**《面向对象Java编程》上机作业4**

|  |  |
| --- | --- |
| **学号：** |  |
| **姓名：** |  |
| **班级：** |  |
| **日期：** |  |

***\*样例见附1***

**1.** Define an interface, called **VolumeArea**, within which there’re:

* a static and final variable PI (with the value 3.14159), and
* two abstract methods volume(double radius) and area(double radius).

Write another class **MyCircle** which implements the interface **VolumeArea**. Overriding the two methods:

* volume(double radius) to return a value zero (as the volume of a circle is 0), and
* area(double radius) to return the area of the circle.

Write the third class **MySphere**, also implements **VolumeArea**. Overriding the two methods:

* volume(double radius) to calculate and return the volume of the sphere (using (4π\*r\*r\*r)/3);
* area(double radius) to calculate and return the surface area of the sphere (using 4π\*r\*r).

Write a main class with the main method to test the methods in **MyCircle** and **MySphere**, set the radius of the circle and sphere both to 3.5. And display the results.

|  |
| --- |
| * **问题分析** |
| * **代码实现** |
| * **运行结果** |
| * **思考及总结** |

**2.** Define an abstract class named **Employee**. Define following classes to inherit the abstract class,

a) **Boss** class, indicate the boss who get paid with a fixed salary without considering working hours;

b) **CommissionWork** class, indicate the workers who some basic salary and floating wages according to sales;

c) **PieceWorker** class, indicate those who are paid according to the numbers of products;

d) **HourlyWorker** class, indicate those get paid according to their working hours.

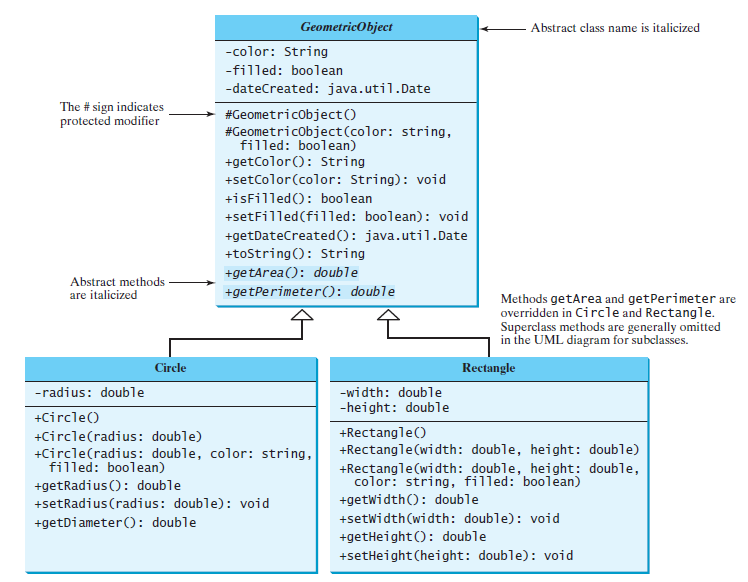
Define above four classes, to implement the abstract methods declared in the abstract class:

* getPaid() method without any parameter but have a return value for total salary, and,
* display() method for output the information about the employee.

And define another class with main method (also called test class or driven class), to test the methods defined.

|  |
| --- |
| * **问题分析** |
| * **代码实现** |
| * **运行结果** |
| * **思考及总结** |

**3.** 请根据以下UML图，编写相关的三个类：



另外定义一个主类，main方法中对上图中设计的各个方法进行测试。

TIPs: ’+’符号表示public修饰的属性/方法；’-‘表示private修饰；’#’表示protected修饰；斜体表示abstract修饰的类/方法。

|  |
| --- |
| * **问题分析** |
| * **代码实现** |
| * **运行结果** |
| * **思考及总结** |

**4.** (*Math: The* **Complex** *class*) A complex number is a number of the form where *a* and *b* are real numbers and *i* is The numbers **a** and **b** are known as the real part and imaginary part of the complex number, respectively. You can perform addition, subtraction, multiplication, and division for complex numbers using the following formula:

文本, 信件

描述已自动生成

You can also obtain the absolute value for a complex number using the following formula:



Design a class named **Complex** for representing complex numbers and the methods **add**, **subtract**, **multiply**, **divide**, and **abs** for performing complex-number operations, and override the **toString** method for returning a string representation for a complex number. The **toString** method returns **a + bi** as a string. If **b** is **0**, it simply returns **a**.

Provide three constructors **Complex(a, b)**, **Complex(a)**, and **Complex()**. **Complex()** creates a **Complex** object for number **0** and **Complex(a)** creates a **Complex** object with **0** for **b**. Also provide the **getRealPart()** and **getImaginaryPart()** methods for returning the real and imaginary part of the complex number, respectively.

Write a test program that prompts the user to enter two complex numbers and display the result of their addition, subtraction, multiplication, and division. Here is a sample run:

|  |
| --- |
| Enter the first complex number: 3.5 5.5  Enter the second complex number: -3.5 1  3.5 + 5.5i + -3.5 + 1.0i = 0.0 + 6.5i  3.5 + 5.5i - -3.5 + 1.0i = 7.0 + 4.5i  3.5 + 5.5i \* -3.5 + 1.0i = -17.75 + -15.75i  3.5 + 5.5i / -3.5 + 1.0i = -0.5094 + -1.7i  |3.5 + 5.5i| = 6.519202405202649 |

|  |
| --- |
| * **问题分析** |
| * **代码实现** |
| * **运行结果** |
| * **思考及总结** |

**注1：**

* **独立完成！切勿抄袭！**
* **提交的文档请重命名为“*学号*\_*姓名*\_*上机n”*命名，如“*202112340001\_徐利锋\_上机n.doc”*).**
* **截止日期：4月17日 20:00**

**注2：作业样例（见后）**

**附1.作业样例：**

编写一个程序，打印100～200之间的素数，要求每行按10个数(数与数之间有一个空格间隔)的形式对其输出。

(***以下为本题的解题样例，后续题目请参考样例解答***)

|  |
| --- |
| * **问题分析**   用循环遍历101-200之间的数，通过筛选法判断每个数是否为素数，如是则按题目要求格式输出。  筛选法求素数的大致思路：一个数n分别除以2、3…根号n，若全部不能整除，则为素数；反之则不是素数。  题目要求格式的满足：每行10个数，需要对每次输出进行计数，逢10换行。不换行的时候则输出空格。  根号运算需要调用Math类里的sqrt方法，如：Math.*sqrt*(*n*)表示n的平方根。 |
| * **代码实现**   **public** **class** Prime {  **public** **static** **void** main(String[] args) {  **int** counter = 0;  **for**(**int** i=101; i<=200; i++) {  **if**(*isPrime*(i)) {  System.***out***.print(i);  counter++;  **if**(counter % 10 == 0) {  System.***out***.println();  } **else** {  System.***out***.print(" ");  }  }  }  }  **public** **static** **boolean** isPrime(**int** num) {  **boolean** result = **true**;  **for**(**int** i=2; i<=Math.*sqrt*(num); i++) {  **if**(num % i == 0) {  result = **false**;  **break**;  }  }  **return** result;  }  } |
| * **运行结果**   运行结果如图1-1所示。    图1-1 问题1程序运行结果 |
| * **思考及总结**   学习到了以下几个用法。（以上述代码为例）  1、其他方法的定义： **public** **static** **boolean** isPrime(**int** num)  表示声明一个在main方法中可以直接调用的方法（因有static修饰，表示静态方法），需要一个int类型的参数，返回值为boolean类型。  2、for循环的使用: **for**(**int** i=101; i<=200; i++)  循环时注意循环变量的取值边界及累进值  **break**;  表示退出当前循环。  3、Math类中开根方法的调用：Math.*sqrt*(*n*)  4、输出：换行及空格的输出：  System.***out***.println();  System.***out***.print(" "); |