

Lab 7

- 要点：

0.lab6 回顾

- 1.学习并使用 Javadoc
- 2.练习二维数组的使用
- 3.对象和类的熟习

- 具体内容

1、Javadoc

Javadoc 是 Sun 公司提供的技术，它从程序源代码中抽取类、方法、成员等注释形成一个和源代码配套的 API 帮助文档。也就是说，只要在编写程序时以一套特定的标签作注释，在程序编写完成后，通过 Javadoc 就可以同时形成程序的开发文档了。

Java 的注释格式有三种：

- (1) 单行，由//标记，如：//This is comment
- (2) 多行，可由/* ... */标记，如下图

```
/*  
 * comment1  
 * comment2  
 */
```

- (3) 多行，也可由/** ... */标记，如下图：

```
/**  
 * This is test.  
 * @author LiuQishuo  
 * @version 1.0  
 */
```

而第(3)种格式，就是专门为 Javadoc 的自动化生成而规定的。关于 Javadoc 的详细信息以及规范，大家可以参考本次 Lab 目录下的资料，下面仅给出简略的教程：

Javadoc 的生成，简单的说就是我们通过在 Java 代码文件中通过固定的注释格式来声明一些信息，然后在运行 Javadoc 命令的时候，相关的 Java 组件就可以通过抽取这些信息来组成一个规范化的文档（类似于大家使用的 Java API 文档）

Javadoc 一般会用到如下的关键字来描述相应的信息：

标记	用于	作用
@author	对类的说明	标明开发该类模块的作者
@version	对类的说明	标明该类模块的版本
@see	对类、属性、方法的说明	参考转向，也就是相关主题
@param	对方法的说明	对方法中某参数的说明
@return	对方法的说明	对方法返回值的说明
@exception	对方法的说明	对方法可能抛出的异常进行说明

以下是一个简单的有 Javadoc 描述的例子：

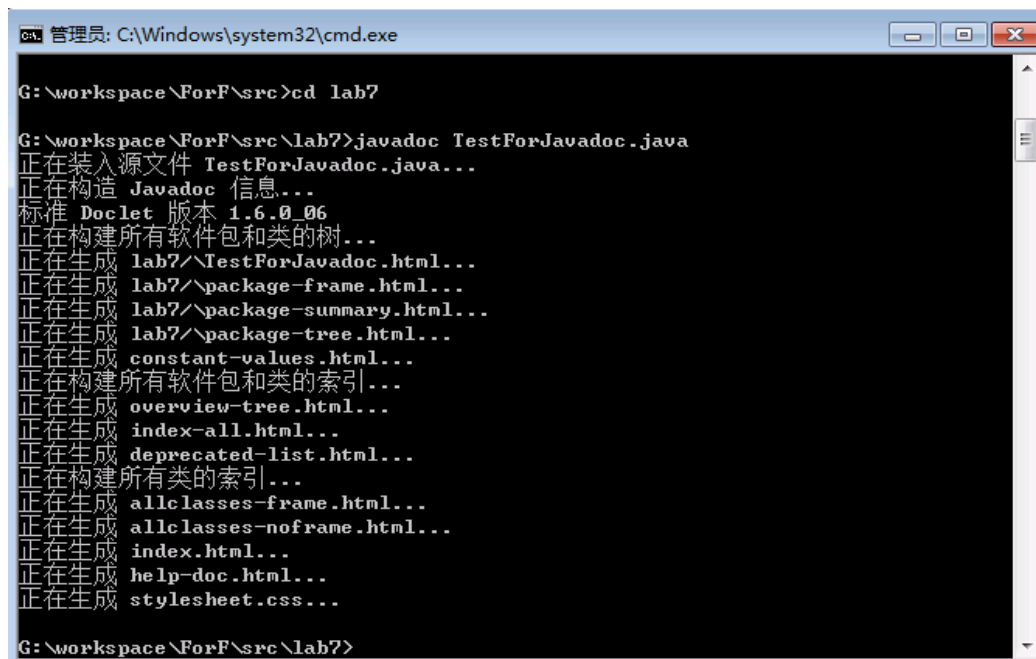
```
import javax.swing.JOptionPane;

/**
 * This class is to change upper case into lower case.
 * @author ZJ
 * @version 1.0
 */
public class TestForJavadoc {

    /**
     * This method is to change one upper-case character into lower-case.
     * @author ZJ
     * @param ch is the character which need to be changed into lower case.
     * @return the lower-case character.
     */
    public static char changeUpperCaseToLowerCase(char ch){
        return (char)((int)ch-((int)'A'-(int)'a'));
    }

    /**
     * This is main method.Input a upper case character and then use the
     * method of changeUpperCaseToLowerCase to change it into a lower case
     * @author ZJ
     * @param args
     */
    public static void main(String[] args){
        String in = JOptionPane.showInputDialog("请输入一个大写字母：");
        if(in.charAt(0)>='A'&in.charAt(0)<='Z'){
            JOptionPane.showMessageDialog(null, changeUpperCaseToLowerCase(in.charAt(0)));
        }else{
            JOptionPane.showMessageDialog(null,"对不起，您的输入无效！");
        }
    }
}
```

然后，在定位到源代码文件所在目录的 cmd 下或使用 Eclipse 自带的 Javadoc 导出功能就可以导出 Javadoc。以 cmd 为例，执行 javadoc ClassName.java 命令就可以在当前目录下生成 Javadoc，如下图：



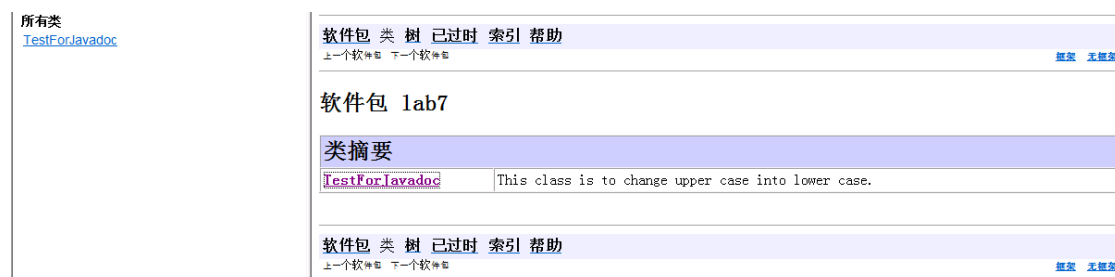
```
管理员: C:\Windows\system32\cmd.exe

G:\workspace\ForF\src>cd lab7

G:\workspace\ForF\src\lab7>javadoc TestForJavadoc.java
正在装入源文件 TestForJavadoc.java...
正在构造 Javadoc 信息...
标准 Doclet 版本 1.6.0_06
正在构建所有软件包和类的树...
正在生成 lab7\TestForJavadoc.html...
正在生成 lab7\package-frame.html...
正在生成 lab7\package-summary.html...
正在生成 lab7\package-tree.html...
正在生成 constant-values.html...
正在构建所有软件包和类的索引...
正在生成 overview-tree.html...
正在生成 index-all.html...
正在生成 deprecated-list.html...
正在构建所有类的索引...
正在生成 allclasses-frame.html...
正在生成 allclasses-noframe.html...
正在生成 index.html...
正在生成 help-doc.html...
正在生成 stylesheet.css...

G:\workspace\ForF\src\lab7>
```

然后我们可以看到源代码所在目录下生成了一大坨 html 文件，我们可以双击打开 index.html 文件查看生成的 Javadoc：



软件包

类树

已过期

索引

帮助

上一个类

下一个类

概要

类图

字段

构造方法

方法

框架

无框架

详细摘要

字段

构造方法

方法

lab7

类 TestForJavadoc

java.lang.Object

└ lab7.TestForJavadoc

public class TestForJavadoc
extends java.lang.Object

This class is to change upper case into lower case.

构造方法摘要

TestForJavadoc()

方法摘要

static char

changeUpperCaseToLowerCase(char ch)

This method is to change one upper-case character into lower-case.

static void

main(java.lang.String[] args)

This is main method. Input a upper case character and then use the method of changeUpperCaseToLowerCase to change it into a lower case

从类 java.lang.Object 继承的方法

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

构造方法详细信息

TestForJavadoc

本次 Lab 的第一个任务是，参考以上给出的简单例子和相关参数信息，写出并正确生成 TestForJavadoc.java 的 Javadoc。 课后大家可以详细阅读 Lab 目录下给出的其他资料来加深对于 Javadoc 的认识，同时我们也鼓励大家生成 Project1 的源代码文件的相应的 Javadoc。

注意：

- (1) Javadoc 的信息必须严格按照/** ... */的注释格式来写
- (2) 描述某个类、方法或者某个变量的 Javadoc 信息的注释块必须严格置于这个类、方法或这个变量的前面，两者之间不能有空行、其他语句等无关信息，如以上例子所示。

2、扫雷游戏

如下图所示，16*16 的方格，初始时生成一个地图并打印出来，用@来表示地雷，无地雷处可用 L 显示。

```

L L L L L L L L L L @ L L L L L
@ L L L L L @ L L L @ L @ @ L L
L L L L L @ L L L L L @ L L @ L
L L L L L L @ L L L L L L @ L L
L L L L @ @ @ L L L L L @ L @ L
@ L L L L @ @ L L L @ L L L L L
@ L L L L L L @ @ @ L L L @ L L
L L L @ L L @ @ @ L L L L L L @
L L L @ @ L L L L @ L @ L @ L L
L L L L L L L L L L L L L L @ L
L L L L L L L @ L L L @ @ L L L
L L L @ @ L L @ L @ @ L @ L L L
L @ @ L @ L L @ @ L L @ L L L @
@ @ @ L L L L L L @ @ L L @ L L

```

从终端读取所要输入的地理坐标，如果不是地雷，则输出其相邻 8 个方格地雷的个数，并打印当前地图，未知区域可用 Q 来代替；如果是地雷，则输出“@@我踩地雷了@@”，显示全地图信息，退出游戏（效果与输入-1,-1 等同）

请输入地雷的坐标位置：

1,2

@@我踩地雷了@@

```

2 @ @ 3 2 1 0 0 2 @ 2 0 1 1 1 0
@ 4 4 @ @ 4 2 1 2 @ 3 1 1 @ 2 1
1 2 @ 4 @ @ @ 2 3 4 @ 1 1 1 3 @
0 2 2 3 2 3 2 2 @ @ 3 2 1 0 2 @
1 2 @ 1 0 0 1 3 4 3 2 @ 3 2 2 1
@ 4 2 1 0 0 1 @ @ 2 3 3 @ @ 2 1
@ @ 1 1 1 1 2 3 5 @ 3 @ 3 2 3 @
@ 4 2 1 @ 2 3 @ 4 @ 3 1 1 0 2 @
3 @ 3 2 3 @ 4 @ 3 2 2 1 1 2 3 2
2 @ @ 1 2 @ 4 2 1 2 @ 3 3 @ @ 2
1 3 4 3 2 2 @ 2 1 2 @ @ 3 @ 5 @
0 1 @ @ 1 2 3 @ 1 1 3 3 3 3 @ 4
1 2 3 2 1 2 @ 3 1 0 1 @ 2 3 @ @
2 @ 2 0 0 3 @ 4 1 0 1 2 @ 2 2 2
3 @ 2 0 0 2 @ @ 1 0 0 1 1 1 0 0
@ 2 1 0 0 1 2 2 1 0 0 0 0 0 0

```

请输入地雷的坐标位置：

33

输入的格式不对，请重新输入！

请输入地雷的坐标位置：

-23,45

输入的坐标越界，请重新输入！

请输入地雷的坐标位置：

请输入地雷的坐标位置：

2,6

```
Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q
Q Q Q Q Q 2 Q Q Q Q Q Q Q Q Q Q
Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q
Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q
Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q
Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q
Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q
Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q
Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q
Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q
Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q
Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q
Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q
Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q
Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q
Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q
Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q
Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q
Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q
```

请输入地雷的坐标位置：

-1,-1

```
1 1 1 2 3 3 2 2 1 2 1 2 1 1 1 1
1 1 1 2 3 3 2 2 2 3 2 2 2 2 3 2
2 2 2 2 3 1 1 1 2 4 3 2 2 2 3 1
2 2 1 1 1 0 0 0 1 2 2 1 2 2 4 2
2 3 3 4 3 1 0 0 1 3 4 3 2 2 3 2
1 2 2 3 2 1 1 1 2 3 5 4 3 2 4 3
0 1 2 4 4 3 2 1 2 3 5 4 3 2 4 3
1 1 0 2 3 3 2 1 1 1 2 2 2 2 4 3
1 1 0 2 3 3 1 0 1 1 2 1 2 2 3 2
2 2 1 1 1 1 0 0 1 1 2 1 2 2 3 2
1 1 1 1 1 1 1 1 2 1 3 3 4 2 3 2
1 2 2 3 2 2 1 2 2 1 2 3 3 2 3 3
0 1 1 3 2 2 1 2 2 1 2 3 3 2 3 3
0 1 1 3 2 2 0 1 1 1 1 1 1 1 2 2
0 1 1 2 1 1 0 0 0 0 0 0 0 0 2 2
0 1 1 2 1 1 0 0 0 0 0 0 0 0 1 1
```

注意：

- 相邻的 8 个位置是：左上、正上、右上；正左、正右；左下、正下、右下；不包括自身
- 在四个角和四条边上，相邻的位置没有 8 个
- 输入：如 2,2（英文逗号隔开，表示第二行第二列的位置）-1,-1（计算地图上地雷的分布情况，有雷位置用@表示，无雷位置标示各个相邻位置的地雷总数，打印全图）（坐标如何设计更方便，可以好好想想）

要求：

- 建立一个名为 game 的包，包含 Mine.java，来实现主要的函数
- 另建一个名为 run 的包，里面有文件 Run.java，它调用 game 包里 Mine 类中的处理方法
- Run.java 只是调用 game 包所提供的类和方法，不得还有其他语句。即在 Run.java 的 main 方法中只能包含类似下面的语句以及其他注释语句。

```
Mine mine = new Mine();
mine.readInput();
```

```
mine.process();  
mine.printOut();
```

- 检查内容

本次 lab 需要提交简单的开发文档，简要说明自己程序的结构（200 字以上），（鼓励用 javadoc 的方式得到开发文档，最后需提交一坨生成的文件）并附上程序的运行图（至少一张截图）

Javadoc 的正确生成（index.html）——2 分

扫雷游戏的完成——8 分

- 截止时间

完成后，将相关文件和代码以“学号+姓名”（例：13302010xxx_李四）上传到 FTP 目录

“/classes/13/131 程序设计（戴开宇）/WORK_UPLOAD/Lab7/”

Deadline: 11 月 5 日 22 点 59 分

抄袭 0 分 !!!