

《Java 技术》实验报告

实验一：

2020 年 5 月 20 日

学院	软件学院	专业班级	软件工程 18-4 班	姓名	刘铭源		成绩	
课程名称	Java 技术	实验项目名称	实验一 Java 开发环境的安装与配置			指导教师	薛峰	
教师评语	<div>教师签名：_____ 年 月 日</div>							
	<p>一、实验目的</p> <ol style="list-style-type: none">1. 掌握 JDK 的安装步骤（有条件的话可以在 Linux 系统下安装）。2. 理解环境变量 PATH, CLASSPATH 的作用，以及它们的设置方法。3. 熟悉记事本（或 JCreator）编辑环境，编写简单的 Application 程序，并编译和解释执行。4. 记录编译和执行 Java 程序当中的系统错误信息提示，并给出解决办法。5. 复习、理解 OOP 的三大特征——封装、继承和多态。6. 学会使用 Java 编写简单的类。理解类的属性和方法。7. 掌握关键字 private, protected, public 的作用，掌握静态类，内部类的概念。8. 掌握多态的概念。 <p>二、实验原理</p> <ol style="list-style-type: none">1. 下载最新版本的 JDK，并安装。2. 设置环境变量 PATH, CLASSPATH，使得 Java 程序能正确编译和执行。3. 在记事本（或 JCreator）环境下编写一个 HelloWorld.java 程序，在 DOS 提示符下编译并执行这个程序。4. 编写一个类 A，它位于包 a 中，保存到 A.java 中，再编写一个类 B，它位于包 b 中，保存到 B.java 中。在类 B 中实例化类 A 的一个对象。分别编译类 A 和类 B。5. 分别编写两个类 Point2D, Point3D 来表示二维空间和三维空间的点，使之满足下列要求：<ol style="list-style-type: none">1) Point2D 有两个整型成员变量 x, y（分别为二维空间的 X,Y 方向坐标），Point2D 的构造方法要实现对其成员变量 x, y 的初始化。2) Point2D 有一个 void 型成员方法 offset(int a, int b)，它可以实现 Point2D 的平移。3) Point3D 是 Point2D 的直接子类，它有有三个整型成员变量 x,y,z（分别为三维空间的 X,Y,Z 方向坐标），Point3D 有两个构造方法：Point3D(int x, int y, int z)和 Point3D(Point2D p, int z)，两者均可实现对 Point3D 的成员变量 x, y, z 的初始化。4) Point3D 有一个 void 型成员方法 offset(int a, int b, int c)，该方法可以实现 Point3D 的平移。5) 在 Point3D 中的主函数 main()中实例化两个 Point2D 的对象 p2d1, p2d2，打印							

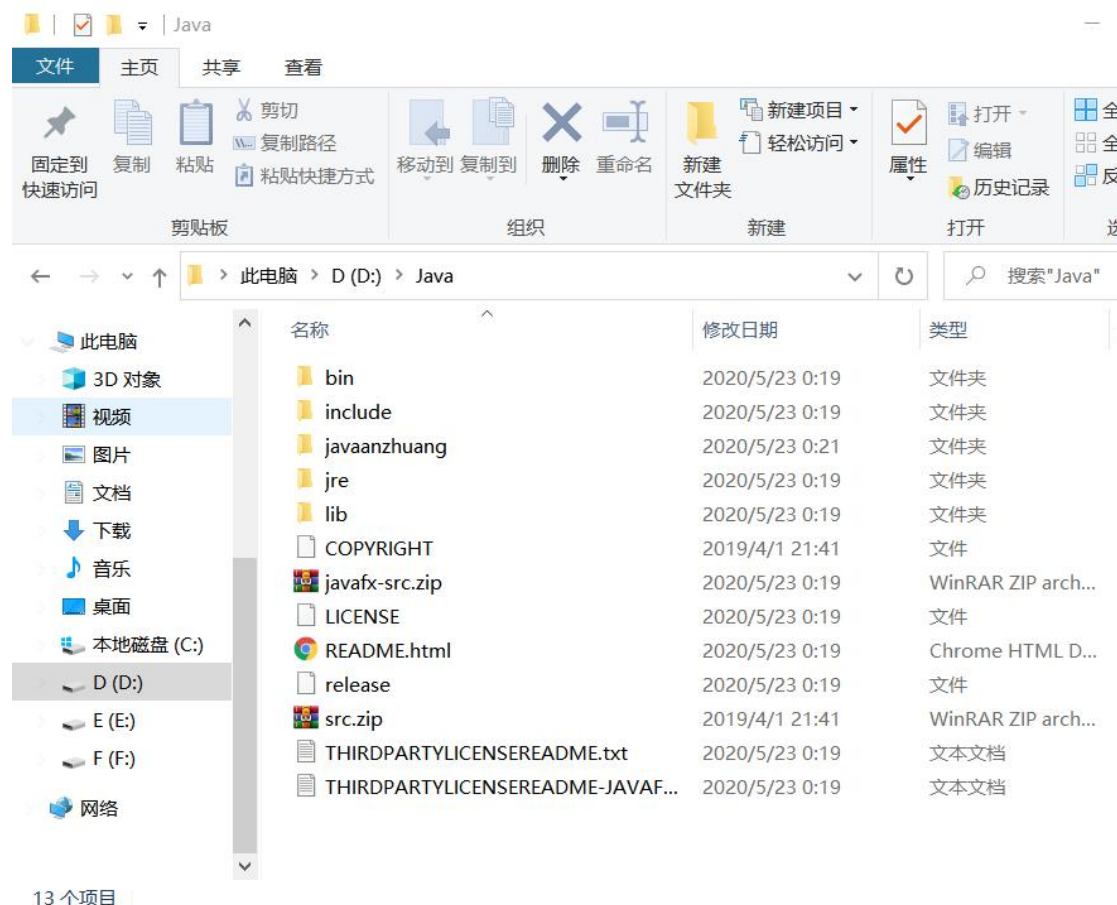
	<p>出它们之间的距离，再实例化两个 Point2D 的对象 p3d1, p3d2，打印出他们之间的距离。</p>
--	---

三、使用硬件、软件环境

Windows10, 内存 8g, 硬盘 1TB, JDK1.8, eclipse

四、实验过程、步骤及原始记录(算法、原程序、测试结果, 分析等)

1. 下载最新版本的 JDK, 并安装



2. 设置环境变量 PATH, CLASSPATH, 使用 Java 程序进行正常编译和执行

环境变量

DELL 的用户变量(U)

变量	值
IntelliJ IDEA	E:\javaexer\IntelliJ IDEA 2020.1.2\bin;
OneDrive	C:\Users\DELL\OneDrive
OneDriveConsumer	C:\Users\DELL\OneDrive
Path	C:\Users\DELL\AppData\Local\GitHubDesktop\bin;C:\Users\DEL...
TEMP	C:\Users\DELL\AppData\Local\Temp
TMP	C:\Users\DELL\AppData\Local\Temp

新建(N)...

编辑(E)...

删除(D)

系统变量(S)

变量	值
ADSK_3DSMAX_x64_2020	F:\adobe\3dsmax2020\3ds Max 2020\
CLASSPATH	.;%JAVA_HOME%\lib;%JAVA_HOME%\lib\tool.jar
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
JAVA_HOME	D:\Java
LM_LICENSE_FILE	E:\modelsim\LICENSE.txt
NUMBER_OF_PROCESSORS	12

新建(W)...

编辑(I)...

删除(L)

确定

取消

编辑环境变量



C:\Program Files\NVIDIA Corporation\NVIDIA NvDLISR
C:\Program Files\dotnet\
C:\Program Files\Intel\WiFi\bin\
C:\Program Files\Common Files\Intel\WirelessCommon\
C:\mingw64\bin
C:\Program Files\Common Files\Autodesk Shared\
C:\Program Files\Microsoft SQL Server\120\Tools\Binn\
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\170\T...
C:\Program Files (x86)\Microsoft SQL Server\150\Tools\Binn\
C:\Program Files\Microsoft SQL Server\150\Tools\Binn\
C:\Program Files\Microsoft SQL Server\150\DTS\Binn\
C:\Program Files (x86)\Microsoft SQL Server\150\DTS\Binn\
E:\putty\
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
%JAVA_HOME%\bin
%CLASSPATH%
D:\GIT\Git\cmd

新建(N)

编辑(E)

浏览(B)...

删除(D)

上移(U)

下移(O)

编辑文本(T)...

确定

取消

命令提示符

用法: javac <options> <source files>
其中, 可能的选项包括:

-g	生成所有调试信息
-g:none	不生成任何调试信息
-g:{lines, vars, source}	只生成某些调试信息
-nowarn	不生成任何警告
-verbose	输出有关编译器正在执行的操作的消息
-deprecation	输出使用已过时的 API 的源位置
-classpath <路径>	指定查找用户类文件和注释处理程序的位置
-cp <路径>	指定查找用户类文件和注释处理程序的位置
-sourcepath <路径>	指定查找输入源文件的位置
-bootclasspath <路径>	覆盖引导类文件的位置
-extdirs <目录>	覆盖所安装扩展的位置
-endorseddirs <目录>	覆盖签名的标准路径的位置
-proc:{none, only}	控制是否执行注释处理和/或编译。
-processor <class1>[, <class2>, <class3>...]	要运行的注释处理程序的名称; 绕过默认的搜索进程
-processorpath <路径>	指定查找注释处理程序的位置
-parameters	生成元数据以用于方法参数的反射
-d <目录>	指定放置生成的类文件的位置
-s <目录>	指定放置生成的源文件的位置
-h <目录>	指定放置生成的本机标头文件的位置
-implicit:{none, class}	指定是否为隐式引用文件生成类文件
-encoding <编码>	指定源文件使用的字符编码
-source <发行版>	提供与指定发行版的源兼容性
-target <发行版>	生成特定 VM 版本的类文件
-profile <配置文件>	请确保使用的 API 在指定的配置文件中可用
-version	版本信息
-help	输出标准选项的提要
-A关键字[=值]	传递给注释处理程序的选项
-X	输出非标准选项的提要

3. 在记事本环境下编写一个 HelloWorld.java 程序, 在 DOS 提示下编译并执行这个程序

```
Microsoft Windows [版本 10.0.18363.900]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\DELL>cd Desktop

C:\Users\DELL\Desktop>javac lmy.java

C:\Users\DELL\Desktop>java lmy
helloworld 2018214937

C:\Users\DELL\Desktop>
```

4. 编写一个类 A, 它位于包 a 中, 保存到 A.java 中, 再编写一个类 B, 它位于包 b 中, 保存到 B.java 中。在类 B 中实例化类 A 的一个对象。分别编译类 A 和类 B。

```
1 package a;
2
3 public class A {
4     int x;
5     int y;
6
7     public A(int x,int y){
8         this.x=x;
9         this.y=y;
10    }
11
12 }
13
```

```

1 package b;
2 import a.*;
3
4 public class B extends A{
5     B(int x,int y){
6         super(x,y);
7     }
8
9     public static void main(String args[]) {
10         A a=new A(1,2);
11     }
12
13 }
14

```

5. 分别编写两个类 Point2D, Point3D 来表示二维空间和三维空间的点, 使之满足下列要求:

- 1) Point2D 有两个整型成员变量 x, y (分别为二维空间的 X, Y 方向坐标), Point2D 的构造方法要实现对成员变量 x, y 的初始化。
- 2) Point2D 有一个 void 型成员方法 offset(int a, int b), 它可以实现 Point2D 的平移。
- 3) Point3D 是 Point2D 的直接子类, 它有三个整型成员变量 x, y, z (分别为三维空间的 X, Y, Z 方向坐标), Point3D 有两个构造方法: Point3D(int x, int y, int z) 和 Point3D(Point2D p, int z), 两者均可实现对 Point3D 的成员变量 x, y, z 的初始化。
- 4) Point3D 有一个 void 型成员方法 offset(int a, int b, int c), 该方法可以实现 Point3D 的平移。

5) 在 Point3D 中的主函数 main() 中实例化两个 Point2D 的对象 p2d1, p2d2, 打印出它们之间的距离, 再实例化两个 Point2D 的对象 p3d1, p3d2, 打印出他们之间的距离

```

1 package point;
2
3 public class Point2D {
4     int x,y;
5
6     Point2D(int x,int y){
7         this.x=x;
8         this.y=y;
9     }
10
11     void offset(int a,int b) {
12         x=x+a;
13         y=y+b;
14     }
15
16 }
17

```



```

package point;

public class Point3D extends Point2D{

    int z;

    Point3D(int x, int y,int z) {
        super(x, y);
        this.x=x;
        this.y=y;
        this.z=z;
    }

    void offset(int a,int b,int c){
        x=x+a;
        y=y+b;
        z=z+c;
    }

    public static void main(String args[]) {

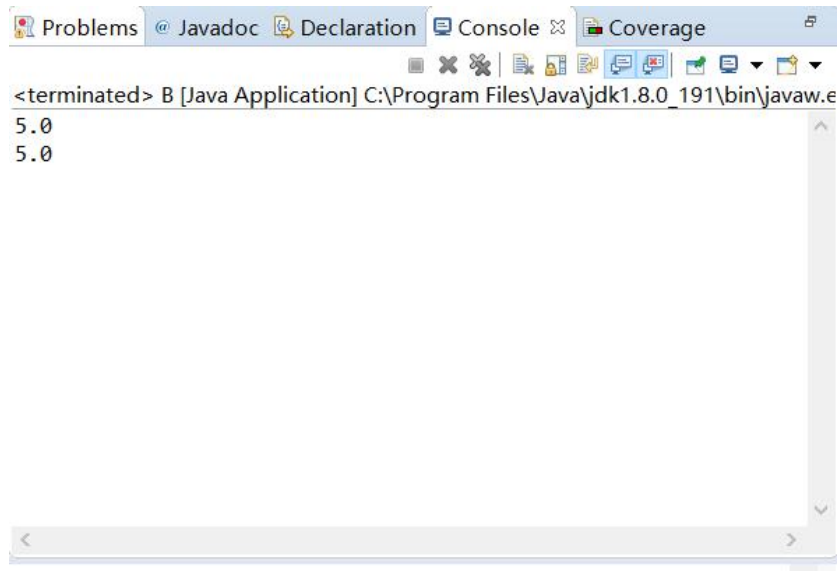
        Point2D p2d1=new Point2D(3,5);
        Point2D p2d2=new Point2D(6,9);
        System.out.println(java.lang.Math.sqrt(((double)((p2d1.x-p2d2.x)*(p2d1.x-p2d2.x)+(p2d1.y-p2d2.y)*(p2d1.y-p2d2.y))));

        Point3D p3d1=new Point3D(1,2,3);
        Point3D p3d2=new Point3D(1,5,7);
        System.out.println(java.lang.Math.sqrt(((double)((p3d1.x-p3d2.x)*(p3d1.x-p3d2.x)+(p3d1.y-p3d2.y)*(p3d1.y-p3d2.y)
        +(p3d1.z-p3d2.z)*(p3d1.z-p3d2.z))));

    }
}

```

运行结果：



五、实验结论、分析、思考题与心得体会

本次实验的收获体会、认识、体会：

1. 此次实验学会编写第一个 Java 程序，在使用文档情况下进行编写，还学会了在 eclipse 中编写 Java 代码，认识了程序的运行方法
2. 学会相关 Java 的基本语法，进行编写程序，在编写 Java 代码时候总是把 System 打成 system 导致编译报错，不熟悉 Java 的基本语法。
3. 了解的 Java 的虚拟机，有了 jvm 可以到多个平台上进行运行代码
4. 学会自己配置 Java 的 path 和 classpath

思考题：

什么是 JVM，它的作用是什么？

JVM 是 java 虚拟机，Java 虚拟机是一个抽象的计算机，和实际的计算机一样，它具有一个指令集并使用不同的存储区域。

它负责执行指令，还要管理数据、内存和寄存器。Java 解释器负责将字节代码翻译成特定机器的机器代码。

如何理解 PATH 和 CLASSPATH, , 如何设置它们

path 是 windows 使用的系统环境变量, 其作用是记录一些可运行的软件 (例如 java.exe, javac.exe) 所在的位置, 使得当我们可以方便快捷的打开这些软件。

classpath 与 path 功能差不多, path 指定的是 windows 上可运行的软件 (一半 xx.exe) 的位置, classpath 值 java 专用的, 它指定的就是 java 中运行的文件(xx.class)的位置