

- 1 原型模式核心
- 2 原型代码
- 3 和工厂模式、单例模式结合使用
- 参考

## 1 原型模式核心

- 通过一个原型对象，快速的创建一个新对象，并且原型对象和新对象之间互不影响
- 客户端完全不需要知道这个对象是怎么创建的，核心就是快速、互不影响
- 可以应用于经常创建副本的场景

## 2 原型代码

- 原型模式相关类必须实现 Cloneable 接口，并且重新 clone() 方法
- 对于复杂的对象 clone() 方法也比较复杂
- 可以使用序列化来克隆对象，但是相关类必须实现 Serializable 接口

```
public class Resume implements Cloneable {
    private String name;
    private String sex;
    private String age;
    private List<WorkExperience> wks;
    @Override
    protected Object clone() throws CloneNotSupportedException {
        Resume o = (Resume) super.clone();
        o.wks = new ArrayList<>(wks.size()); // 这一步目的是改变新对象wks的地址
        for (int i = 0, iMax = wks.size(); i < iMax; i++) {
            o.wks.add((WorkExperience) wks.get(i).clone());
        }
        return o;
    }
}

public class WorkExperience implements Cloneable, Serializable {
    private String timeArea;
    private String company;
    @Override
    protected Object clone() throws CloneNotSupportedException {
        return super.clone();
    }
}
```

## 3 和工厂模式、单例模式结合使用

- 通过和工厂模式结合使用可以方便的创建不同的原型工厂，使得客户端使用更方便

```
public abstract class ResumePrototypeFactory {
```

```
        public abstract Resume createResume() throws CloneNotSupportedException;
    }

    public class StudentResumeFactory extends ResumePrototypeFactory {
        private static final List<WorkExperience> wks = Arrays.asList(new
        WorkExperience("1-2","nanjing"),new WorkExperience("2-3","beijing"));
        public static final Resume prototype = new Resume("wk","男","22",wks);
        @Override
        public Resume createResume() throws CloneNotSupportedException {
            return (Resume) prototype.clone();
        }
    }

    @Test
    public void test() throws CloneNotSupportedException {
        Resume resume = new StudentResumeFactory().createResume();
        resume.getWks().get(0).setTimeAArea("111-222");
        System.out.println(resume);
        System.out.println(StudentResumeFactory.prototype);
    }
}
```

## 参考

大话设计模式 Head First 设计模式 设计模式 [Java 浅/深克隆、克隆数组 github 源码地址](#)