准备先决条件

1.创建一个ModelTest01类

```
public class ModelTest01 {
}
```

2.创建一个ModelTest02类

```
public class ModelTest02 {
}
```

3.先创建一个配置文件bean.properties

```
modelTest01=com.hh.pro.ModelTest01
modelTest02=com.hh.pro.ModelTest02
```

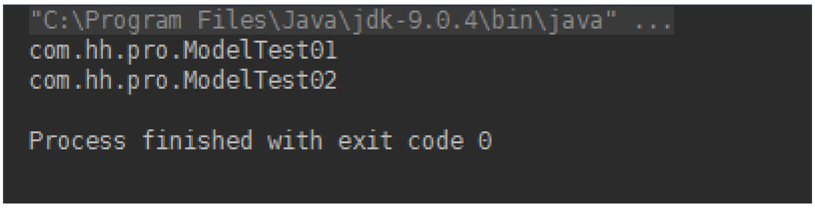
获取配置文件的第一种方法

```
package com.hh.pro;

import org.springframework.beans.factory.BeanFactory;

import java.io.IOException;
import java.io.InputStream;
import java.util.Enumeration;
import java.util.HashMap;
import java.util.Map;
import java.util.Properties;

public class PropertiesFactory {
    //定义一个Properties对象
    private  static Properties props;
    //定义一个Map,用于存放我们创建的对象，我们把它称之为容器
    private static Map<String,Object> beansMap;
    static{
        try{
           //实例化对象
           props = new Properties();
           //获取properties文件的流对象
            InputStream in =
BeanFactory.class.getClassLoader().getResourceAsStream("bean.properties");
            props.load(in);
            beansMap = new HashMap<String,Object>();
            //取出配置围文件中所有的key
            Enumeration keys = props.keys();
            //遍历枚举
            while (keys.hasMoreElements()){
              //取出每个key
                String key = keys.nextElement().toString();
                //根据key获取value
                String beanPath = props.getProperty(key);
                //反射创建对象
                Object value = Class.forName(beanPath).newInstance();
                //将bean.properties里面的所有配置都put到Map中
                beansMap.put(key, value);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static Object getBean(String beanName){
        //直接从Map中获取对应key的value值
       return beansMap.get(beanName);
    }
}
```

测试类

```
public class PropTest {

    public static void main(String[] args) {
        Object modelTest01 = PropertiesFactory.getBean("modelTest01");
        Object modelTest02 = PropertiesFactory.getBean("modelTest02");
        System.out.println(modelTest01);
        System.out.println(modelTest02);
    }
}
```

运行结果：



```
"C:\Program Files\Java\jdk-9.0.4\bin\java" ...
com.hh.pro.ModelTest01
com.hh.pro.ModelTest02

Process finished with exit code 0
```

获取配置文件的第二种方法

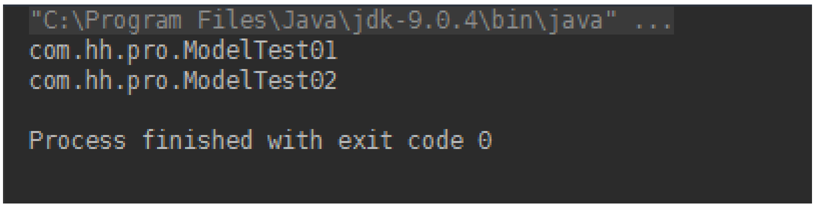
这块加了一个时间判断，如果你的配置文件里面有连接数据库(容易改变),则间隔一段时间就得重新加载

```
package com.hh.pro;
+
import java.util.Date;
import java.util.Properties;

public class PropertiesFactory02 {
    private static Properties pro = new Properties();
    private static Date lastUpdate = new Date(0);//这里获取到 0
    private static long updateInterval = 2*60*1000;//时间间隔分钟
    static{
        try{

pro.load(PropertiesFactory02.class.getClassLoader().getResourceAsStream("bean.properties"
));
        }catch(Exception e){

        }

    }

    public static String getProperty(String key){
        Date date = new Date();
        if(date.getTime()-lastUpdate.getTime()>updateInterval){
            synchronized (lastUpdate){
                date = new Date();
                if(date.getTime()-lastUpdate.getTime()>updateInterval){
                    Properties newPro = new Properties();
                    try{

newPro.load(PropertiesFactory02.class.getClassLoader().getResourceAsStream("bean.properti
es"));
                        lastUpdate.setTime(date.getTime());
                        pro = newPro;
                    }catch(Exception e){

                    }
                }
            }
        }
        return pro.getProperty(key);
    }
}
```

测试类

```
public class PropTest {

    public static void main(String[] args) {

        Object modelTest01 = PropertiesFactory02.getProperty("modelTest01");
        Object modelTest02 = PropertiesFactory02.getProperty("modelTest02");
        System.out.println(modelTest01);
        System.out.println(modelTest02);
    }
}
```

运行结果：



```
"C:\Program Files\Java\jdk-9.0.4\bin\java" ...
com.hh.pro.ModelTest01
com.hh.pro.ModelTest02

Process finished with exit code 0
```