实验三 Socket编程实现网络通信

1. **实验目的**

掌握Socket编程思想，并实现简单的Socket应用的连接通信过程。

1. **实验要求**

（1）调试“半成品”Java程序，仅在部分地方需要同学加入自己的内容，分析实验现象，完成实验思考题。主体框架已给出，需要改动的位置较少，不需要Java基础。

（2）课后须提交任务清单。

a、不带服务器身份验证的Simple SMTP的Java程序（SMTPSender.java）；

b、带服务器身份验证的Auth SMTP的Java程序（AuthSender.java，请将个人用户名、密码部分\*\*\*\*代替。）；

c、自主编写通过socket收发信息的客户端和服务器端程序；

d、实验报告：需要试验结果截图并回答实验思考和分析题。

1. **实验环境**

配置了Eclipse的Java开发环境的联网计算机。（配置方法见附件）

1. **实验内容**
2. 不带服务器身份验证的邮件发送客户端实验（Simple SMTP）

按照SMTP通信协议（RFC2821）编写一个简单的邮件发送程序，实验采用学校的SMTP服务器（mails.tsinghua.edu.cn）作为邮件发送服务器。此次实验已经给出了程序的“半成品”，要求同学们先分析，读懂程序，然后添加必要的代码，即可运行。

发送邮件到自己的校内邮箱，提交控制台输出截图和收到的邮件截图。

1. 带服务器身份验证的邮件发送客户端实验（Auth SMTP）

需服务器身份验证要求加上SMTP扩充协议（RFC2554）进行用户信息验证。在不适用服务器身份验证的情况下，学校的SMTP服务器只能把邮件发送到同一个服务器上，而不能把邮件发送到其他服务器上。如服务器为mails.tsinghua.edu.cn，[则只能发送到xx@mails.tsinghua.edu.cn](mailto:则只能发送到xx@mails.tsinghua.edu.cn)的邮件地址。

发送邮件到自己的校外邮箱（如163邮箱或qq邮箱等均可），提交控制台输出截图和收到的邮件截图。

1. 自主编写通过socket收发信息的客户端和服务器端程序

同学们可以自主选取一种熟悉的编程语言和编程环境（C/C++,JAVA,PYTHON等均可），完成最简单的通过socket收发信息的客户端和服务器端程序。

附件中提供了Python编写的样例程序，供大家参考。需要提交服务器端和客户机端的代码源程序及运行结果截图。

有课余精力的同学可尝试完成发送、接收文件或视频等功能，无需提交、不作要求。

1. **实验思考与分析题**
2. Simple SMTP和常用的E-mail客户端在功能结构上的比较。
3. 使用TCP和UDP各自的优缺点比较。
4. 针对程序中出现的问题及解决方法，写出实验体会。

## 附录：配置实验环境

### 配置Java运行环境

先从网络学堂下载所需的文件，包括：

* 执行Java应用程序的Java Runtime Environment (JRE )： jre-6u1-windows-i586-p.exe
* 编辑环境Eclipse IDE：eclipse-SDK-3.2.2-win32.zip
* 可选的Java API文档：j2se6\_documentation.zip

其中第三个为Java 6.0的API文档，用来查阅各个类的方法及属性，对编程没有影响。

下载完成后先安装JRE，执行jre-6u1-windows-i586-p.exe安装程序，完成后就可执行Java应用程序。之后解压eclipse-SDK-3.2.2-win32.zip，执行文件夹内的eclipse.exe即可运行Eclipse，运行后它会询问你设置工作目录workspace，就是存放工作时源代码及其他相关文件的目录，按喜好指定一个，按Ok即可。

### 在Eclipse中开发Java程序

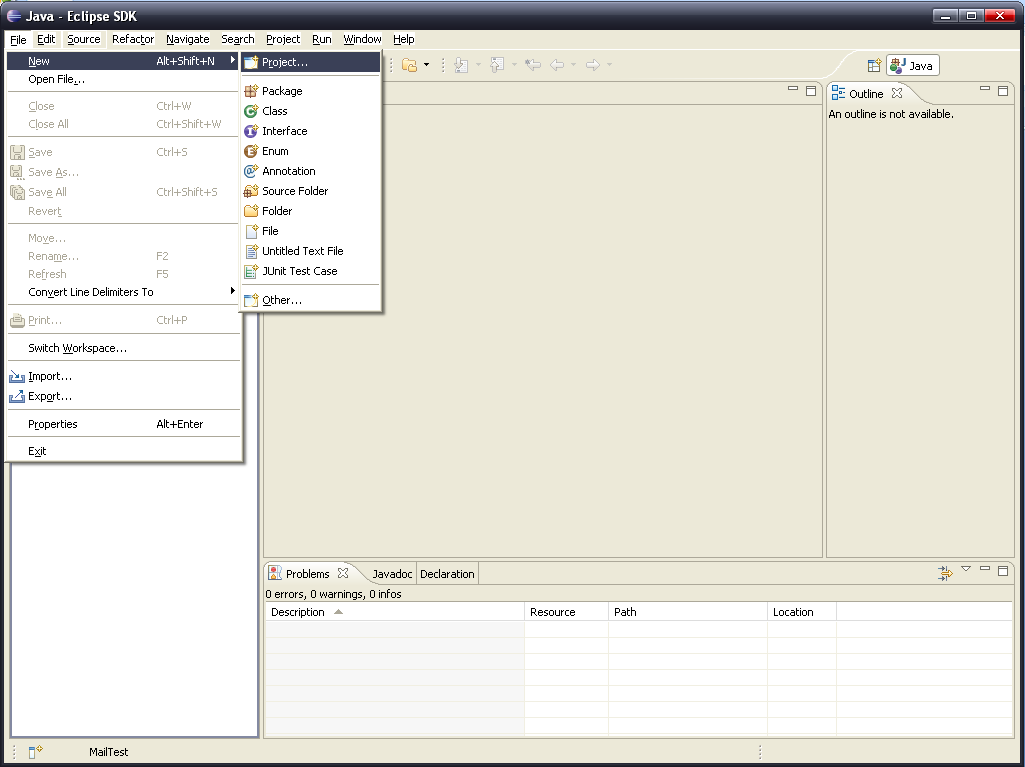
打开程序后如下图，先关闭Welcome标签页。



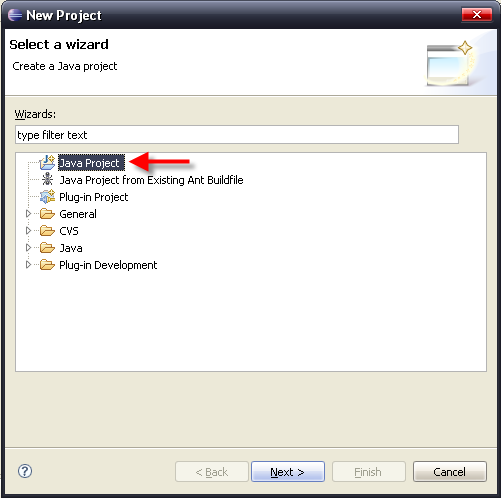
在建立Java程序前先要建Java项目，以下以图示的方式讲述如何运行简单的Hello程序。

* + - 1. 新建Project

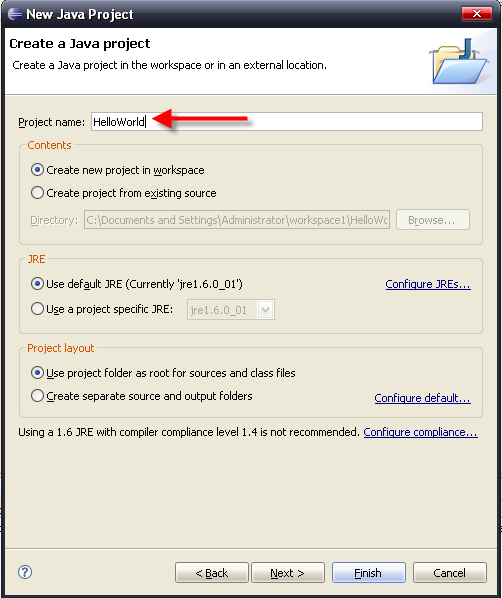
选择File->New->Project



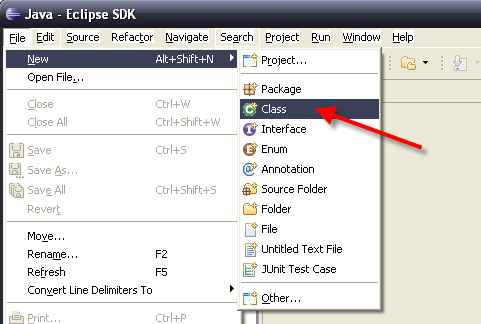
* + - 1. 选择Java Project，按Next。



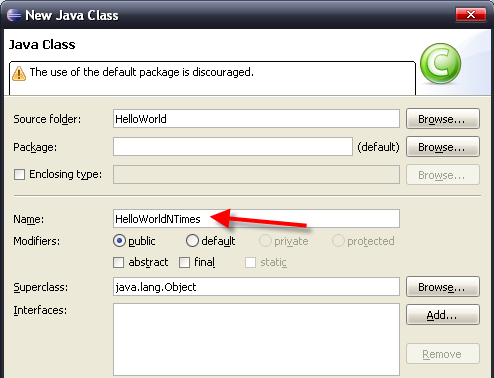
* + - 1. 输入Project名字，这例子中设为HelloWord，之后点Finish即可。



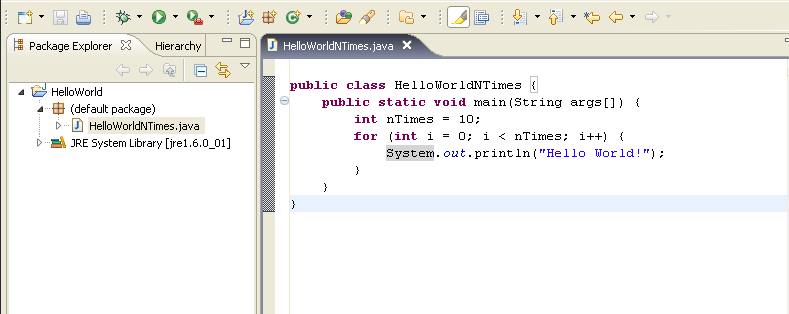
* + - 1. 在菜单中选File->New->Class.



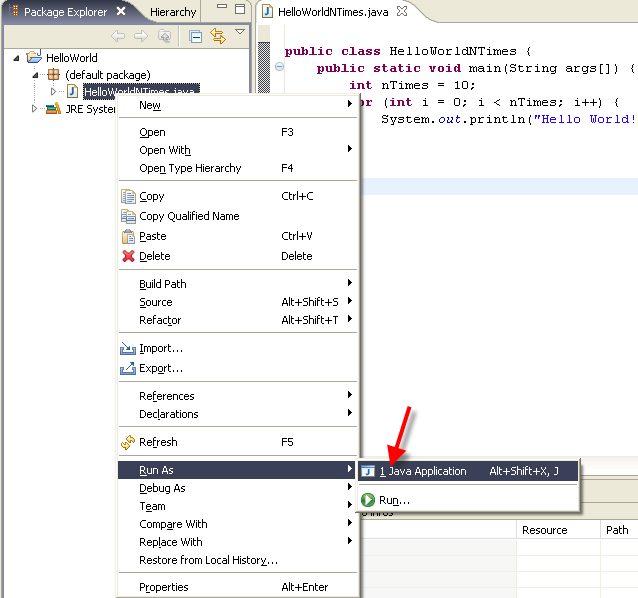
* + - 1. 在Name中输入类的名字，然后点Finish.



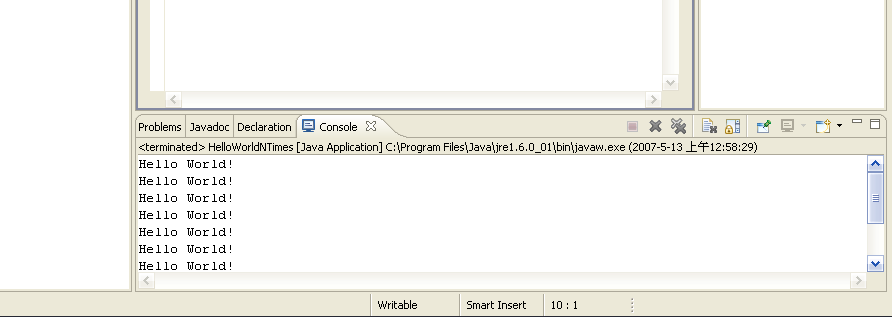
* + - 1. 在HelloWorldNTimes中输入如下程序代码。



* + - 1. 输入完成后，在右边的Package Explorer中点右键，选Run As->Java Application执行程序。



* + - 1. 结果输出在下图的Console中。

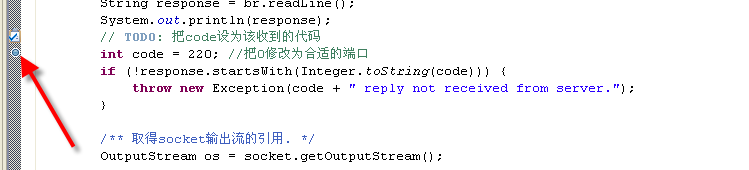


### 在Eclipse中使用调试器

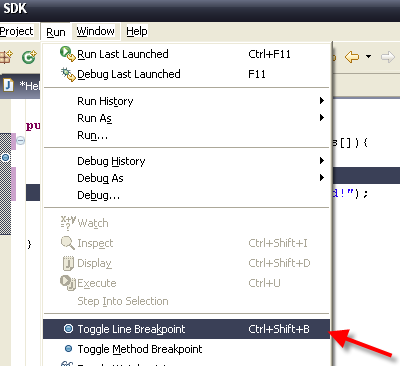
Eclipse中的调试器十分强大，除了可以帮助找出程序出错原因外，还可以帮助了解程序的运行过程。

进行调试时先要设置断点，设置方法有几种。

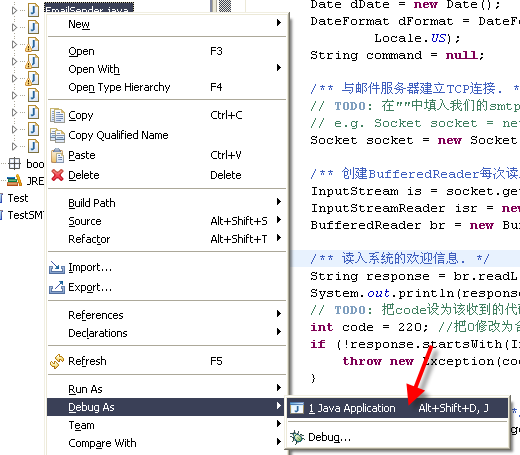
1. 设置断点时，先选中想插入断点的行，然后：
   1. 如图中的第三行，然后在该行最前端的位置（如下图红箭头所指的位置），双击鼠标。



* 1. 选中菜单中的Run->Toggle Line Breakpoint.

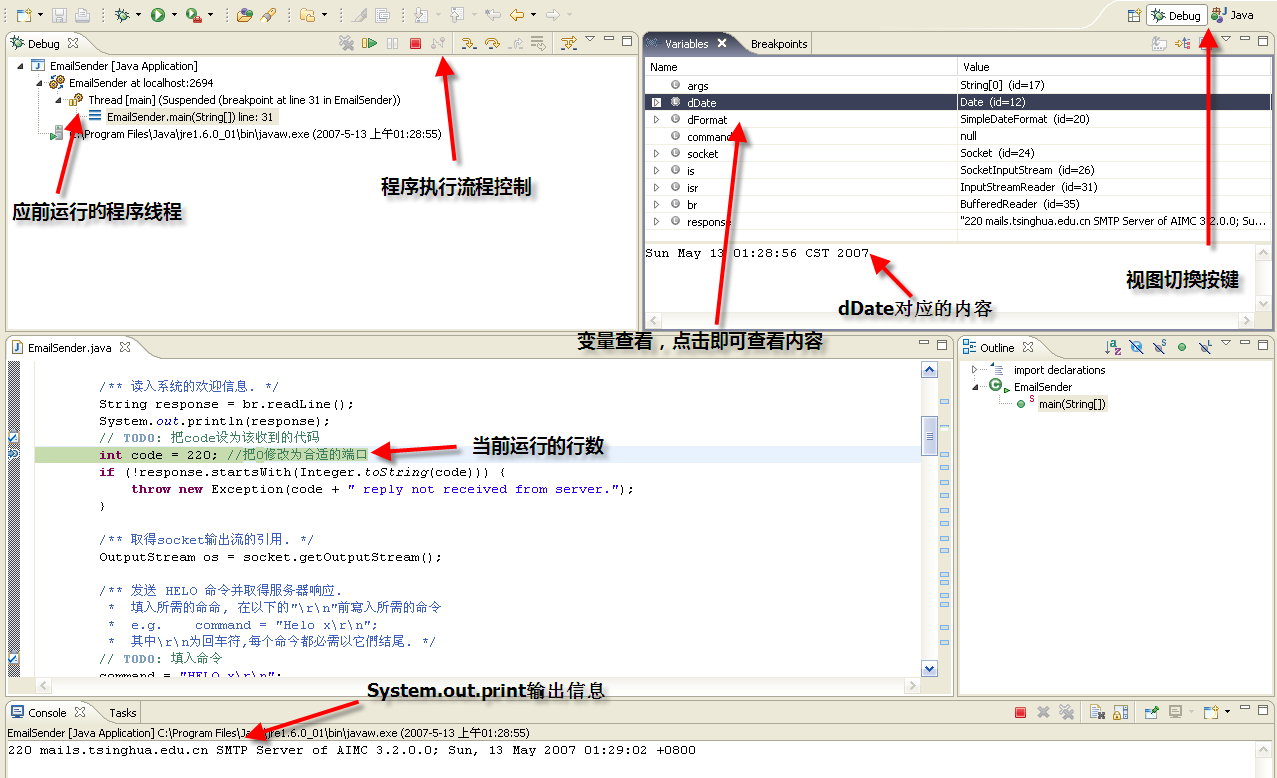


1. 执行程序，与运行程序相似，不过这次我们选中期望执行的程序点右键后选Debug As->Java Application



之后出现Save and Launch窗口，选Ok即可。之后会弹出询问转换视图(Perspective)的对话框，选yes即可。Perspective是Eclipse的特色之一，在使用不用的模式时，方便切换到不同的布局，方便使用。

注：若没有自动切换视图，可在菜单上选Window->Open Perspective->Other…之后在弹出的窗口中选Debug，再选Ok即可。



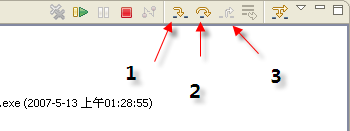
1. 执行语句时，使用下图的执行控制器，其中绿色的Play按键为継续执行，红色的Stop 为停止。下图的1，2，3分别对应Step Into，Step Over和Step Return三种执行方式。

Step Into为程序单歩运行，遇到其他方法调用时，跳转到被调用方法内部継续执行。

Step Over也是单步运行，但在遇到方法调用时，不跳转到方法内部执行，而是执行整句完整语句。

Step Out主要是配合Step Into使用，当Step Into到某个被调用的方法内，一步一步地执行该方法可能会花费大量的时间，此时可使用Step Out跳出。

**注：由于这次实验较为简单，几乎没有什么方法调用，有需要调试的话，为了免去不必要的麻烦，建议只使用Step Over按键。**



程序按以上歩骤单步执行下去，不想执行时点上图红色的Stop按键即可。