**实验1熟悉上机环境，编写并调试面向对象程序**

### 1.实验目的

（1）分析和实践基于Java的面向对象程序调试技巧。

（2）能够区分与面向过程程序的不同，能够使用面向对象程序语言（Java）实现300行以内的程序，并进行调试。

### 2.知识要点

##### 2.1 Java编程运行环境的安装（请自学）

（1）在http://java.sun.com根据自己的操作系统下载安装程序，按安装说明进行安装。

（2）设置环境变量。

* 以WinXP为例，在“我的电脑”上点右键，新建系统变量JAVA\_HOME 和CLASSPATH

变量名：JAVA\_HOME

变量值：C:\Program Files\Java\jdk1.6.0\_10（填写你自己的JDK安装路径）

变量名：CLASSPATH

变量值：.;%JAVA\_HOME%\lib\dt.jar;%JAVA\_HOME%\lib\tools.jar;（不同版本的JDK .jar略有不同，有的可能是rt.jar，具体要到lib目录下看看。另外注意.;代表相对路径。）

* 选择“系统变量”中变量名为“Path”的环境变量，双击该变量，把JDK安装路径中bin目录的绝对路径，添加到Path变量的值中，并使用半角的分号和已有的路径进行分隔。

变量名：Path

变量值：%JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin;

（例如把该路径添加到Path值的起始位置，

即%JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin;%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem，以上路径在不同的计算机中可能不同。）打开属性→高级→环境变量→系统变量，选中变量path进行编辑，在后面加入“;C:\ jdk1.7.0\_51\bin”，注意：这里的目录C:\ jdk1.7.0\_51是你安装Java的目录。

（3）打开命令提示符窗口，输入javac命令按回车，如果出现命令的使用参数说明，

则表示设置正确，如果出现非法命令的错误提示，则应该重新设置。

（4）打开Eclipse会提示是否安装了JDK。

（5）JDK包含的基本组件包括：

•javac – 编译器，将源程序转成字节码

•jar – 打包工具，将相关的类文件打包成一个文件

•javadoc – 文档生成器，从源码注释中提取文档

•jdb – debugger，查错工具

##### 2.2 使用Eclipse调试Java程序代码

###### 2.2.1 创建Java项目

**注意：支持Java开发的集成工具有很多(Eclipse, NetBeans, JCreator, JBuilder,…)，我们这里选用Eclipse讲解。**

* Eclipse开发工作台简介

　　Eclipse首次启动时进入的是资源透视图，在这个透视图中可以管理项目、文件夹、文件和其它资源，按照Eclipse的说法，这些面板叫做视图，一套完整的视图叫做透视图，在资源透视图中，你可以使用左上方的视图（称为“导航视图”）导航和创建资源。

* Java创建一个新的Java项目

在Eclipse中创建Java程序，需要先创建一个Java项目，创建Java项目的步骤如下：

　　1、在导航视图上点击右键，在弹出的上下文菜单中选择新建项目；

　　2、在新建项目对话框中，Eclipse提供了项目选项：Java，插件开发等。因此你想要创建一个Java项目，在对话框左边选择Java；

　　3、在对话框右边选择Java项目，如果你还安装了其它Java插件开发包，在这里也会一起列出来(如EJB，Servlet等)，Eclipse默认安装的JDT仅支持标准Java应用程序，因此你必须选择Java项目选项；

　　4、点击下一步进入新建项目向导对话框；

　　5、首先给项目命一个名字，点击下一步，设置构建Java项目的设置选项，对于这个项目你不需要做任何修改即可；

　　6、点击完成。

###### 2.2.2 创建Java类

　　当你创建好Java项目后就可以开始创建Java程序了，尽管不是非得按照先建一个项目再创建类的步骤这么做，可以将你的Java类组织到包中，也便于后续的开发，这是一个很好的编程习惯。通常使用域名做为包名，这样可以减少类的名字相互之间冲突的可能性。

　　按照下列步骤创建Java程序：

　　1、在你所建好的项目名上点击右键，选择New->Class弹出新建Java类向导;

　　2、第一个字段区域“源文件夹”默认是项目的文件夹，保持默认值;

　　3、在包字段区域输入org.eclipseguide.demo;

　　4、在类名字段区域输入PolyNode;

　　5、在“Which Method Stubs Would You Like to Create?”区域下，选中public static void main(String[] args)，自动地生成Main函数。新建Java类。

　　6、点击完成，新建Java类向导将会为程序包创建一系列目录，PolyNode.java源文件将会放在这个包名下面。

自动创建的代码包括一个main( )方法，你可以向里面添加任何功能。在编写JAVA类代码的时候，Eclipse提供的许多周到的服务：

1）每一对括号的自动对齐

2）语法的高亮显示。类的元素将以独特的颜色显示，比如说注释、关键字、字符串等等。

3）显示错误。读懂对错误的追溯。

4）提供代码辅助功能。

###### 2.2.3运行Java程序

运行以上的Java程序。

　　1、从Eclipse菜单中选择运行按钮以运行Java应用程序;

　　2、因为你已经对Java程序做出了修改，Eclipse会提示你在运行Java程序前先保存一下，点击确定;

3、任务视图切换到控制台视图，显示程序输出。

Java程序的调试分编辑、编译、运行3个步骤。并没有单独的步骤将.java文件先编译成.class文件，这是因为Eclipse JDT包含了一个增量的编译器来评估你输入的Java程序代码，它可以高亮显示语法错误和不完整的引用（示范），如果编译成功，.class文件会在保存源文件时同步保存。

###### 2.2.4调试Java程序

在Eclipse中交互式运行代码是其最强大的特性之一，使用JDT调试器，你可以逐行执行你的Java程序，检查程序不同位置变量的值，这个过程在定位代码中的问题时非常有用。也是一款检查和修复Java程序代码问题的不可替代的工具。

先实验一下简单的设置断点调试功能。

　　为了准备调试，你需要在代码中首先设置一个断点，以便让调试器暂停执行允许你调试，否则，程序会从头执行到尾，你就没有机会调试了。为了设置一个断点，在编辑器左边灰色边缘双击，这里将我们将AddPolyNode() 函数位置设置为断点，此时将会显示一个蓝色的小点，表示一个活动的断点。

　　在调试器下运行程序和运行它非常类似，Eclipse提供了两个选项：选择“使用全方位服务运行调试？”菜单使用一个快捷配置，如果默认选项正确，也可以选择“运行以...调试Java应用程序？”。在这里我们使用后者。

确保编辑器中的代码被全部选中了，然后从主菜单中选择“调试Java应用程序”，Eclipse将会启动程序，切换到调试透视图，在断点暂停执行。

调试透视图包括多个新的视图，都是用于调试使用的，首先，在左上方是调试视图，它显示了所有调用堆和当前所有线程的状态，包括所有已经执行完毕的线程，程序运行到断点位置时，状态显示为暂停。

**方法1：单步调试代码**

　　调试视图的标题栏是一个让你可以控制Java程序执行的工具栏，前面几个按钮和电子设备，如CD播放器的控制按钮风格非常类似，允许你暂停、继续和终止程序，这些按钮让你可以一行一步地执行程序代码，鼠标移动到每个按钮上时都会显示按钮提示信息，如跟踪（step into），单步，返回等。

如按钮step into跟踪，当前执行的程序代码在编辑器中处于高亮状态。(打到jar包里的文件是没有源码的)

**方法2：分布过滤**

一般情况下，你只想跟踪你自己写的类，对于标准类和第三方类通常没有什么问题，是不需要跟踪的，这时候可以使用过滤后跟踪功能，即界面上的“Step With Filters”，设置的方法是选择Window->Preferences->Java->Debug->Step Filters，然后选择列出的包和类，多花点时间设置过滤器可以有效减轻调试难度，当你使用普通的跟踪调试功能时经常会遇到许多未知错误，这时使用过滤跟踪就能很好解决这一问题。

**方法3：评估变量和表达式**

调试视图的右边是一个标签视窗包含视图，在这里你可以检查和修改变量和断点，选择变量标签页，这个视图显示了当前范围的变量及其值。

**方法4：监视点**

有时一个Java程序有许多变量，但你仅对其中一个或几个感兴趣，为了监视选择的变量和表达式，你可以将它们添加到表达式视图中的监视列表中， variables->右键watch。

**方法5：修改变量值**

在调试过程中，我们可以修改变量值。先选好一个变量然后进入变量视图（Variables view），根据变量类型在其对应的Value列里输入值即可。

　　在同一个标签视窗中的是显示视图，它允许你输入任何允许的变量或包括这些变量的实例表达式，选择显示视图然后输入。

**方法6：异常断点**

要找到异常发生的地方比较困难，可以打一个异常断点。Breakpoints工具框中提供了"增加Exception异常断点"。当异常发生时，代码会停在异常发生处，定位问题时应该比较有帮助。如果，我们期望某个特定异常发生时程序能够被中断，以方便查看当时程序所处的状态。通过设置Exception Breakpoint就能达到这一目标。这时候我们不直接为此行代码设置Line Breakpoint，而是为Exception设置Exception Breakpoint。设置Exception Breakpoint的方法与其它类型断点都不同，它不能通过双击左侧栏的方式在代码编辑器上直接进行设置。点击Breakpoints视图右上角形如Ji的图标完成。

**方法7：使用断言**

断言(assertion)是一个Java语句，布尔表达式，程序员认为在程序执行时该表达式的值应该为true。系统通过计算该布尔表达式执行断言，若该表达式为false系统会报告一个错误。

断言是通过assert关键字来声明的，断言功能的使用有两种格式：

* assert expression
* assert expression : detailMessage ;

其中，expression为布尔表达式，detailMessage是基本数据类型或Object类型的值。当断言语句被执行时，Java计算expression的值，如果其值为false，抛出AssertionError异常。对于第二种带有一个详细信息的断言语句，将使用AssertionError类的与消息的数据类型匹配的构造方法。由于AssertionError类是Error类的子类，当断言为false时，程序将在控制台显示一条消息并终止程序的执行。

Eclipse默认是没有开启断言功能的，所以你使用断言也不会看到效果的，首先我们用以下步骤打开断言的开关：

Windows--> Preferences--->Java--->Installed JREs--->点击正使用的JDK--->Edit---> Default VM Arguments文本框中输入:-ea

**本节总结**：

在调试过程中，以下快捷方式必须要了解（最好掌握）的东西：

　　F11――进入DEBUG视图

F5——进入：移动到下一个步骤，如果当前行有一个方法调用，该控件将会跳转到被调用方法的第一行执行。

F6——跳出：移动到下一行。如果在当前行有方法调用，那么会直接移动到下一行执行。不会进入被调用方法体里面。

F7——返回：从当前方法中跳出，继续往下执行。

F8——移动到下一个断点处执行。

### 3. 实验任务

编写一个方阵类，其中封装有对方阵（所有元素均为整数）进行操作的方法，要求完成的功能包括：

1）两个n阶方阵的加；

2）两个n阶方阵的减；

3）两个n阶方阵的乘法；

4）求方阵的转置矩阵；

矩阵转置示例

5）按行、列描述矩阵，能够输出矩阵的标准格式（即其中包含\n字符），例如：

6）从控制台读取命令和解析命令，每个矩阵的输入格式类似于：，要求用户在输入的时候，用以+、-、\*、t分别表示两个矩阵之间相加、相减、相乘和某矩阵的转置。

输入样例：

{{1,2},{3,4}}

+

{{4,5},{5,6}}

输出样例：

5 7

9 10

考虑以下特殊情况情况的处理：

1）不接受空矩阵的加减乘及转置运算

2）方阵是一类特殊的矩阵，其行数和列数应该相等

3）行列数不同的方阵不能进行加减乘运算

4）当矩阵元素数值过大，矩阵运算可能出现过于复杂的情况，本题目中只允许输入int范围内的有效值，如果超出范围了，需要给出输入溢出的提示。

下面给出了一个同学所编写的案例，不同于用C语言完成的数据结构的题目上，这个案例试图写一个对象式程序去完成以上所有要求。但这个程序不能正确地完成题目所要求的功能，请你运用各种测试和调试手段，捕获这些错误并将程序更改正确。**请按照样例格式依次罗列你找到的错误（直接在网页文本框内输入），并将改正后的源码（作为附件，程序需要有注释）一并上交**。

说明：

1. 对于输入格式的其它异常情况（如非法字符、括号不匹配等）进行处理，归属于同一类错误，不重复得分。

2. 给出的程序在矩阵计算的过程中也有可能出现溢出，导致计算结果是不正确的，请你把它修改正确。

**提交实验结果的格式如下：**

错误1：\*\*\*\*\*\*\*

改正方法：\*\*\*\*\*\*\*