

Chapter 2. Basic Object Oriented Programming Concepts

I. Exercises with solutions

1. Study and use class Math (java.lang.Math)

Test at least 10 methods of the class Math, such as sin, cos, sqrt, ceil, round, min, max,....

```
public class Exercise1 {
    public static void main(String[] args) {
        int a =10, b =20 , c=30;
        int max;

        max = Math.max(a, b);
        System.out.println("Max of a, b is " + max);

        max = Math.max(max, c);
        System.out.println("Max of a, b, c is " + max);

        //put your code for other methods from here
        //.....
    }
}
```

2. Study and use class String (java.lang.String). Test at least 10 methods of the class String.

```
public class Exercise2 {
    public static void main(String[] args) {
        String str1 = new String("VKU is one of the best
                                universities in Vietnam");
        String str2 = new String("Prof. Phap is an OOP
                                teacher");

        System.out.println("str1 contains "+
                            str1.length()+" character");

        int n=str1.compareTo(str2);
        if (n>0)
            System.out.println("str1 is greater than
                                str2");

        else
        if (n==0)
            System.out.println("str1 is equal to str2");
        else System.out.println("str1 is less than str2");

        //put your code for other methods from here
        //.....
    }
}
```

```
}  
}
```

3. Study and use class Calendar (java.util.Calendar)

```
import java.util.*;  
public class Exercise3 {  
    public static void main(String[] args) {  
  
        Calendar rightNow = Calendar.getInstance();  
        System.out.println(rightNow.getTime());  
  
        System.out.println(rightNow.getTimeZone());  
  
        System.out.println(rightNow.getWeekYear());  
  
        //put your code for other methods from here  
        //.....  
  
    }  
}
```

4. Study and use class Date (java.util.Date)

```
import java.util.*;  
  
public class Exercise4  
{  
    public static void main(String[] args)  
    {  
        Date d1 = new Date();  
        System.out.println("Current date is " + d1);  
        System.out.println("Current hour is  
            "+d1.getHours());  
  
        //put your code for other methods from here  
        //.....  
  
    }  
}
```

5. Study and use class LocalDate (java.time.LocalDate)

```
import java.time.LocalDate; // import the LocalDate class

public class Exercise4 {
    public static void main(String[] args) {
        LocalDate myObj = LocalDate.now();
        System.out.println(myObj);
        //put your code for other methods from here
        //.....
    }
}
```

6. Specify and implement an OOP program to calculate the area of a rectangle.

+ Specification:

class Rectangle
properties: double width, height
method: area() width * height

+ Implementation:

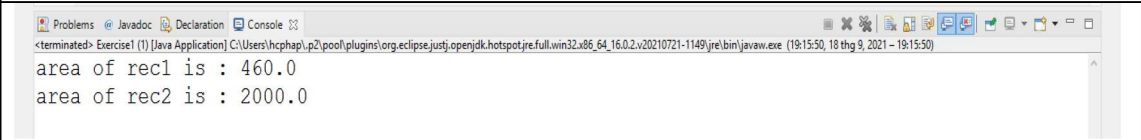
```
class Rectangle
{
    private double width, height;
    Rectangle(double widVal,double heightVal)
    {
        width = widVal;
        height = heightVal;
    }
    double area()
    {
        return width * height;
    }
}

public class Exercise6
{
    public static void main(String args[])
    {
        Rectangle rec1, rec2;
```

```

rec1 = new Rectangle(23,20);
rec2 = new Rectangle(40,50);
System.out.println("area of rec1 is : " +
                    rec1.area());
System.out.println("area of rec2 is : " +
                    rec2.area());
    }
}

```



7. Specify and implement an OOP program to calculate the cube volume.

+ Specification:

class Cube
properties: double size
methods: + setSize(val) size=val + getSize() get size value + volume() size * size* size + details() display detailed infos

+ Implementation:

```

class Cube
{
    private double size;
    Cube(double val)
    {
        size = val;
    }

    void setSize(double val)
    {
        size = val;
    }
}

```

```
double getSize()
{
    return size;
}
double volume()
{
    return size*size*size;
}
void details()
{
    System.out.println("\ndetails of rectangle");
    System.out.println("width="+size);
    System.out.println("height="+size);
    System.out.println("length="+size);
    System.out.println("volume="+volume()+"\n\n");
}
}
public class Exercise7
{
    public static void main(String args[])
    {
        Cube cube1, cube2;
        cube1 = new Cube(4);
        cube2 = new Cube(3);

        System.out.println("cube1's size= "+cube1.getSize());
        System.out.println("cube2's size= "+cube2.getSize());

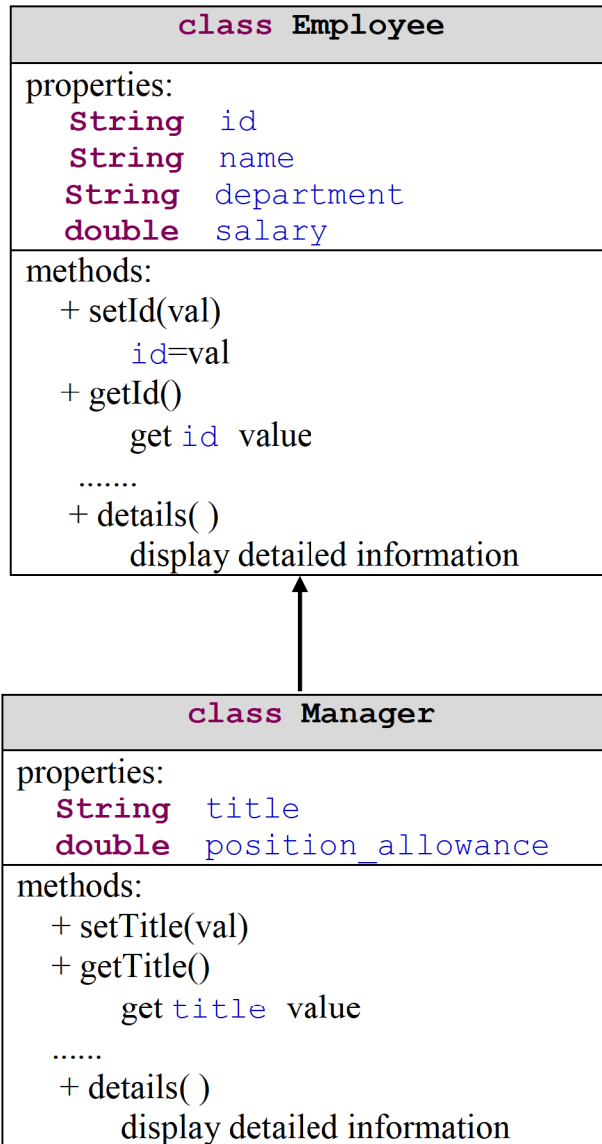
        cube1.setSize(10);
        cube2.setSize(20);

        cube1.details();
        cube2.details();
    }
}
```



- Specify and implement an OOP program to create a class Employee and a class Manager inheriting from the class Employee.

+ Specification:



+ Implementation:

```
class Employee
{
    String    id;
    String    name;
    String    department;
    double    salary;
    public Employee(String id, String name, String dep,
                    double sal)
    {
        this.id=id;
        this.name=name;
        department=dep;
        salary =sal;
    }
    public void setId(String val)
    {
        id=val;
    }
    public String getId()
    {
        return id;
    }
    //complete code for getX, setX of other properties
    public void details()
    {
        System.out.println("ID: "+id);
        System.out.println("Name: "+name);
        System.out.println("Department: "+department);
        System.out.println("Salary: "+salary);
    }
}

class Manager extends Employee
{
    String    title;
    double    position_allowance;

    public Manager(String id, String name, String dep,
                    double sal, String tit, double pa)
    {
        super(id,name,dep,sal);
        title=tit;
        position_allowance=pa;
    }
    public void setTitle(String val)
    {
        title=val;
    }
}
```



```

public String getTitle()
{
    return title;
}
//complete code for getX, setX of other properties
public void details()
{
    super.details();
    System.out.println("Title: "+title);
    System.out.println("Position allowance: `
                        + position_allowance);
}
}
public class Exercise8 {

public static void main(String[] args) {
    Employee A = new Employee("VKU01",
        "David Tho", "Academic Office", 10000);
    Manager B = new Manager("VKU02",
        "Jonh Phap", "Steering Board", 10000, "Rector", 1000);
    A.details();
    System.out.println("-----");
    B.details();
}
}

```

```

ID: VKU01
Name: David Tho
Department: Academic Office
Salary: 10000.0
-----|
ID: VKU02
Name: Jonh Phap
Department: Steering Board
Salary: 10000.0
Title: Rector
Position allowance: 1000.0

```

II. Do it yourself

1. Study and use class Character (java.lang. Character). Test at least 10 methods of this class.
2. Study and use class StringBuffer (java.lang. StringBuffer). Test at least 10 methods of this class
3. Study and use class Random (java.util. Random). Test at least 10 methods of this class

Hint: import java.util. Random;

4. Study and use other classes in packages java.lang and java.util. For each class, test at least 05 methods.

5. Specify and implement an OOP program to calculate average mark and display information of the student.

Hint: student information includes: name, age, address, marks of subjects,...

6. Specify and implement an OOP program to calculate benefit and display information of the company.

Hint: benefit = income – cost

Company information includes name, address, cost, income, benefit...

7. Specify and implement an OOP program to create a class Person and a class Teacher inheriting from the class Person.

Hint: Person (name, age, address)

Teacher(..., institution_name, courses,...)