

# 03 原型模式、模板方法模式

C-3 创建: 张林伟, 最后修改: 张林伟 2018-10-08 11:34

## 原型模式

### 原型模式

用原型实例制定创建对象的种类，并且通过拷贝这些原型创建新的对象。

### 浅拷贝和深拷贝

浅拷贝，被复制对象的所有变量都含有与原来的对象相同的值，而所有的对其他对象的引用都仍然指向原来的对象。

深拷贝，把引用对象的变量指向复制过的新对象，而不是原有的被引用的对象。

### 模板方法模式

定义一个操作中的算法的骨架，而将一些步骤延迟到子类中。模板方法使得子类可以不改变一个算法的结构即可重定义该算法的某些特定步骤。

### Demo

#### 抽象类

^ 代码块

Java

```
1 public abstract class Game {
2
3     protected abstract void init();
4
5     protected abstract void startPlay();
6
7     protected abstract void playing();
8
9     protected abstract void endPlay();
10
11     public final void play() {
12         init();
13         startPlay();
14         playing();
15         endPlay();
16     }
17 }
```

#### 子类

^ 代码块

Java

```
1 public class Football extends Game {
2     @Override protected void init() {
3         System.out.println("Football init");
4     }
5
6     @Override protected void startPlay() {
7         System.out.println("Football start...");
8     }
9
10    @Override protected void playing() {
11        System.out.println("playing...");
12    }
13
14    @Override protected void endPlay() {
15        System.out.println("Football end...");
16    }
17 }
```

^ 代码块Java

```
1 public class Client {
2     public static void main(String[] args) {
3         Football football = new Football();
4         football.play();
5     }
6 }
```