

实验一.txt  
实验一：模拟SpringIoC功能

```
PreSpringDemo
|
|---src
|
|---tarena
```

步骤一，创建IDeviceWriter接口

```
public interface IDeviceWriter {
    public void saveToDevice();
}
```

-----

步骤二，创建SoftWriter实现类

```
public class SoftWriter implements IDeviceWriter {
    public void saveToDevice() {
        System.out.println("存储到软盘...");
    }
}
```

-----

步骤三，创建UsbWriter实现类

```
public class UsbWriter implements IDeviceWriter {
    public void saveToDevice() {
        System.out.println("存储到U盘...");
    }
}
```

-----

步骤四，创建Business类

```
public class Business {
    private IDeviceWriter deviceWriter;
    public void setDeviceWriter(IDeviceWriter deviceWriter) {
        this.deviceWriter = deviceWriter;
    }
    public void save() {
```

### 实验一.txt

```
deviceWriter.saveToDevice();
```

```
}
```

```
}
```

#### 步骤五，创建属性配置文件

```
business.class=tarena.Business
```

```
deviceWriter.class=tarena.UsbWriter
```

```
//deviceWriter.class=tarena.SoftWriter
```

#### 步骤六，创建BusinessFactory类

```
public class BusinessFactory {
```

```
    private static BusinessFactory factory;
```

```
    private Business business;
```

```
    private IDeviceWriter deviceWriter;
```

```
    private Properties props = new Properties();
```

```
    private BusinessFactory() throws Exception{
```

```
        props.load(new
```

```
FileInputStream("src/tarena/config.properties"));
```

```
        String businessClass =
```

```
props.getProperty("business.class");
```

```
        String deviceWriterClass =
```

```
props.getProperty("deviceWriter.class");
```

```
        business =
```

```
(Business)Class.forName(businessClass).newInstance();
```

```
        deviceWriter = (IDeviceWriter)Class.forName
```

```
(deviceWriterClass).newInstance();
```

```
// 创建关系
```

```
business.setDeviceWriter(deviceWriter);
```

```
}
```

```
    public static BusinessFactory getInstance() throws
```

```
Exception{
```

```
        if (factory == null) {
```

```
            factory = new BusinessFactory();
```

```
        }
```

```
        return factory;
```

```
}
```

## 实验一.txt

```
        public Business getBusiness() {  
            return business;  
        }  
    }  
}
```

---

步骤七，创建测试类

```
public class Test {  
  
    public static void main(String[] args) throws Exception{  
        Business business =  
BusinessFactory.getInstance().getBusiness();  
        business.save();  
    }  
}
```

---

步骤八，结果

存储到U盘...

或

存储到软盘...