1、访问修饰符有几种？他们的作用范围是什么？

private：被private修饰的属性和方法，不能被其他类访问，子类不能继承也不能访问。只能在所在类内部访问。

缺省：变量或者方法前没有访问修饰符时，可以被所在类访问，可以被同一包内的其他类访问 或者继承。但是不能被其他包访问。

protected：被protected修饰的方法和属性，在同一包内可被访问和继承。不同包内，子类可继承，非子类不能访问。

public：方法和属性前有public修饰，可以被任意包内的类访问。

另外，类要想被其他包导入，必须声明为public。被public修饰的类，类名必须与文件名相同。

2、jvm由哪几部分组成

JVM 的结构基本上由 4 部分组成：

类加载器，在 JVM 启动时或者类运行时将需要的 class 加载到 JVM 中；

执行引擎，执行引擎的任务是负责执行 class 文件中包含的字节码指令，相当于实际机器上的 CPU；

内存区，将内存划分成若干个区以模拟实际机器上的存储、记录和调度功能模块，如实际机器上的各种功能的寄存器或者 PC 指针的记录器等；

本地方法调用，调用 C 或 C++ 实现的本地方法的代码返回结果。

3、要求：

1）定义一个交通工具Vehicle类，类中包含：

属性：速度(speed)、体积(size)

方法：加速speedUp(), 减速speedDown()，方法中控制台输出字符串“加速”/“减速”即可

1. 编写测试类Main, main方法中实例化Vehicle对象，为所有属性赋值并输出，最后调用加速，减速方法。

package student;

public class Vehicle {

double speed;

double size;

public Vehicle(double speed,double size){

this.speed=speed;

this.size=size;

}

public void speedUp(){

System.out.println(”加速”);

}

public void speedDown(){

System.out.println(”减速”);

}

public static void main(String[] args) {

Vehicle Bike=new Vehicle(“Bike”, 50.5, 20.3);

Bike.speedDown();

Bike.speedUp();

}

}

4、要求：

编写Java程序用于显示人的姓名和年龄。

1）定义一个Person类，该类中应该有两个私有属性，姓名(name)和年龄(age)，以及性别(gender)。

2）定义不同参数列表的构造方法并初始化，控制台打印输出。

3）编写测试类Main,在main方法中根据不同重载的构造方法，分别创建Person类的实例

Public class people{

  private string name;

  private int age;

  public string getname() {

   return name;

  }

  public void setname(string name) {

   this.name = name;

  }

  public int getage() {

   return age;

  }

  public void setage(int age) {

   this.age = age;

  }

  public people(string name, int age) {

   super();

   this.name = name;

   this.age = age;

  }

  public people() {

   super();

  }

 @override

 public string tostring() {

   return "people [name=" + name + ", age=" + age + "]";

 }

}

5、使用文件流复制myfile.txt文件为myfile\_cp.txt

package test;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.IOException;

public class CopyFileDemo2 {

public static void copyFile(String src,String dest) throws IOException{

FileInputStream in=new FileInputStream(src);

File file=new File(dest);

if(!file.exists())

file.createNewFile();

FileOutputStream out=new FileOutputStream(file);

int c;

byte buffer[]=new byte[1024];

while((c=in.read(buffer))!=-1){

for(int i=0;i<c;i++)

out.write(buffer[i]);

}

in.close();

out.close();

}

public static void main(String[] args) throws IOException {

String fileSrc="C:/JMPX/myfile.txt";

String fileDes="C:/JMPX/Copy/myfile\_cp.txt";

copyFile(fileSrc,fileDes);

System.out.println("C:/JMPX/Copy文件夹复制成功);

}

}

6、使用缓冲流复制myfile.txt文件为myfile\_cp2.txt

package test;

import java.io.BufferedInputStream;

import java.io.BufferedOutputStream;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

public class CopyDemo2 {

public static void main(String[] args) throws IOException {

FileInputStream fis=new FileInputStream("myfile.txt");

BufferedInputStream bis=new BufferedInputStream(fis);

FileOutputStream fos=new FileOutputStream("myfile\_cp2.txt");

BufferedOutputStream bos=new BufferedOutputStream(fos);

//byte[] data=new byte[1024\*10];

int len=-1;

while((len=bis.read())!=-1){

bos.write(len);

}

System.out.println("复制完毕");

bis.close();

bos.close();

}

}