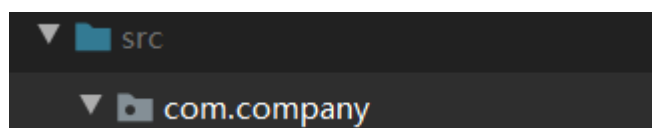


包的声明和导入

包其实就是用来区分类位置的，就是可以将类进行分类，随着程序的不断扩大，可能会创建各种各样的类，它们可能用法不同，如果这些用法不用的类放在一起，就会显得很杂乱无章，所以就可以用到包来将这些类来进行分类

包的命名规则有点类似于域名,可以通过字母和数字来组合，比如有个域名为 `blog.tanc.fun:9999`，后面的9999为端口不需要管，而 `blog.tanc.fun` 就是域名，包名就通常是反着来 `fun.tanc.blog`

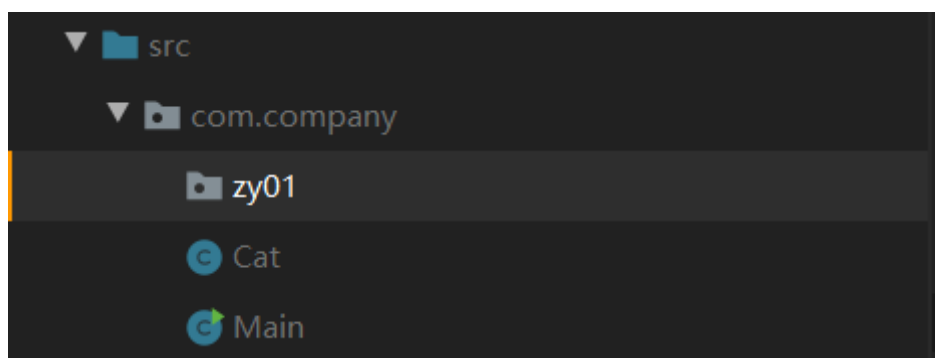
在idea中创建项目的时候就会创建包名可以在里面创建类文件



`package` 关键字是用来指定当前类所处的包

```
1 package com.company; //这里就是指当前类所处的包
2
3 public class Main {
4
5     public static void main(String[] args) {
6         // write your code here
7
8     }
9 }
10
```

当然也可以在包的内部在创建一个包



如果我将Main类放入 `com.company.zy01` 包中(鼠标直接移动即可)，那么它所指向的包就会改变

```

1 package com.company.zy01; //改变了
2
3 public class Main {
4
5     public static void main(String[] args) {
6         // write your code here
7
8     }
9 }
10

```

此时 `Main` 和 `Cat` 都不在同一个包类，现在在 `Main` 中是不能直接调用 `Cat` 的，之前我们可以直接调用是因为我们开始将文件都创建在同一个包内不需要导入即可直接调用，如果当我们需要使用到其他包内的类时就要导入，编译器已经提示了是否要 导入类？



导入内是使用 `import` 关键字来导入，如果包下面有很多类，你不想一个一个导入的话，可以直接使用 `import com.company.*` 来直接导入，这样就可以直接使用 `Cat` 类了

```

1 package com.company.zy01;
2
3
4 import com.company.Cat;
5
6 public class Main {
7
8     public static void main(String[] args) {
9         // write your code here
10        Cat.info = "Cat和Main不在同一个包内";
11    }
12 }
13

```

当然也可以不需要使用 `import` 调入，不过很麻烦，如下列代码所示，需要将包名和类名写全，编写时就会很麻烦

```

1 package com.company.zy01;
2
3
4 public class Main {
5
6     public static void main(String[] args) {
7         // write your code here
8         com.company.Cat.info = "Cat和Main不在同一个包内";
9     }
10 }
11

```

看到这个形式可能会很熟悉，我女朋友直接就和我提到过“Java使用printf需要写很长一段System.out.println()而C语言和C++就只需要print()或者cout就好了”

我的理解: 就如在类和对象中说的静态类和静态变量，它是属于类的而不是对象，System.out中的out就是一个静态类的成员变量，

```
public final static PrintStream out = null;
```

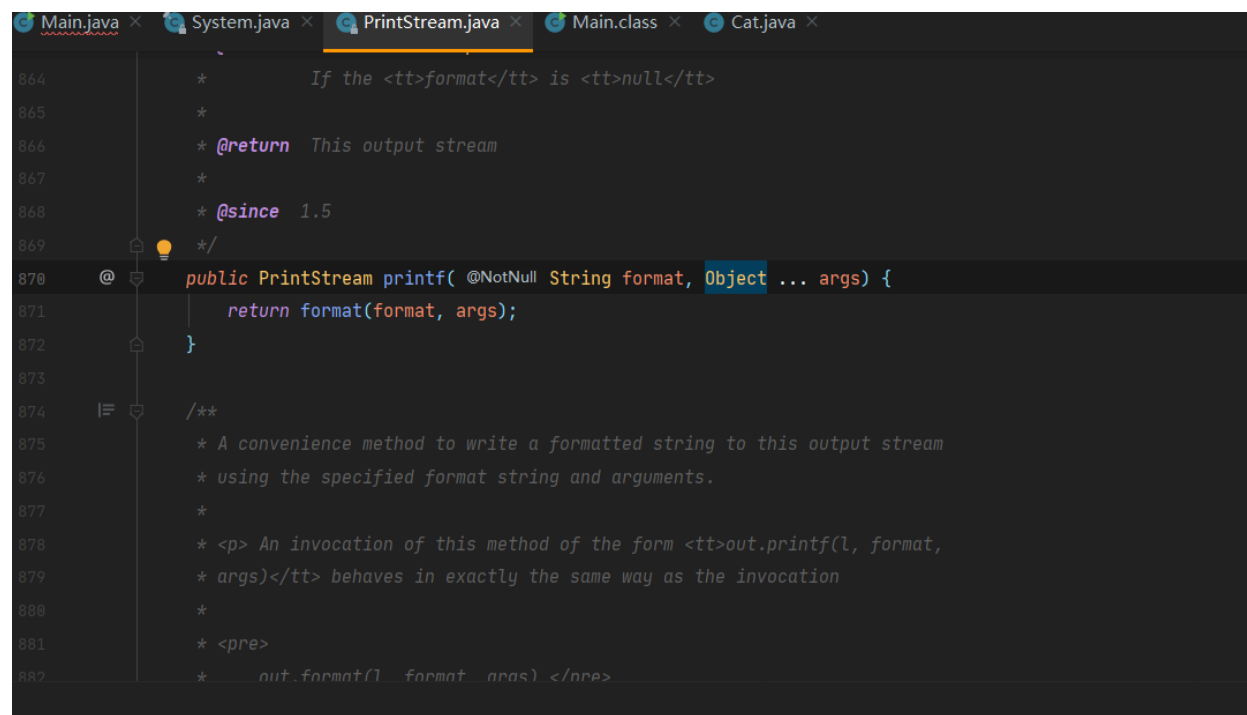
它就是PrintStream类的一个实例(也可以叫类对象吧); 静态变量调用可以不用创建好类对象，通过类对象来调用，可以直接通过类来调用，但是System的构造函数是私有的，不能在类以外的地方调用

```

/** Don't let anyone instantiate this class */
private System() {
}

```

所以说他不能通过创建类对象来调用，而printf是PrintStream类的一个方法，out又是通过PrintStream创建的类对象，所以说我们需要通过System.out.printf来调用



```

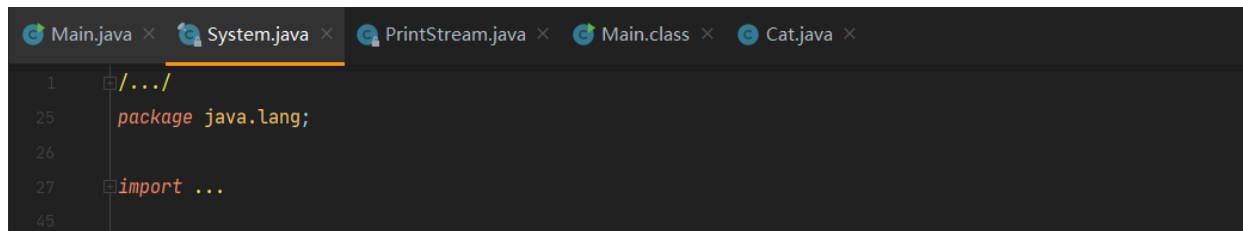
Main.java x System.java x PrintStream.java x Main.class x Cat.java x
864      *      If the <tt>format</tt> is <tt>>null</tt>
865      *
866      * @return This output stream
867      *
868      * @since 1.5
869      */
870  @ public PrintStream printf( @NotNull String format, Object ... args) {
871      return format(format, args);
872  }
873
874  /**
875   * A convenience method to write a formatted string to this output stream
876   * using the specified format string and arguments.
877   *
878   * <p> An invocation of this method of the form <tt>out.printf(l, format,
879   * args)</tt> behaves in exactly the same way as the invocation
880   *
881   * <pre>
882   * out.format(l format args) </pre>

```

好说到这里应该大概懂了一点点为什么需要那么一大串来使用 `printf` 还有个偷懒一点点的小办法，可以使用静态导入，`import static` 来导入静态方法和字段；

```
1 package com.company.zy01;
2
3 import java.lang.System.*;
4
5 import static java.lang.System.out; //导入静态类
6
7
8 public class Main {
9
10     public static void main(String[] args) {
11         // write your code here
12         out.println("Hello world!");
13     }
14 }
15
16
```

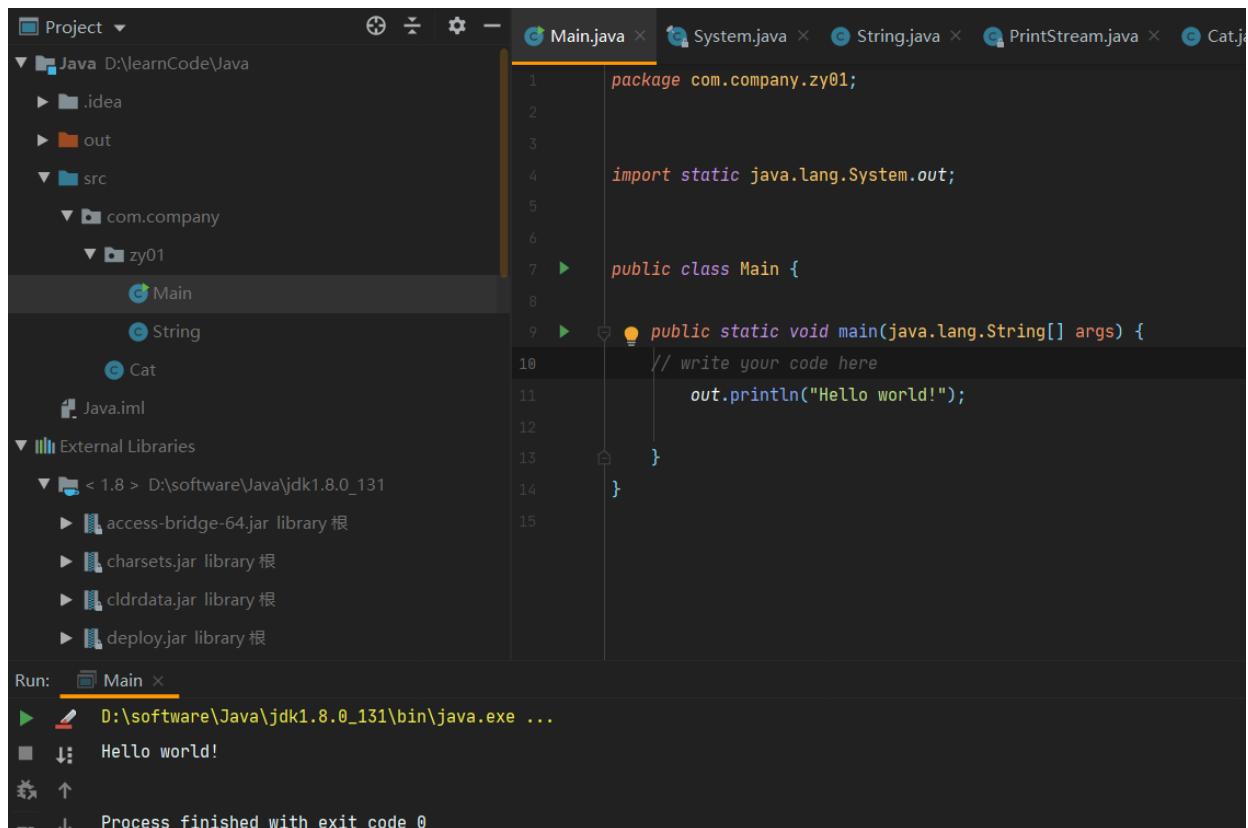
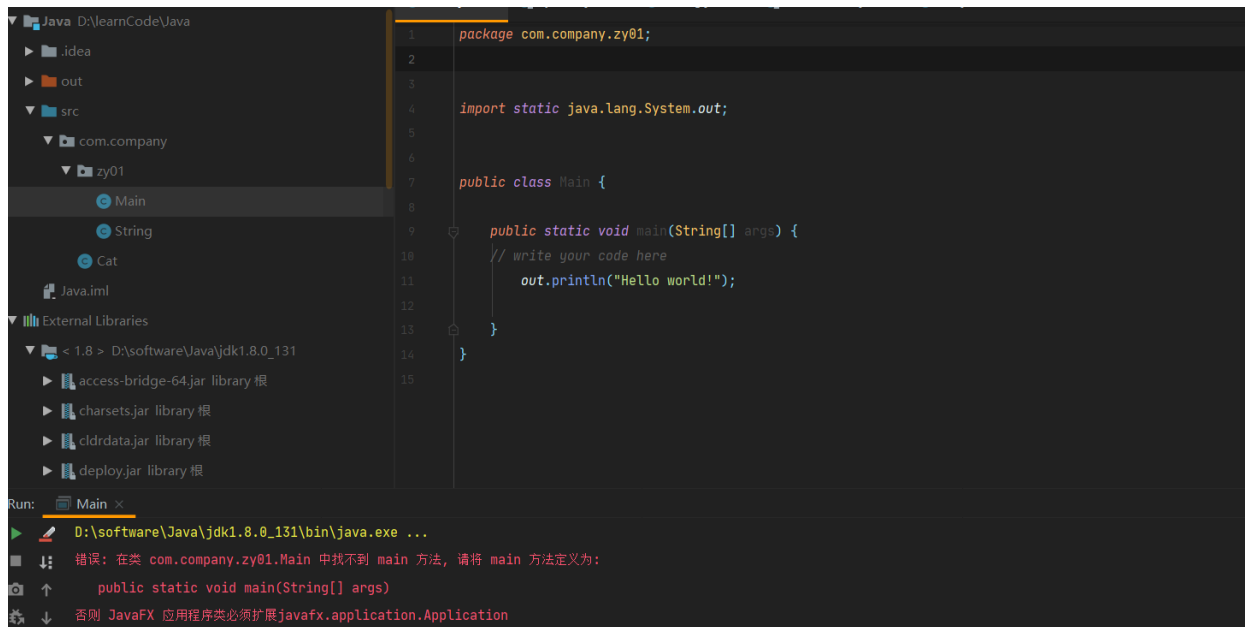
我们会发现 `System` 类也来自于一个包叫做 `java.lang` 包，为什么这个包不需要导入我们就可以使用 `java.lang` 呢？



因为Java中会默认导入了 `java.lang` 这个包下的所有类，所以说不需要导入直接就可以使用

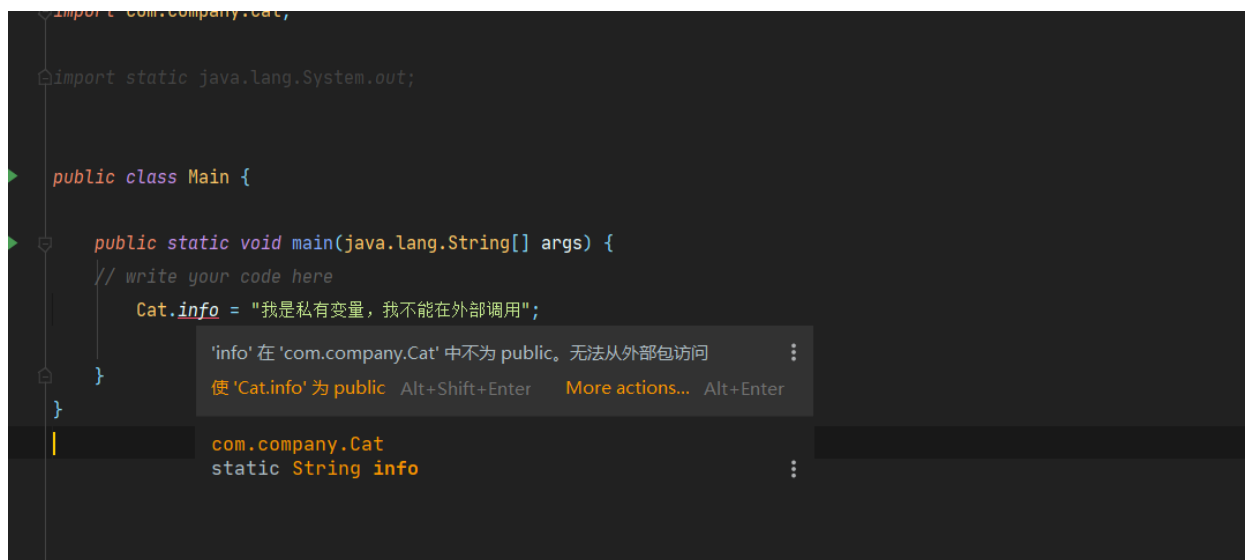
★ 如果我们创建了两个不同包但是名字一样的类（有时候不同包和不同类的方法或者类名字一样但是用途不一定一样），且都在一个代码中调用的时候，程序就会报错

可以看到我在包中创建了一个类名为 `String` 这个 `String` 和 `java.lang` 中的 `String` 很明显冲突了，但是我这里的编译器默认是调用同包下的 `String` 导致代码报错，如果是这样，可以通过写完整包名的全称来避免



★ 权限问题: 上面提到了System的构建方法是私有的所有不能创建类对象, 那么开始我们调用 `Cat` 类的时候, 也会报错

在此有个问题, `info` 静态变量是个默认的变量不是私有变量



权限和访问

在Java中有权限访问的功能，就和我们人与人之间一样，谁都不会随随便便就让别人知道自己的隐私的吧？只有经过你自己的同意之后才行；Java的权限控制也是同理，在Java中可以为成员变量、成员方法、静态变量、静态方法、类来指定访问权限，权限不同访问的范围也不同

- `private` - 私有，标记为私有的内容无法被除当前类以外的任何位置访问。
- 什么都不写 - 默认，默认情况下，只能被类本身和同包中的其他类访问。
- `protected` - 受保护，标记为受保护的内容可以能被类本身和同包中的其他类访问，也可以被子类访问（子类我们会在下一章介绍）
- `public` - 公共，标记为公共的内容，允许在任何地方被访问。

这四种访问权限，总结如下表：

	当前类	同一个包下的类	不同包下的子类	不同包下的类
public	✓	✓	✓	✓
protected	✓	✓	✓	✗
默认	✓	✓	✗	✗
private	✓	✗	✗	✗

版权声明：本文为柏码知识库版权所有，禁止一切未经授权的转载、发布、出售等行为，违者将被追究法律责任。

原文链接：<https://www.itbaima.cn/document/dncxjecdv4wciqcp>

在上面我们出现的问题就是出现在权限上，上面是默认权限将它们换成 `public` 就可以了

```
1 package com.company;  
2  
3 import sun.awt.windows.WPrinterJob;  
4
```

```

5 public class Cat {
6     public String name;
7     public String sex;
8     public int age;
9     public static String info;
10
11     Cat(String name,String sex,int age){
12         this.name = name;
13         this.sex = sex;
14         this.age = age;
15     }
16     Cat(){
17
18     }
19     static void show(){
20         System.out.println("我是一个静态方法！我不能调用成员属性！因为我属于类，不属于对
象！");
21         System.out.println("我可以调用 "+info+" 因为我们都属于类");
22     }
23 }
24

```

```

1 package com.company.zy01;
2
3
4 import com.company.Cat;
5
6 import static java.lang.System.out;
7
8
9 public class Main {
10
11     public static void main(java.lang.String[] args) {
12         // write your code here
13         Cat.info = "我是公共变量，我可以在外部调用";
14         out.println(Cat.info);
15     }
16 }
17

```

⚠ 注意类的权限只能被定义为 默认 和 `public`

```

package com.company;

import sun.awt.windows.WPrinterJob;

private class Cat {
    public static String info;

```

此处不允许使用修饰符 'private'

更改访问修饰符 Alt+Shift+Enter More actions... Alt+Enter