

1 Spring 简介

- Spring: 轻量级的开源的 JavaEE 框架
- Spring: 可以解决企业应用开发的复杂性
- Spring两个核心部分: IOC、AOP
 - IOC: 控制反转, 把对象创建过程交给Spring管理
 - AOP: 面向切面, 在不修改源码情况下进行功能增强

Spring特点

- 方便解耦, 简化开发
- AOP 编程支持
- 方便程序测试
- 方便和其他框架进行整合
- 方便进行事务操作
- 降低API开发难度

Spring 前身

- interface21

2 Spring 体系结构

- 1) 核心容器: `spring-beans`、`spring-core`、`spring-context`、`spring-expression`
- 2) AOP 和设备支持: `spring-aop`、`spring-aspects`、`spring-instrument`
- 3) 数据访问集成: `spring-jdbc`、`spring-tx`、`spring-orm`、`spring-jms`、`spring-oxm`
- 4) Web: `spring-web`、`spring-webmvc`、`spring-websocket`、`spring-webflux`
- 5) 报文: `spring-messaging`
- 6) 测试: `spring-test`

3 反射

```
DriverManager.registerDriver(new com.mysql.jdbc.Driver());  
Class.forName ("com.mysql.jdbc.Driver");
```

反射是框架设计的灵魂, 是在程序运行过程中:

- 对任意一个类, 都能够[知道]它的所有属性和方法;
- 对任意一个对象, 都能够[调用]它的所有属性和方法;
- 这种动态获取信息以及动态创建对象的功能称为 `Java` 的反射机制

获取Class类对象的三种方式

1. 对象.getClass() 方法
2. 类.class 属性
3. Class.forName (String className) , 加载指定的类并初始化

Simple:

```
public static void main(String[] args) throws ClassNotFoundException {  
    // new 操作时, 产生一个 Student 对象, 一个 Class 对象, Student 类的唯一一个 Class 对象  
    // 再次 new 时, 产生一个 Student 对象, 不再产生 Class 对象  
    Student stu = new Student();  
    Class cls1 = stu.getClass(); //1  
    Class cls2 = Student.class; //2  
    Class cls3 = Class.forName ("com.qst.bean.Student"); //3  
    // test  
    System.out.println(cls2 == cls1);  
    System.out.println(cls2 == cls3);  
    // create object  
    Student s1 = (Student) cls1. newInstance ();  
    Student s2 = (Student) cls2. newInstance();  
    Student s3 = (Student)cls3. newInstance();  
}
```