラスのみ。

UnderlinePenクラス

```

import framework.Product;

public class UnderlinePen implements Product{
    private char ulchar;
    public UnderlinePen(char ulchar) {
        this.ulchar = ulchar;
    }
    public void use(String s) {
        int length = s.getBytes().length;
        System.out.println("\n" + s + "\n");
        System.out.print(" ");
        for(int i=0; i<length; i++) {
            System.out.print(ulchar);
        }
        System.out.println("");
    }
}

```

```

    }
    public Product createClone() {
        Product product = null;
        try {
            product = (Product)clone();
        } catch (CloneNotSupportedException e) {
            e.printStackTrace();
        }
        return product;
    }
}

```

UnderlinePenクラス

Mainクラス

```

import framework.Manager;
import framework.Product;

public class Main {
    public static void main(String[] args) {
        Manager manager = new Manager();
        UnderlinePen underlinePen = new UnderlinePen('~');
        MessageBox messageBox = new MessageBox('*');
        MessageBox messageBox2 = new MessageBox('/');
        manager.register("strong message", underlinePen);
        manager.register("warning box", messageBox);
        manager.register("slash box", messageBox2);

        // 生成
        Product p1 = manager.create("strong message");
        p1.use("Hello, world.");
        Product p2 = manager.create("warning box");
        p2.use("Hello world.");
        Product p3 = manager.create("slash box");
        p3.use("Hello world.");
    }
}

```
