

```
package com.sxt.reflection;
```

```
/**
```

```
 * 测试各种类型(class, interface, enum, annotatino, primitive, type, void)的  
java.lang.Class对象的获取方式
```

```
 *
```

```
 * @author 江
```

```
 *
```

```
 */
```

```
@SuppressWarnings("all")
```

```
public class Demo01 {
```

```
    public static void main(String[] args) {
```

```
        String path="com.sxt.bean.User";
```

```
        try {
```

```
            Class clazz=Class.forName(path);
```

//对象是表示或者封装一些数据。一个类被加载后，JVM会创建一个对应的Class对象，类的整个结构信息会被放到对应的Class对象中。

//这个Class对象就像一面镜子一样，通过这面镜子我可以看到对应类的全部信息。

```
        Class strClazz=String.class;
```

```
        Class strClazz2=path.getClass();
```

```
        System.err.println(strClazz==strClazz2); //true
```

```
        Class intClazz=int.class;
```

```
        //跟维度有关
```

```
        int[] arr01=new int[10];
```

```
        int[] arr02=new int[30];
```

```
        int[][] arr03=new int[10][10];
```

```
        double[] arr04=new double[10];
```

```
System.err.println(arr01.getClass().hashCode());
//2018699554
System.err.println(arr02.getClass().hashCode());
//2018699554
System.err.println(arr03.getClass().hashCode());
//1311053135
System.err.println(arr04.getClass().hashCode());
//118352462
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }
}
}
```