

JAVA匿名内部类(fromthinkinginjava3th) PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/278/2021_2022_JAVA_E5_8C_BF_E5_90_8D_c104_278568.htm 简单地说：匿名内部类就是没有名字的内部类。什么情况下需要使用匿名内部类？如果满足下面的一些条件，使用匿名内部类是比较合适的：只用到类的一个实例。类在定义后马上用到。类非常小（SUN推荐是在4行代码以下）给类命名并不会导致你的代码更容易被理解。在使用匿名内部类时，要记住以下几个原则：匿名内部类不能有构造方法。匿名内部类不能定义任何静态成员、方法和类。匿名内部类不能是public,protected,private,static。只能创建匿名内部类的一个实例。一个匿名内部类一定是在new的后面，用其隐含实现一个接口或实现一个类。因匿名内部类为局部内部类，所以局部内部类的所有限制都对其生效。下面的例子看起来有点奇怪：//在方法中返回一个匿名内部类

```
public class Parcel6 { public Contents cont() { return new Contents() { private int i = 11. public int value() { return i. } }. // 在这里需要一个分号 } public static void main(String[] args) { Parcel6 p = new Parcel6(). Contents c = p.cont(). } } cont()方法将下面两个动作合并在一起：返回值的生成，与表示这个返回值的类的定义！进一步说，这个类是匿名的，它没有名字。更糟的是，看起来是你正要创建一个Contents对象：return new Contents() 但是，在到达语句结束的分号之前，你却说：“等一等，我想在这里插入一个类的定义”：return new Contents() { private int i = 11. public int value() { return i. } }. 这种奇怪的语法指的是：“创建一个继承自Contents的匿名类的对
```

象。”通过new 表达式返回的引用被自动向上转型为对Contents的引用。匿名内部类的语法是下面例子的简略形式：

```
class MyContents implements Contents { private int i = 11. public int value() { return i. } } return new MyContents().
```

在这个匿名内部类中，使用了缺省的构造器来生成Contents。下面的代码展示的是，如果你的基类需要一个有参数的构造器，应该怎么办：

```
public class Parcel7 { public Wrapping wrap(int x) { // Base constructor call: return new Wrapping(x) { // Pass constructor argument. public int value() { return super.value() * 47. } }. // Semicolon required } public static void main(String[] args) { Parcel7 p = new Parcel7(). Wrapping w = p.wrap(10). } }
```

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com