



```
import java.util.ArrayList;

public class MethodDemo4 {
    public static void main(String[] args) {
        ArrayList list = new ArrayList();

        ArrayList<String> list1 = new ArrayList();
        list1.add("hello");
        //list2.add(20);错误的
        Class c1 = list.getClass();
        Class c2 = list1.getClass();
        System.out.println(c1 == c2);
    }
}
```

true

反射的操作都是编译之后的操作

`c1 == c2` 结果返回 `true` 说明编译之后集合的泛型是去泛型化的

Java 中集合的泛型，是防止错误输入的，只在编译阶段有效，绕过编译就无效了

```
reflect.txt MethodDemo4.java Problems <terminated> Me
/*
 * c1==c2结果返回true说明编译之 true
 * Java中集合的泛型，是防止错误转 2
 * 绕过编译就无效了
 * 验证：我们可以通过方法的反射来
 */
try {
    Method m = c2.getMethod("
    m.invoke(list1, 20);
    System.out.println(list1.size()
} catch (Exception e) {
    e.printStackTrace();
}
}
```

```

/*
 * c1==c2结果返回true说明编译之后集合的泛型是去泛型化的
 * Java中集合的泛型，是防止错误输入的，只在编译阶段有效
 * 绕过编译就无效了
 * 验证：我们可以通过方法的反射来操作，绕过编译
 */
try {
    Method m = c2.getMethod("add", Object.class);
    m.invoke(list1, 20); // 绕过编译操作就绕过了泛型
    System.out.println(list1.size());
    System.out.println(list1);
} catch (Exception e) {
    e.printStackTrace();
}
}

```

但是foreach不行，因为int转string报的异常

