

Corrigé du TD de synthèse
P.O.O en JAVA

QCM : les réponses justes sont colorées en rouge

1. Java est un langage de programmation développé par : (0,5 Pts)
a. Microsoft b. Oracle c. IBM d. **Sun Microsystems**
2. Le Bytecode est un : (0,5 Pts)
a. Langage intérimaire, dont les instructions sont directement exécutées par le microprocesseur.
b. Langage intérimaire qui s'exécute par la machine virtuelle (JVM) de Java. Cependant, pour assurer la portabilité du langage Java la même JVM doit être installée sur différentes plateformes.
c. **Langage intérimaire qui s'exécute par la JVM et dont les instructions sont codées sur un Byte.**
d. Langage intérimaire qui s'exécute par la JVM et dont les instructions sont codées sur quatre Bytes.
3. Selon la convention de nommage, lesquels des noms suivants peuvent être une constante : (0,5 Pts)
a. Pi b. Max_Size c. **PI** d. **MAX_SIZE**
4. Une classe peut : (0,5 Pts)
a. Implémenter plusieurs classes mais doit étendre une seule classe.
b. Implémenter plusieurs classes et peut étendre plusieurs interfaces.
c. **Implémenter plusieurs interfaces mais doit étendre une seule classe.**
d. Une classe doit implémenter une seule interface et étendre une seule classe.
5. Laquelle des déclarations suivantes est correcte ? (0,5 Pts)
a. boolean b=TRUE ;
b. float a = 0,5 ;
c. **int i = new Integer(5) ;**
d. int i = new Integer(5.2) ;
6. Combien d'instances de la classe A sera créé par le code suivant ? (0,5 Pts)

```
A a,b,c ;  
a=new A() ;  
A d=a ;  
A f= new A() ;  
b=a ;
```

- a. Une **b. Deux** c. Trois d. Quatre

7. Pour la classe B définie comme suit : (0,5 Pts)

```
Classe Test {  
    public static int i ;  
    public int j ;  
    public Test() {i++ ;  
                  j=i ;}
```

qu'affiche le code suivant ?

```
Test x = new Test() ;  
Test y = new Test() ;  
Test z = x ;  
System.out.println(z.i+ "et"+z.j) ;
```

- a. 2 et 2 b. 1 et 3 c. 1 et 1 **d. 2 et 1**

8. Soit le code suivant, quelles sont les instructions qui provoquent une erreur de compilation. (0,5 Pts)

<pre>public class Test { public static void main(String[] args) { m(new EngineerStudent()); m(new Student()); m(new Person()); m(new Object()); } public static void m(Student x) { System.out.println(x.toString()); } }</pre>	<pre>class EngineerStudent extends Student { } class Student extends Person { public String toString() { return "Student"; } } class Person extends Object { public String toString() { return "Person"; } }</pre>
---	--

- m(new EngineerStudent()) provoque une erreur.
- m(new Student()) provoque une erreur.
- m(new Person()) provoque une erreur.
- m(new Object()) provoque une erreur.

9. Considérer l'extrait de code Java suivant. Quel est le résultat obtenu, si vous essayez de compiler et exécuter ce code ? (1 Pts)

<pre>public class Test { private int i; public Test() { } public Test(int i) { this.i = i; } public static void printStatic(Test t){ System.out.print(t.i) ; } public void print(Test t){ System.out.println(t.i) ; } public static void main(String[] args) { Test t1 = new Test(5); Test.printStatic(t1); (new Test(10)).print(t1); } }</pre>	<p>Quel résultat affichera le programme ci-contre ?</p> <ol style="list-style-type: none"> 5 5 5 0 5 10 10 5
---	--

10. Pour les classes A et B définies comme suit: (1 Pts)

<pre>class A { public int x; public A() {x=5; } } }</pre>	<pre>class B extends A { public B() {x++;} public B(int i){this(); x=x+i; } public B(String s){super(); x--; } }</pre>
---	--

qu'affiche le code suivant?

```
B b1=new B();
B b2 =new B(2003);
B b3= new B("Bonjour");
System.out.println(b1.x + " et " + b2.x + " et encore " + b3.x );
```

- 6 et 2009 et encore 4
- 1 et 2004 et encore 4
- 1 et 2004 et encore 2003
- Autre chose.

Exercice 1 :

L'interface IOperation :

```
public interface IOperation {  
    public Object plus (Object obj);  
    public Object moins (Object obj);  
}
```

La classe Affichage :

```
public abstract class Affichage {  
    public abstract String affiche();  
}
```

La classe Complexe :

```
public class Complexe extends Affichage implements IOperation {  
  
    private double im ;  
    private double re;  
  
    public Complexe (double re , double im){  
        this.re=re;  
        this.im=im;  
    }  
  
    public Object plus(Object c){  
        Complexe com = (Complexe) c;  
        return new Complexe (this.re + com.re, this.im + com.im);  
    }  
  
    public Object moins (Object c){  
        Complexe com = (Complexe) c;  
        return new Complexe (this.re - com.re, this.im - com.im);  
    }  
  
    public String affiche() {  
        if(this.im > 0) return this.re + " + "+this.im+" i";  
        else if(this.im < 0) return this.re + " "+this.im+" i";  
        else return this.re+"";  
    }  
}
```

La classe Reel :

```
public class Reel extends Affichage implements IOperation{  
    private double x;  
  
    public Reel(double x){  
        this.x=x;  
    }  
  
    public Object moins(Object obj) {  
        Reel reel = (Reel) obj;  
        return new Reel(this.x - reel.x);  
    }  
  
    public Object plus(Object obj) {  
        Reel reel = (Reel) obj;  
        return new Reel(this.x + reel.x);  
    }  
  
    public String affiche() {  
        return x+"";  
    }  
}
```

La classe Test :

```
public class Test {  
    public static void main (String[] args){  
  
        Complexe c1 = new Complexe(2,2) ;  
        Complexe c2 = new Complexe(-3,4) ;  
        Complexe c3,c4 ;  
        Reel r1=new Reel(3.4);  
        Reel r2=new Reel(3.89);  
        Reel r3;  
        c3 =(Complexe) (c1.plus(c2));  
        c4=(Complexe) (c1.moins(c2));  
        System.out.println("c2 + c1 = " +c3.affiche());  
        System.out.println("c2 - c1 = " +c4.affiche());  
        r3=(Reel) (r1.plus(r2));  
        System.out.println("r1 + r2 = " +r3.affiche());  
    }  
}
```

Exercice 2 :

1.

```
public class InvalidAgeException extends Exception{
    private int anormalAge;
    public InvalidAgeException(String message, int anormalAge)
    { super(message);
      this.anormalAge=anormalAge;}
    // Vous pouvez ajouter une méthode getter ;
}
```

2.

```
public class InvalidMoyenneException extends Exception{

    private double anormalMoyenne;
    public InvalidMoyenneException(String message, double anormalMoyenne)
    { super(message);
      this.anormalMoyenne=anormalMoyenne;
    }
    // Vous pouvez ajouter une méthode getter ;
}
```

3.

```
public class Student {
    private String code;
    private int age;
    private double moyenne;
    public Student (String code, int age, double moyenne) throws
InvalidAgeException,InvalidMoyenneException
    {      this.code=code;
          if (age<18 || age>26)
              throw new InvalidAgeException("L'âge que vous avez saisi doit
être entre 18 et 26",age);
          this.age=age;
          if (moyenne<0 || moyenne >20)
              throw new InvalidMoyenneException("La moyenne que vous avez saisi
doit être entre 0 et 20", moyenne);
          this.moyenne=moyenne;
    }}
}
```

4.

```
import java.util.Scanner;
import java.util.InputMismatchException;
public class Main {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner clavier = new Scanner(System.in);
        boolean test;

        do {
            test=true;

            try {
                System.out.println("Donner le code de l'étudiant");
                String nom = clavier.nextLine();
                System.out.println("Donner l'âge de l'étudiant");
                int age = clavier.nextInt();
                System.out.println("Donner la moyenne de l'étudiant");
                double moyenne = clavier.nextDouble();
                Student S1= new Student(nom,age,moyenne);
                System.out.println("Opération effectuée");
            }
            catch(InvalidAgeException e) {
                System.out.println(e.getMessage());
                test=false;
            }
            catch(InvalidMoyenneException e) {
                System.out.println(e.getMessage());
                test=false;
            }
            catch(InputMismatchException e) {
                System.out.println("Veuillez saisir une valeur entière pour
l'âge et une valeur réel pour la moyenne");
                test=false;
            }
        }
        finally
        {
            clavier.nextLine(); //vider le buffer du clavier
        }

    }while(!test);
}
}
```