黑客派

Java 泛型学习

作者：zjhch123

原文链接：https://hacpai.com/article/1478265748884

来源网站：黑客派

许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

Java 泛型学习

在复习Java的时候，发现了对于泛型的了解还不是很好，所以今天特别画时间看了一下泛型方面的东
。

1. 泛型

泛型是JDK1.5版本之后引入的，其目的是为了让编程人员能够使用类型抽象。常见于集合中。
以下就是使用泛型的一个最典型例子:

```
List<Integer> list = new ArrayList<>();
```

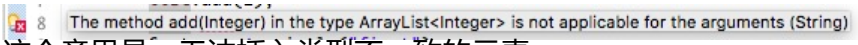
上述代码的意思是，新建一个ArrayList，并且通过泛型限制这个list中只能存放Integer类型的元素。
目的主要是为了防止在对集合操作时，各种类型不一样所导致的java.lang.ClassCastException的问
。

2. 泛型只在编译阶段有效

看以下代码

```
ArrayList<Integer> list = new ArrayList<>();
list.add(1);
list.add("two"); // error
System.out.println(list);
```

在Eclipse中，第3行会有一个错误提示。



这个意思是，无法插入类型不一致的元素。
可是，使用Java的反射机制，可以绕过泛型验证。

```
list.getClass().getMethod("add", Object.class).invoke(list, "two");
System.out.println("list");
```

以上代码的运行结果为:

成功绕过。

3. 泛型类

参考以下代码

```
public class Class1 {

    public static void main(String[] args) {
        Box<String> b = new Box<>("123");
        System.out.println(b.getData());
    }

}


class Box<T> {

    private T data;

    public Box() {

    }

    public Box(T data) {
        this.data = data;
    }

    public T getData() {
        return data;
    }

    public void setData(T obj) {
        data = obj;
    }

}
```

这是使用泛型类的最简单情况。

4. 通配符

查看以下代码

```
public static void main(String[] args) {
    Box<Integer> b = new Box<>(1);
    System.out.println(b.getData());
    getData(b); // error
}

public static void getData(Box<Number> box) {
    System.out.println(box.getData());
}
```

在注释处，Eclipse提示一个错误，大致意思是，类型不匹配，需要Box<Number>类型的参数。

为了满足我们getData方法的需求，我们必须对其参数进行修改。此时，我们使用类型通配符?来进行改操作。

```
public static void getData(Box<? extends Number> box) {
```

```
    System.out.println(box.getData());
}
```

进行这样的修改之后，编译通过。

同理，还有<? supers Number> 这样的情况，这里就不多做赘述了。