设模模式参考菜鸟教程:<http://www.runoob.com/design-pattern/design-pattern-tutorial.html>

1. 简单工厂模式,主要是用以生产工厂对象.

主要情况是针对一个接口有多个实现类,需要需要调用哪个接口的实现类的时候,使用工厂进行处理,处理后返回相应的对象.

因为一般情况下:实例化对象都是父类的引用指向子类的实现( person p=new student()),但是一般一个接口都有多个实现,具体的需要实现哪个对象,则通过工厂来进行判断.

例子;

有一个抽象类.(或是是接口)

public abstract class Video {

public abstract void produce();

}

该接口有两个实现类:

public class JavaVideo extends Video {

@Override

public void produce() {

System.out.println("录制Java课程视频");

}

}

public class PythonVideo extends Video {

@Override

public void produce() {

System.out.println("录制Python课程视频");

}

}

一般情况下使用对象都是直接new的方式:

即 Video v=new JavaVideo();

但是使用了工厂模式来进行对象的实例化如下:

public class VideoFactory {

//方式一

public Video getVideo(Class c){

Video video = null;

try {

**//这里强制转换为父类的接口类型.调用的时候传入的是子类的类名**

video = (Video) Class.forName(c.getName()).newInstance();

} catch (InstantiationException e) {

e.printStackTrace();

} catch (IllegalAccessException e) {

e.printStackTrace();

} catch (ClassNotFoundException e) {

e.printStackTrace();

}

return video;

}

//方式二 (直接传入字符串进行区分,但是这种方法扩展很麻烦,所以一般都用的上面的方法)

public Video getVideo(String type){

if("java".equalsIgnoreCase(type)){

return new JavaVideo();

}else if("python".equalsIgnoreCase(type)){

return new PythonVideo();

}

return null;

}

}

简单工厂模式在JDK源码中的列子:

下面的Calendar类,是一个抽象类,有两个实现类,在其Canlender的获取子类的实例的时候,使用的是简单工厂方法来获取到子类的实例的,是通过的传入的是字符串,然后通过if...else的判断来实现分支的.

