TD de synthèse : P.O.O en JAVA

QCM

1. Java est un langage de programmation développé par :
   a. Microsoft b. Oracle c. IBM d. Sun Microsystems
2. Le Bytecode est un :
   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 :
   a. Pi b. Max_Size c. PI d. MAX_SIZE
4. Une classe peut :
   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 ?
   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 ?

```
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 :

```
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.

```
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());
  }}
```

```
class EngineerStudent extends Student {
}
class Student extends Person {
  public String toString() {
    return "Student";
  }
}
class Person extends Object {
  public String toString() {
    return "Person";
  }}
```

a. m(new EngineerStudent()) provoque une erreur.
b. m(new Student()) provoque une erreur.
c. m(new Person()) provoque une erreur.
d. 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 ?

```
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); }}
```

Quel résultat affichera le programme ci-contre ?

a. 5 5
b. 5 0
c. 5 10
d. 10 5

10. Pour les classes A et B définies comme suit:

```
class A {
public int x;
public A() {x=5; } } }
```

```
class B extends A {
public B() {x++;}
public B(int i){this();
                x=x+i; }
public B(String s){super();
                   x--; }}
```

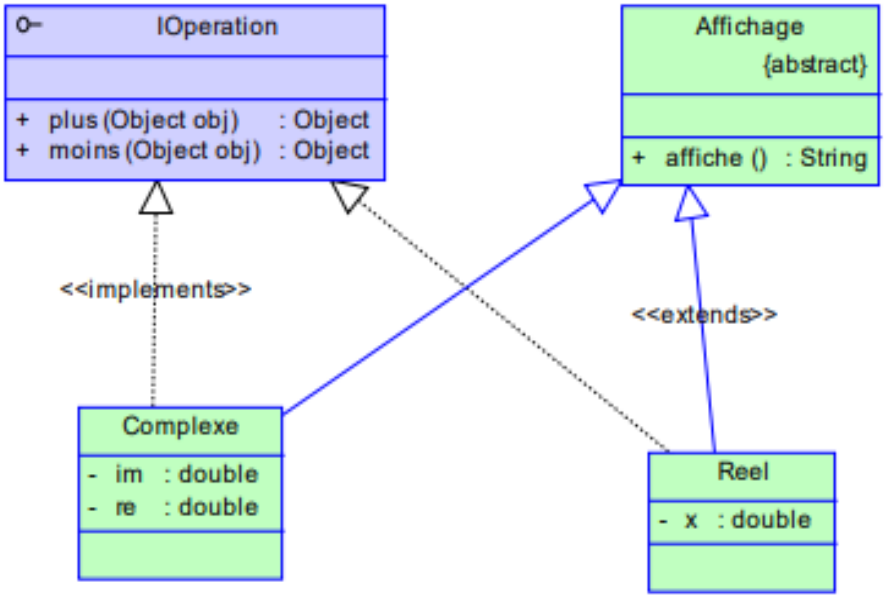
qu'affiche le code suivant?

```
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 );
```

a. 6 et 2009 et encore 4
b. 1 et 2004 et encore 4
c. 1 et 2004 et encore 2003
d. Autre chose

Exercice 1

Soit le diagramme de classe suivant



- **IOperation** : interface.
- **Affichage** : classe abstraite.
- **im** : partie imaginaire d'un nombre complexe.
- **re** : partie réelle d'un nombre complexe.
- **plus** : méthode pour faire l'addition.
- **moins** : méthode pour faire la soustraction.
- **affiche** : méthode abstraite pour l'affichage d'un nombre.

Questions :

1. Ecrire l'interface IOperation.
2. Ecrire la classe abstraite Affichage.
3. Ecrire la classe Complexe. (Penser au transtypage ou casting afin d'implémenter les méthodes de l'interface IOperation).
4. Ecrire la classe Reel. (Penser au transtypage ou casting afin d'implémenter les méthodes de l'interface IOperation).
5. Ecrire un programme principal qui permet de :
   a. Calculer la somme et la soustraction de deux nombres complexes et d'afficher les résultats.
   b. Calculer la somme et la soustraction de deux nombres réels et d'afficher les résultats.

Exercice 2

On souhaite réaliser une classe Student caractérisée par : code (String), âge (int), moyenne (double) et un seul constructeur avec 3 paramètres. Les attributs de cette classe possèdent une visibilité privée (private). L'instanciation d'un objet qui fait appel au constructeur de la classe Student nécessite la gestion des exceptions suivantes :

- L'âge doit être entre 18 et 26 sinon l'exception **InvalidAgeException** est générée (elle affiche le message "L'âge que vous avez saisi doit être entre 18 et 26") au niveau du constructeur de la classe Student.
- La moyenne doit être entre 0 et 20 sinon l'exception **InvalidMoyenneException** est générée (elle affiche le message "La moyenne que vous avez saisi doit être entre 0 et 20") au niveau du constructeur de la classe Student.

Questions :

1. Ecrire la classe d'exception **InvalidAgeException** avec un champ *anormalAge* destiné à conserver la valeur de l'âge qui a généré l'exception.
2. Ecrire la classe d'exception **InvalidMoyenneException** avec un champ *anormalMoyenne* destiné à conserver la valeur de la moyenne qui a généré l'exception.
3. Ecrire la classe Student.
4. Ecrire un programme principal qui (Penser à lire toutes les étapes avant de le rédiger) :
    a. Demande à l'utilisateur de saisir le code d'un étudiant, son âge et sa moyenne puis créer un objet de type Student et l'initialiser par les informations saisies par cet utilisateur.
    b. Attrape et traite (try ……catch) les deux types d'exception qui peuvent être levés.
    c. Ajoute le traitement de l'exception prédéfinie **InputMisMatchException** en affichant à l'utilisateur le message suivant : "Attention, âge est de type entier et moyenne et de type réel".
    d. Affiche le message de confirmation "Opération bien déroulée" si aucune exception n'est levée lors de la création et l'initialisation d'un objet de la classe Student.
    e. Redonne à l'utilisateur la possibilité de saisir à nouveau les informations si une exception est levée.