模拟IOC容器

题目要求

参考`spring-framework`实现IOC容器，加载并解析配置文件配置，完成对象实例化，实现依赖注入。

具体说明

实现`com.cmsz.upay.ioc.context.impl.ClassPathXmlApplicationContext`中的几个方法：

```
public class ClassPathXmlApplicationContext implements ApplicationContext {

    public ClassPathXmlApplicationContext(String configLocation) {}

    @Override
    public Object getBean(String name) throws BeansException {
        return null;
    }

    @Override
    public <T> T getBean(String name, Class<T> requiredType) throws BeansException {
        return null;
    }

    @Override
    public <T> T getBean(Class<T> requiredType) throws BeansException {
        return null;
    }
}
```

功能点概述

1. 配置文件加载、解析。
2. Bean实例化。
3. 属性注入/构造器注入。
4. 依赖、循环依赖。
5. 懒加载。
6. Bean实例的作用域。

功能点解析

配置文件加载

1. 根据`com.cmsz.upay.ioc.context.impl.ClassPathXmlApplicationContext`构造函数中的参数`configLocation`读取`classpath`中的对应文件。

配置文件解析

1. 根据配置文件的配置，完成实例对象的创建工作。

2. 支持元素`beans`，`beans`的子元素为`bean`，每个`bean`元素代表一个实例对象。
3. `bean`元素支持属性`id`、`class`、`scope`、`lazy-init`，支持子元素`property`、`constructor-arg`：

   属性

   - id：实例对象的id，要求唯一。
   - class：需要实例化对象的全类名。
   - scope：实例对象的作用域，支持`singleton`(默认)、`prototype`，`singleton`表示整个容器只维护一个实例对象，`prototype`表示每次应用程序需要获取实例对象时都新生成一个实例对象。
   - lazy-init：懒加载，容器启动时不实例化该对象，在应用程序需要时才加载。
4. `property`元素用于属性注入，即实例化出新对象后，调用其相应的`setter`方法，支持属性`name`、`value`、`ref`：

   属性

   - name：属性名称。
   - value：属性值。
   - ref：属性值引用另一个实例对象。
5. `constructor-arg`元素用于构造器注入，即在实例化对象时调用有参构造函数，支持属性`name`、`value`、`ref`。

   属性

   - name：属性名称。
   - value：属性值。
   - ref：属性值引用另一个实例对象。

Bean实例化

1. 对每个`bean`元素解析并创建实例对象

属性注入/构造器注入

1. 对`bean`元素中子元素`property`的配置使用属性注入。
2. 对`bean`元素中子元素`contructor-arg`的配置使用构造器注入。

依赖

1. 对于`property`和`contructor-arg`中属性为`ref`的，如果引用对象已创建，注入到实例中，如果引用对象未创建，先创建引用对象，再注入到实例中。
2. 循环依赖：A引用B，B引用A。只有属性依赖可以正常执行(即先创建出A和B的实例，在分别`setter`)，对于构造器注入，显然，无法实现，需抛出异常`com.cmsz.upay.ioc.beans.exception.BeansException`。

懒加载

1. 容器启动时不实例化该对象，在应用程序需要时才加载。

> 应用程序需要时：分几种情况，一种是应用程序主动调用了getBean方法，另一种是，非懒加载的bean(不妨叫A)引用了懒加载的bean(不妨叫B)，容器启动时，因为需要实例化A，所以这时B就是应用程序需要的，自然就要实例化了。

Bean实例的作用域

1. 见`bean`属性`scope`的说明。

测试案例

1. 实现`com.cmsz.upay.ioc.context.impl.ClassPathXmlApplicationContext`后，通过`com.cmsz.upay.ioc.test.TestSuit`所有测试案例。

额外功能点(可不实现)

1. Bean自动刷新：配置文件更改时，刷新容器中的Bean，需要考虑同步问题。
2. 支持注解：`@Bean`功能相当于配置文件的`bean`元素，`@Autowired`功能相当于`ref`属性。

考点

1. xml文件的解析。
2. 反射。
3. java自带容器的使用。
4. maven使用。
5. 设计模式。
6. 注解。