

POO: JAVA
TEST BLANC
(DURÉE 1H30)
22.12.2015

Nom :

Prénom :

	Points max	Points
QCM	20	
Exercice	20	
Total	40	

Consignes

1. Les notes du cours **ne sont pas** autorisées.
2. Il est vivement recommandé de lire très attentivement et tranquillement chaque énoncé avant d'y répondre.
3. Il est impératif d'écrire très lisiblement.
4. Le QCM comporte 20 questions. Pour chaque question, une seule des propositions données est correcte. Vous devez mettre une croix correspondante à la réponse choisie sur la fiche des réponses.
(Réponse juste = **1 point** , Réponse fausse = **- 0.5 point** ; Pas de réponse = **0 point** .)

QCM :

1. Quand un programme java est compilé, le fichier généré possède l'extension :
(a) jav (b) exe **(c) class** (d) java
2. Pour exécuter une application java dont le programme source est dans le fichier "Test.java", je dois taper dans l'invite des commandes:
(a) javac Test
(b) javac Test.java
(c) java Test.java
(d) java Test
3. L'interprétation des programmes Java est effectuée par
(a) API (b) JDK **(c) JVM** (d) AWT
4. Quelle classe n'a pas de classe mère ?
(a) Orpheline
(b) String
(c) Object
(d) Une classe abstraite
5. Que va afficher le programme suivant ?

```
Integer nb1 = new Integer (4);  
Integer nb2 = new Integer (4);  
System.out.println (nb1 == nb2);
```


(a) true (b) **false** (c) 4 (d) 1
6. La méthode dont la signature est : `public static double sin(double angle);` se trouve dans la classe Math, comment faire pour calculer le sinus de 1 radian ?
(a) `Math m = new Math(); m.sin (1);`
(b) `new Math().sin (1);`
(c) `Math.sin (1);`
(d) `Math.static.sin (1);`
7. Pour la classe C définie comme suit:

```
public class C {  
    public static int i;  
    public int j;  
    public C() {  
        i++;  
        j++;  
    }  
}
```


qu'affichera le code suivant?

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

- (a) 2 et 2
- (b) 1 et 1
- (c) 2 et 1
- (d) 1 et 3

8. On considère le code:

```
public class A1{
    int i=0;
    public void f(){
        System.out.println(i);
    }
}
```

```
public class B1 extends A1{
    int i=1;
    public void f(){
        System.out.println(i);
    }
}
```

```
public class B2 extends A1{
    int i=2;
    public void f(){
        System.out.println(i);
    }
}
```

Le code suivant: "A1 a1 = new B1(); B2 b2 = (B2)a1; b2.f();"

- (a) provoque une erreur à la compilation
- (b) **provoque une erreur à l'exécution**
- (c) affiche 1
- (d) affiche 2

9. Que va donner le programme suivant ?

```
class A {
    final public int GetResult(int a, int b) {
        return 0;
    }
}
class B extends A {
    public int GetResult(int a, int b) {
        return 1;
    }
}
public class Test {
    public static void main(String args[]) {
        B b = new B();
        System.out.println("x = " + b.GetResult(0, 1));
    }
}
```

- (a) Afficher : x = 0
- (b) Afficher : x = 1
- (c) **Générer une erreur de compilation**
- (d) Lever une exception

10. Soit le code suivant :

```
public void test( boolean a, boolean b){
    if( a ) { System.out.println("A"); }
}
```

```
else if(a && b) { System.out.println( "A && B"); }  
else {  
    if ( !b ) { System.out.println( "notB") ; }  
    else { System.out.println( "ELSE" ) ; }  
}  
}
```

Quelle affirmation est correcte ?

- (a) si a est true et b est true alors "A && B" sera affichée.
- (b) si a est true et b est false alors "notB" sera affichée.
- (c) si a est false et b est true alors "ELSE" sera affichée.
- (d) si a est false et b est false alors "ELSE" sera affichée.

11. Que va afficher le programme suivant ?

```
public class SC2{  
    public void start(){  
        int a = 3;  int b = 4;  
        System.out.print(" " + 7 + 2 + " ");  
        System.out.print(a + b);  
        System.out.print(" " + a + b + " ");  
        System.out.print(foo() + a + b + " ");  
        System.out.println(a + b + foo());  
    }  
    public String foo(){  
        return "foo";  
    }  
    public static void main(String [] args){  
        SC2 s = new SC2();  
        s.start();  
    }  
}
```

- (a) 9 7 7 foo 7 7foo
- (b) 72 34 34 foo34 34foo
- (c) 9 7 7 foo34 34foo
- (d) 72 7 34 foo34 7foo

12. Que sera le résultat du code suivant ?

```
try {  
    int x = 0;  
    int y = 5 / x;  
}  
catch (Exception e){  
    System.out.println("Exception");  
}  
catch (ArithmeticException ae){  
    System.out.println("Arithmetic Exception");  
}  
System.out.println("finished");
```

- (a) finished
- (b) Exception
- (c) Arithmetic Exception
- (d) Erreur de compilation

13. Observez le morceau de code suivant:

```
public class Company{  
    public abstract double calculateSalaries();  
}
```

Quelle affirmation est dès lors correcte?

- (a) Les mots clés abstract et public ne peuvent pas être utilisés ensemble.
- (b) La méthode calculateSalaries dans la classe Company doit avoir un body.
- (c) La méthode calculateSalaries doit avoir void comme type de retour.
- (d) La classe Company doit être définie comme abstraite.

14. Pour les classes A et D définies comme suit :

```
class A {  
    public float f(float x) {  
        return(x+5) ;  
    }  
    public int g(int x) {  
        return (3) ;  
    }  
}
```

```
class D extends A {  
    public int f(int x){  
        return(x+4) ;  
    }  
    public int g( int x) {  
        return (x+8) ;  
    }  
}
```

Que va afficher le code suivant ?

```
D d = new D() ; A a = d ;  
System.out.println(a.f(2)*a.g(3)) ;
```

- (a) 18.0
- (b) 21.0
- (c) 66.0
- (d) 77.0

15. Pour empêcher une classe d'être dérivée, quel est le mot-clé que l'on utilise dans la déclaration de classe? (public ??? class MaClasse {})

- (a) transient
- (b) final
- (c) abstract
- (d) native

16. On considère les déclarations et définitions:

```
interface I{ ... }  
class C implements I{ ... }
```

Parmi les morceaux de codes suivants lequel provoque une erreur à la compilation:

- (a) `I i=new I();`
- (b) `I ic=new C();`
- (c) `C cc=new C();`
- (d) `I ic= new C(); C c= ic;`

17. Soit le code suivant :

```
public interface Base {  
    boolean m1 ();  
    byte m2 (short s);  
}
```

Quel bout de code parmi les suivants ne donnera pas d'erreur à la compilation ?

- (a) `interface Base2 implements Base { }`
- (b) `abstract class Class2 extends Base { public boolean m1(){ return true; } }`
- (c) `abstract class Class2 implements Base { public boolean m1(){ return (7 > 4); } }`
- (d) `abstract class Class2 implements Base { protected boolean m1(){ return (5 > 7) ; } }`

18. Soit le code suivant :

```
class A {  
    protected int method1(int a, int b) {  
        return 0;  
    }  
}
```

Quel bout de code parmi les suivants sera valide dans une classe qui hérite de la classe A ?

- (a) `public int method1(int a, int b) {return 0; }`
- (b) `private int method1(int a, int b) { return 0; }`
- (c) `public short method1(int a, int b) { return 0; }`
- (d) `static protected int method1(int a, int b) { return 0; }`

19. Parmi les suivantes laquelle n'est pas une signature de méthode valide dans un interface?

- (a) `private int getArea();`
- (b) `public float getVol(float x);`
- (c) `boolean setFlag(int test);`
- (d) `public void main(String [] args);`

20. Quelle affirmation en rapport avec les composants GUI est fausse?

- (a) Swing est construit au-dessus de AWT.
- (b) AWT veut dire Abstract Windowing Toolkit.
- (c) Swing utilise encore le modèle de gestion d'évènements de AWT.
- (d) On ne peut pas utiliser AWT et Swing dans la même application.

Exercice :

Dans un établissement d'enseignement supérieur tel que l'ENSAJ, on trouve trois sortes de personnes : des administratifs, des enseignants et des étudiants. Chaque personne est caractérisée par son *nom*, *prénom* et son *adresse* (rue et ville) qui sont des attributs privés et communs à toutes les personnes.

Dans la classe *Personne* on trouve aussi un attribut *nbPersonnes* qui est une variable de classe qui comptabilise le nombre de *Personne* dans l'établissement.

Un *Administratif* est une *Personne*. Il possède toutes les caractéristiques d'une *Personne* (nom, prenom, rue, ville) plus les caractéristiques spécifiques d'un administratif soit un numéro de bureau (*noBureau*). De même, un *Enseignant* est une *Personne* enseignant une *spécialité* (mathématiques, informatique, anglais, gestion, etc.). Un *Etudiant* est une *Personne* préparant un diplôme (*diplomeEnCours*). Les méthodes pour *Administratif*, *Enseignant* et *Etudiant* sont similaires. Une variable de classe dans chaque classe compte le nombre de personnes créées dans chaque catégorie. Une méthode static du même nom que la variable fournit la valeur de cette variable static (*nbAdministratifs*, *nbEnseignants*, *nbEtudiants*).

Pour chacune des trois catégories de personne, on désire disposer d'une méthode *afficheDescription()* qui affiche une description de la personne. Par exemple :

"*Enseignant : Asmaa El Hannani, rue Ahmed Chaouki El Jadida, spécialité : Informatique*"

"*Administratif : Mohammed Kamal, rue Ibnou Badis El Jadida, numéro du Bureau : 32*"

"*Etudiant : Nadia Rebbah, Bd Zerkouni Casa Blanca, diplôme en cours : Ingénieur en GEE*"

1. Ecrivez les classes permettant de gérer la hiérarchie des Personnes. Votre solution doit être fondée sur la notion de *classe abstraite* et de *polymorphisme*.
2. Ecrivez une classe principale *CollectionPersonne* dans laquelle vous définirez un tableau de personnes hétérogènes, que vous remplirez de personnes de votre choix, puis dont vous afficherez la description. A la fin affichez le nombre total des personnes ainsi que le nombre de personne de chaque catégorie.

Fiche des réponses

Nom :

Prénom :

Questions	Réponses				Note
	a	b	c	d	
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
Total					