



链滴

Java数组与泛型

作者: [Eric](#)

原文链接: <https://ld246.com/article/1447082410371>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

是日快下班的时候，我到同事蕾丝大神座位上跟他唠嗑，他说问你个问题看你道不。哥最近面试如流水，身经百战，自然胸有成竹地表示来来来。蕾丝问我：“为什么java的数组不支持泛型？”。擦，这货不按常理出牌呀。哥还真没有想过这问题，想了一会儿，回答他说：“是因为类型擦除吗？”，他说他也不太清楚，但知道跟类型擦除有关。然后他打开IntelliJ写了以下例子：

```
public class GenericTest {
    private List<String> list1 = new ArrayList<String>();
    private List list2 = new ArrayList();
}
```

然后他用IntelliJ查看编译出来的字节码如下(我只节选重要部分)：

```
private Ljava/util/List; list1

// compiled from: GenericTest.java

// access flags 0x2
// signature Ljava/util/List;Ljava/lang/String;
// declaration: java.util.List;java.lang.String;
private Ljava/util/List; list1

// access flags 0x2
private Ljava/util/List; list2

.....
}
```

他说从字节码可以看出编译之后泛型已经被擦除掉了。他接着再写了以下例子：

```
public class GenericTest {
    /* private List<String> list1 = new ArrayList<String>();
    private List list2 = new ArrayList();*/
```

```
private Integer[] integers = new Integer[]{};
private String[] strings = new String[]{};
```

```
}
```

编译后的字节码如下：

```
public class GenericTest {

// compiled from: GenericTest.java

// access flags 0x2
private [Ljava/lang/Integer; integers

// access flags 0x2
private [Ljava/lang/String; strings

....

}
```

可以看到对于数组而言，它作为一段连续的内存空间，在编译期间就必须知道它要存放的是哪一的数据。最后， 蕾丝大神表示要真正理解这个问题可能要去查阅一下java语言规范和java虚拟机规范文档，他还提到一点就是数组是java一开山就存在的，那时还没有泛型这个概念，泛型是到java1.5后有的，SUN为此还做了不少backward Compatible的工作。

说到这，虽然我还是有点疑问，但是对于蕾丝大神的研究精神甚是佩服。回家之后大概地做了一总结。

说到泛型，当然首当其冲要说到类型擦除(Type Erasure)，所谓的类型擦除，Java官方文档给出说明如下：

Generics were introduced to the Java language to provide tighter type checks at compile time and to support generic programming. To implement generics, the Java compiler applies type erasure to:

1. Replace all type parameters in generic types with their bounds or Object if the type parameters are unbounded. The produced bytecode therefore, contains only ordinary classes, interfaces, and methods.
2. Insert type casts if necessary to preserve type safety.
3. Generate bridge methods to preserve polymorphism in extended generic types.
Type erasure ensures that no new classes are created for parameterized types; consequently, generics incur no runtime overhead.

加黑的第一点很重要，意思是说Java编译器将所有泛型类型参数(type parameters)替换成它们绑定的类型，如果类型参数没有绑定，则替换成Object，这样编译出来的字节码，就会只包含了普通类，接口和方法。这里所说的绑定(bound)是指像

```
public class Node<T extends Comparable<T>>
```

```
这样的，而没有绑定，则指像：
```

```
public class Node<T>
```

关于类型擦除，网上有篇文章([摸我](http://blog.csdn.net/lonelyroamer/article/details/7868820))做了非常详细的说明，当然，[Java官方文档](https://docs.oracle.com/javase/tutorial/java/generics/erasure.html)也是必读的。而关于java数组为什么不支持泛型，我在网上搜了一圈，发现中文类的，只有[这篇文章](http://www.blogjava.net/deepnighttwo/articles/298426.html)总结得较为靠谱。