

# 《Java 技术》实验报告

## 实验一

### 一、实验目的

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。

### 二、实验原理

1. 不知道写什么.....

### 三、使用硬件、软件环境

1. 联想小新 Pro 13 笔记本一台。
2. 使用启用了 Windows Subsystem for Linux（以下简称 WSL）的 Windows 10 操作系统。
3. 使用 Visual Studio Code（以下简称 VS Code）作为编辑器。
4. 使用 OpenJDK 11 作为 Java 开发环境。

### 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. 打开 WSL，输入命令：

```
sudo apt install default-jdk
```

安装 Java 开发环境。由于使用了包管理器，这里无需自行配置 PATH 路径。

2. 在 WSL 中输入命令 `code .` 即可在当前目录打开 VS Code，后配置 VS Code 的拓展以便更方便的使用 VS Code。使用 VS Code 创建 `HelloWorld.java`，按下 `ctrl + `` 打开终端，输入命令以下命令以运行：

```
javac HelloWorld.java  
java HelloWorld
```

3. 所在文件夹目录树如下所示：

```
.  
├── A  
│   └── A.java  
└── B  
    └── B.java
```

`A.java` 的源代码如下所示：

```
package A;

public class A {
    public A() {
        System.out.println("A Initialized");
    }

    public String toString() {
        return "This is A";
    }
}
```

B.java 的源代码如下所示：

```
package B;

import A.A;

public class B {
    public static void main(String[] args) {
        A aInstance = new A();
        System.out.println(aInstance.toString());
        System.out.println("This is B");
    }
}
```

在当前目录输入命令：

```
javac B/B.java
```

此时，A.java 也将被编译。在当前目录输入命令 `java B.B` 即可执行，执行结果应如下：

```
A Initialized
This is A
This is B
```

4. 于当前文件夹下创建文件夹 `Point` 并创建 `Point2D.java` 和 `Point3D.java` 文件，源码如下：

`Point2D.java`：

```
package Point;

public class Point2D {
    public int x, y;

    public Point2D(int startX, int startY) {
        x = startX;
        y = startY;
    }

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

    public double caculatedD(Point2D dest) {
        double squareDX = Math.pow(x - dest.x, 2);
        double squareDY = Math.pow(y - dest.y, 2);
        double D = Math.sqrt(squareDX + squareDY);
        return D;
    }
}
```

```
}
```

Point3D.java :

```
package Point;

public class Point3D extends Point2D {
    public int z;

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

    public Point3D(Point2D p, int z) {
        super(p.x, p.y);
        this.z = z;
    }

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

    public double caculated(Point3D dest) {
        double squareDX = Math.pow(x - dest.x, 2);
        double squareDY = Math.pow(y - dest.y, 2);
        double squareDZ = Math.pow(z - dest.z, 2);
        double D = Math.sqrt(squareDX + squareDY + squareDZ);
        return D;
    }

    public static void main(String[] args) {
        Point2D p2d1 = new Point2D(0, 0);
        Point2D p2d2 = new Point2D(3, 4);
        System.out.println("Distance 2D: " + p2d1.caculated(p2d2));
        Point3D p3d1 = new Point3D(p2d1, 0);
        Point3D p3d2 = new Point3D(p2d2, 0);
        System.out.println("Distance 3D: " + p3d1.caculated(p3d2));
    }
}
```

编译并运行，应有如下结果：

```
Distance 2D: 5.0
Distance 3D: 5.0
```

## 五、实验结论、分析、思考题与心得体会

1. 没什么好说的.....