使用Spring提供的注解@Autowired给引用类型属性赋值

新建bao03包，将Student类复制过来，新建一个School类，在类名上添加@Component，将该类中的属性用@Value进行属性赋值，然后在Student类中添加一个私有是School属性，用@Autowired注解赋值。

Student类如下：

```
package com.studymyself.bao03;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Component;


@Component("student03")
public class Student {

    public Student() {
        System.out.println("Student03无参构造方法执行！");
    }

    @Value("终南山")//value可省略
    private String name;
    @Value(value = "5214")
    private int age;

    /**
     * 引用类型注解赋值
     * @Autowired：Spring提供的注解，实现引用类型的赋值
     * 注解实现引用类型的赋值，使用的是自动注入原理，即方式有byType和byName
     * @Autowired注解默认使用byType方式，因为该方式只需要将该注解放到引用类
     * 型属性上面就行了，不需要添加其他东西
     *
     * 使用方式：
     *       1、放在在引用类型属性定义上面，该方式不使用set方法，推荐使用
     *       2、放在属性的set方法上，调用set方法赋值
     *
     * @Autowired注解中有一个属性required，是boolean类型的，默认值为true
     * 一般我们使用@Autowired注解时，不指定该属性的值时，
     *           默认为true，效果就是在为该引用类型属性赋值失败后，会抛出异常，终止整个程序执
行
     *           当我们设置为false时，即使赋值失败，程序正常执行，结果只是该引用属性的值为
null而已
     *           一般我们设定required值为true，可以提早发现程序异常
     *           特殊情况设置为false
     *
     * @Autowired使用byName方式：
     * 除了在引用类型定义上用@Autowired注解外，还需要添加@Qualifier(value"bean的id")注解
```

```
     *  该方式是Spring底层通过查找由@Qualifier(value"bean的id")提供的名为id的对象，判断是否是
     *  引用类型的数据类型，然后赋值给引用变量
     *
     */
    //使用byType方式的@Autowired
    @Autowired
    private School school;

//    //使用byName方式的@Autowired
//    @Autowired
//    @Qualifier("student")
//    private School school;

    public void setName(String name) {
        this.name = name;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public void setSchool(School school) {
        this.school = school;
    }

    @Override
    public String toString() {
        return "Student{" +
                "name='" + name + '\'' +
                ", age=" + age +
                ", school=" + school +
                '}';
    }
}
```

School类如下：

```
package com.studymyself.bao03;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Component;

@Component("school")
public class School {

    @Value("扫大街大学")
    private String name;
    @Value("美国")
    private String addr;

    public School() {
        System.out.println("School的无参数构造方法执行！");
    }

    public void setName(String name) {
        this.name = name;
```

```
    }

    public void setAddr(String addr) {
        this.addr = addr;
    }

    @Override
    public String toString() {
        return "School{" +
                "name='" + name + '\'' +
                ", addr='" + addr + '\'' +
                '}';
    }
}
```

新建测试文件MyTest03.java

```
package com.studymyself;

import com.studymyself.bao03.Student;
import org.junit.Test;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MyTest03 {
    @Test
    public void testStudent(){

        //配置文件路径
        String config = "applicationContext.xml";

        //获取容器对象
        ApplicationContext ac = new ClassPathXmlApplicationContext(config);

        //获取Student类对象
        Student student = (Student) ac.getBean("student03");
        System.out.println(student);

    }

}
```