

术语

面向对象的编程Java API（标准类库）文档
内联文档（注释）类
方法标识符保
留字空白区域

机器语言汇编语言
高级语言编辑器
编译器解释
器源代码
Java字节码
IDE

调试器SDK
/ JDK
语法语义
编译时错误运行
时错误逻辑错误

编码标准由开发人员在编写源代码时使用的指南组成，以确保格式一致。

课程编码标准

Checkstyle支持课程编码标准，这是一个自动化工具，可与jGRASP IDE和Web-CAT程序评分系统配合使用，根据课程采用的规则或检查自动检测样式错误。前两条规则介绍如下。

- 代码行不得超过80个字符（包括缩进）。您可以使用字符串连接在新行上继续字符串文字，字符串连接从两个或多个其他字符串创建字符串）。

例如：

```
System.out.println（“假设这是一个长输出字符串。”）；
```

可以改写为：

```
System.out.println（“假设这是一个很长的”  
+“输出字符串。”）；
```

请注意，串联不会更改程序的输出。

- 对于活动和项目分配，所有公共类和公共方法都需要Javadoc注释。在下面的HelloWorldCommented程序中，斜体文本描述了该行的信息。显示的信息应由您自己的信息替换。

```
/**  
 * 打印Hello World的简单程序。 你的程序描述在这里。  
 *  
 * 活动!（您的活动或项目编号在此处）。  
 * @author你的名字 - 课程 - 适当的部分  
 * @version日期  
 */  
公共类HelloWorldCommented {  
    /**  
     * 打印Hello World一次。 通常在这里描述较短。  
     * @param args命令行参数 - 未使用。  
     */  
    public static void main (String [] args)  
    {System.out.println（“Hello World”）；  
    }  
}
```

目标

在本活动结束时，您应该能够执行以下操作：


- 创建一个程序，然后编译并运行它。将Javadoc注释添加到您的程序中并为您的程序生成文档
- 使用Checkstyle更正源代码结构
- 将完成的程序提交到Web-CAT进行评分


地图

A部分：使用jGRASP和Checkstyle

- 设置文件夹 - 在计算机上打开文件浏览器（例如，Windows上的文件浏览器或MacOS上的Finder），并为此活动设置适当命名的文件夹（例如，Activity_01）。
- 打开jGRASP - 例如，在Windows上，单击屏幕左下角的“开始”按钮，然后在“搜索”窗口中，输入jgrasp，然后选择jGRASP。在MacOS上，输入Cmd-space，输入jgrasp，然后双击它。
- 使用jGRASP中的顶层菜单，打开一个新的Java窗口（File> New> Java）。输入以下Java语句以创建名为StudentInfo的类：

```
公共类StudentInfo  
{  
  
}
```


- 单击“保存”  顶部菜单上的按钮，然后在另存为对话框中，使用对话框顶部附近“查找范围”最右侧的按钮向上导航/到您所在的文件夹

Look In:  Activity_01





想保存你的程序。如果您尚未创建课程文件夹（并且在其中包含此活动的文件夹），您现在可以通过单击文件夹/按钮创建新文件夹来执行此操作。单击名称“新建文件夹”以选择它，然后再次单击它并将名称更改为此课程的编号。双击该文件夹将其打开，然后（如上所述）为此活动创建一个文件夹并将其打开。如果您的程序代码正确，您应该看到StudentInfo.java作为文件名。如果不是这种情况，请手动输入文件名。现在单击对话框底部的Save按钮。

- 单击Compile按钮/，并修复任何编译时错误。
- 通过按F2或单击jGRASP桌面顶部工具栏上的Generate CSD按钮/来生成CSD。现在打开自动生成CSD（查看>然后选中自动生成CSD框），以便每次加载或编译文件时都会生成CSD。您可以随时关闭/打开它。
- 在类中添加main方法。请务必使用方法名称替换下面代码中的空白（不要忘记重新生成CSD）：

```
public class StudentInfo
{
    public static void _____(String[] args)
    {
    }
}
```

- 单击Compile按钮, 并修复任何编译时错误。
- 现在, 通过调整第1页上的Javadoc注释, 为该类和该程序的主要方法添加Javadoc注释, 以便它们描述该程序并获得您的姓名, 课程信息和今天的日期。
- 在main方法中, 写下三个或更多的打印语句: (1) 第一行打印名称: 第一行打印您的姓名 (第一个和最后一个), (2) 打印上一个计算机课程的第二个: 在第二行, (3) 第三个打印三个空格, 后跟您所拥有的计算机课程名称 (如果这是您的第一个课程, 则为无), (4) 如果您有多个计算机课程, 则为每个课程打印一个额外的行 (务必以三个空格开始每个课程)。不要忘记在println语句中的String文字周围加上双引号。

```
public static void main(String[] args)
{
    System.out.println("Name: _____");
    System.out.println("Previous Computer Courses:");
    System.out.println("    _____");
}
```

- 单击“编译”按钮, 并修复任何编译时错误。 请注意, 如果启用了“自动保存” (默认设置), 则每次编译时都会保存程序。
- 单击“运行”按钮。 确保输出正确。
- 单击jGRASP工具栏上的Run Checkstyle按钮/并更正Checkstyle标识的任何问题。 如果您没有看到Checkstyle按钮, 则可能未在您的计算机上安装Checkstyle。 在您的个人计算机上, 您需要下载并安装 (解压缩) Checkstyle。 您可能还需要在jGRASP中配置Checkstyle (工具> Checkstyle>配置), 以便jGRASP可以找到包含Checkstyle JAR文件的文件夹。
- 单击工具栏上的/可以打开/关闭行号。 大多数用户都会留下行号。
- 单击“浏览”按钮在工具栏上。 这将打开包含CSD窗口中文件的文件夹上的jGRASP Browse选项卡。 您应该在“浏览”选项卡中看到StudentInfo.java带下划线。 如果文件编译成功, 您还应该看到编译器生成的相应.class文件 (StudentInfo.class)。 此文件包含程序的字节码, 用于在单击“运行”按钮时运行程序_____。

在B部分中, 您将开发第二个程序CourseInfo.java, 然后同时将StudentInfo.java和CourseInfo.java提交给Web-CAT。

B部分：使用Checkstyle, Web-CAT和jGRASP项目

1. 使用Checkstyle查找和纠正样式错误

- 下载Part B zip文件并将其保存在此活动的文件夹中。 然后解压缩CourseInfo.java 文件（Windows：右键单击zip文件，然后选择Extract All ...；Mac OS X：只需打开zip文件）。
- 双击Browse选项卡中的文件打开CourseInfo.java。 您将负责纠正程序中存在的样式错误和逻辑错误。 第一步是修改程序以遵守课程编码标准。 为此，请运行Checkstyle并更正出现的所有格式问题。
 - 提示：Checkstyle声明主方法缺少Javadoc注释，这是真的。 提示：确保您知道//到行尾注释，/*单行或多行注释*/和/** Javadoc注释*/之间的区别。
 - 提示：要处理超过80个字符的源行，请参阅此活动的第1页。
- 程序的预期输出如下所示（左侧的行号不是预期输出的一部分）：

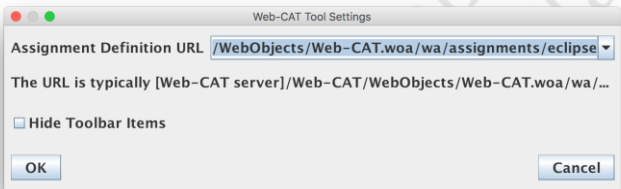
1	本课程提供介绍	至 Java和
2	面向对象的编程。	
3		
4	该课程还介绍了基础知识	的 软件 发展。
5		

该程序的原作者声称该程序是正确的，因为它“得到了重点”，但您知道它是不正确的，因为程序的输出与您在测试期间运行它时的预期输出不匹配。 通过修改程序来纠正输出错误。 提示：查找格式化/间距，拼写，大小写等问题。跳过一行的一种方法是使用以下语句打印一个空行：`System.out.println () ;`

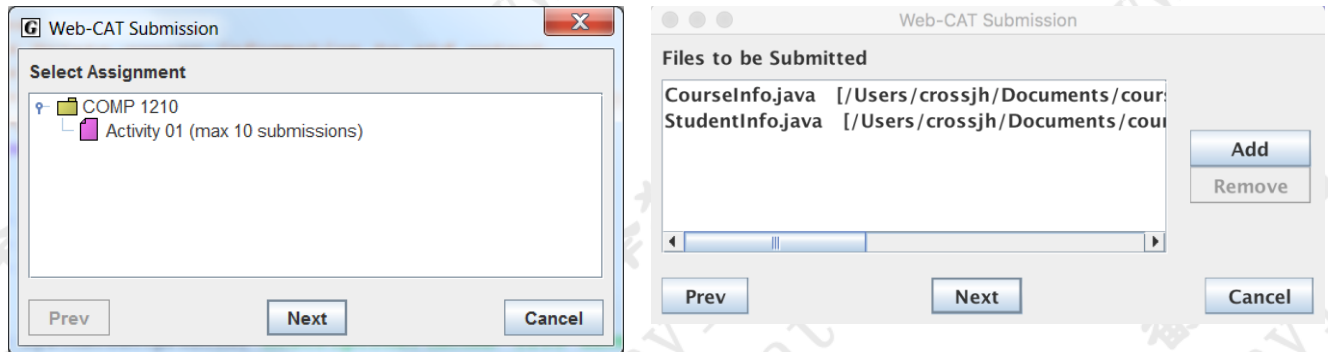
2. 使用Web-CAT对您的程序进行评分

在jGRASP中，导航到工具> Web-Cat>配置。 在“Web-CAT工具设置”对话框中，删除“分配定义URL”的当前条目（如果有）。 选择（而不是单击）下面的URL，右键单击并选择“复制”，然后右键单击并选择“粘贴”。 最后，单击“确定”。 您现在应该在工具栏上有一个Web-CAT按钮/。（有关更多信息，请参阅类网站上的Web-CAT文档。）

`http: // WebCAT. eng. auburn . edu: 8443 /网络/猫/提供web-cat. woa/wa/assignments/eclipse`



- 打开要提交的Java文件，然后可以通过Web-CAT提交程序。 单击Web-CAT工具栏按钮/（或转到工具> Web-Cat>提交文件）。 例如，在下图中，显示了一个课程文件夹，其中的分配标记为“活动01”。单击“活动01”将其选中，然后单击“下一步”；单击“添加”以添加此活动的其他文件，以便列出CourseInfo. java和StudentInfo. java。 然后单击“下一步”，登录Web-CAT，然后单击“提交”。 应该打开一个网页，表明您的Web-CAT提交正在处理中，通常在几秒钟内您就会看到结果。

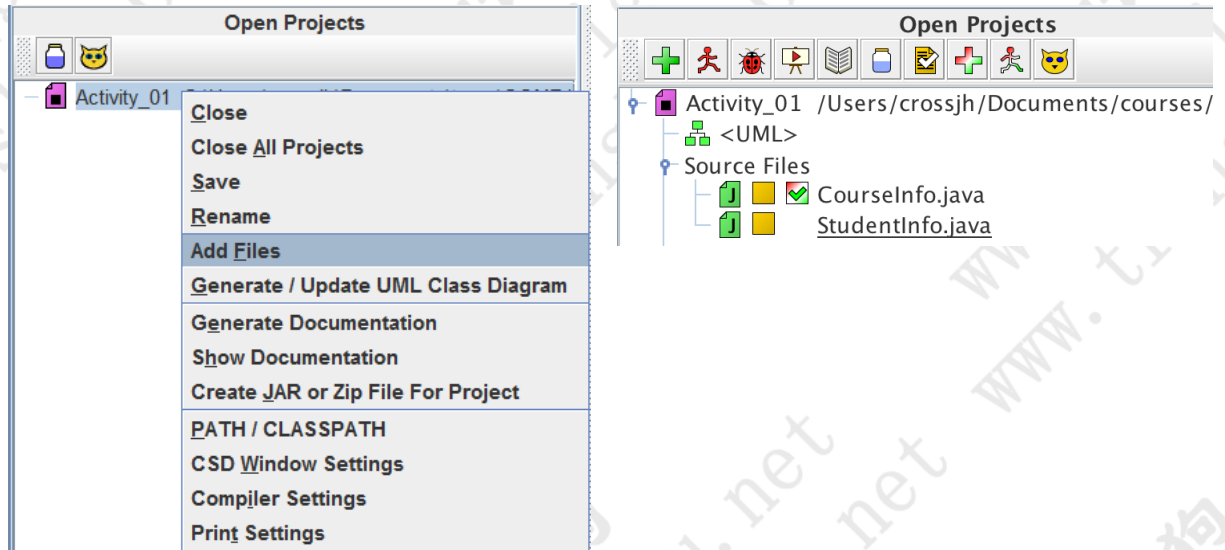


- 您应该尝试使用Web-CAT中的样式/编码+正确性/测试的最大点数（TA点将在后面分配）。 您将进行10次尝试，因此如果您进行更改/更正，请确保您的文件在jGRASP中按预期编译/运行，并且在您再次将文件提交到Web-CAT之前通过了Checkstyle审核。
- 请记住，为了提交两个或更多文件，您可以在一个文件上按Web-CAT /按钮，然后在提交之前添加其他文件（如上所述）或者您可以创建一个jGRASP项目（见下文）。

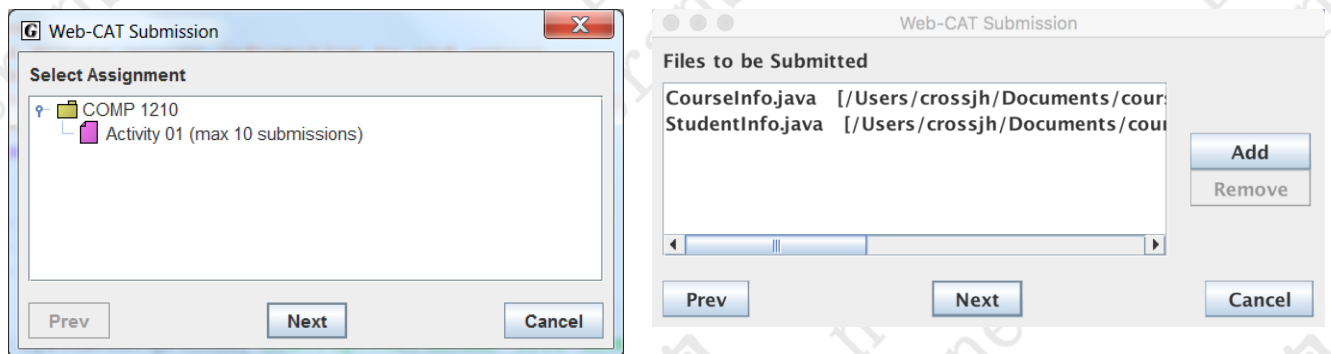
3. 创建jGRASP项目并提交到Web-CAT

您可以创建一个jGRASP项目，将两个Java文件添加到项目中，然后将项目提交到Web-CAT。 这将简化提交过程并节省时间，尤其是当分配中的文件数量增加时。

- 要在jGRASP中创建项目，请转到Project> New，并确保您与源代码文件位于同一目录中。 对于项目名称，输入项目名称（例如，Activity_01），然后单击下一步。 再次单击“下一步”以创建项目。
- 现在，在“浏览”选项卡的下半部分，您将看到“打开项目”部分。 右键单击项目名称，然后选择“添加文件”。 将java文件添加到项目中。



- 添加所有文件后，单击“打开项目”选项卡上的Web-CAT符号/。
- 选择Activity 01分配，单击Next，确保包含所有文件，单击Next，登录Web-CAT（如果尚未登录），然后单击Submit。



- 将来，您应始终通过为其创建单独的文件夹来开始每个活动或项目。 在创建了一个或多个 .java 文件并将它们保存到文件夹之后，您应该创建一个jGRASP项目并添加 .java 文件。 除了在提交到Web-CAT时节省时间，使用jGRASP项目还有许多其他优点，我们将在课程进展中探索这些优势。 例如，您可以从Javadoc注释中以链接网页的形式自动生成项目文档。 作为此活动的最后一个练习，在“浏览”选项卡中找到“打开项目”工具栏，单击“显示/生成项目文档”按钮/。 应打开一个显示项目文档的网页。

大多数软件组织都要求在所有程序中使用Javadoc（或类似）注释来支持从程序本身自动生成文档，文档范围从几个文件到数千个文件。