

Exercice 1 (05 points)

Choisissez la (ou les) réponse(s) correctes.

1. **Que signifie le mot clé "static" associé à un attribut ?**
 - a) Que la valeur de cet attribut est constante
 - b) Que cet attribut n'est visible que dans la classe où il est défini
 - c) Que cet attribut sera toujours passé par valeur
 - d) Que cet attribut a une valeur unique pour toutes les instances de la classe
2. **Au sujet du mot clef "abstract", quelle assertion est fausse ?**
 - a) Une classe abstraite ne peut être instanciée
 - b) Une méthode abstraite n'a pas d'implémentation
 - c) Une classe abstraite n'a pas forcément de classe fille
 - d) Une classe abstraite doit contenir au moins une méthode abstraite
3. **Une interface en Java peut contenir :**
 - a) Des déclarations de méthodes et de constantes de classe
 - b) Des déclarations de méthodes et d'attributs
 - c) Des déclarations de méthodes et de méthodes statiques
 - d) Des déclarations de méthodes et de constructeurs
4. **Quel est le droit d'accès le plus restrictif pour donner l'accès aux membres d'une autre classe de même package ?**
 - a) public
 - b) default access
 - c) abstract
 - d) protected
5. **Qu'est-ce que l'état d'un objet?**
 - a) C'est l'ensemble des méthodes d'un objet.
 - b) C'est l'ensemble des attributs d'un objet.
 - c) C'est l'ensemble des valeurs des attributs d'un objet.
 - d) C'est l'ensemble des méthodes et attributs d'un objet
6. **Qu'est-ce qu'un attribut de classe?**
 - a) C'est un attribut visible que dans la classe où il est défini.
 - b) C'est un attribut qui a une valeur constante.
 - c) C'est un attribut qui a une valeur unique pour toutes les instances de la classe.
 - d) C'est un attribut dont le type est un type primitif.
7. **De quel concept se rapproche le plus le concept d'abstraction des données ?**
 - a) Héritage
 - b) Instanciation
 - c) Encapsulation
 - d) Association
8. **Donnez la ou les réponses correctes**
 - a) Une classe peut implémenter plusieurs interfaces mais doit étendre une seule classe.
 - b) Une classe peut implémenter plusieurs classes mais doit étendre une seule interface.
 - c) Une interface peut implémenter plusieurs interfaces.
 - d) Une interface peut étendre une seule classe.
9. **Qu'est-ce qu'une méthode de classe?**
 - a) Une méthode déclarée static
 - b) Une méthode déclarée abstract
 - c) Une méthode déclarée finale
 - d) Une méthode surchargée

10. La classe B est définie comme suit:

```
class B {
    public B(){
        System.out.print(" !");
    }
    public B(int i) {
        this();
        System.out.println("Bonjour "+i);};
}
```

Quel sera le résultat de l'appel suivant ?

```
B monB=new B(2019);
```

- a) Aucun affichage
- b) Une erreur d'exécution
- c) Affichage sur la console de " !Bonjour 2019"
- d) Affichage sur la console de "Bonjour 2019 !"

Exercice 2 (05 points)

1. Trouvez et corrigez les sept (07) erreurs dans les classes A, B et C.

```
package a;
class A {
    int a;
    public A(int a){this.a=a;}
    public void m(){System.out.println(a); }
}
```

```
package b;
public class B extends A {
    int b;
    public B (int a, int b){ this.b = b;}
    void m(){ System.out.println(a + b); }
}
```

```
package b;
public class C extends B {
    public void m(){ System.out.println(b);}
}
```

2. Trouvez et expliquez les deux (02) erreurs de casting dans la classe Test.

```
package c;
import a.*;
import b.*;
public class Test {
    public static void main(String args[]){
        1. A a=new A(5);
        2. B b=new B(10,20);
        3. b=a;
        4. ((A)b).m();
        5. Object d1 = (B) a;
        6. a=b;
        7. a.m();
        8. Object d2 = (B) a;
        9. ((B) d2).m();    }
}
```

3. Qu'affiche ce programme après suppression des deux lignes de code erroné ?

Exercice 3 (10 points)

Soit l'interface "IOperation".

```
package operations;

public interface IOperation {
    Object plus (Object obj); // addition de deux "Objects"
    Object moins (Object obj); // soustraction de deux "Objects"
}
```

1. Ecrivez la classe concrète "Complexe" qui implémente "IOperation".

```
package nombres;
...
.....Complexe..... {
    private double im ; //partie imaginaire
    private double re; //partie réelle

    ...
    ...
    ...
    ...
}
```

2. Redéfinissez dans la classe Complexe, la méthode toString() de la classe Object de telle façon qu'elle retourne une chaîne de caractères sous la forme : $re + im i$ (Exemple : "1+ 2 i")

3. Ecrivez la classe concrète "Reel" qui implémente "IOperation".

```
package nombres;
...
.....Reel..... {
    private double x; // valeur du réel

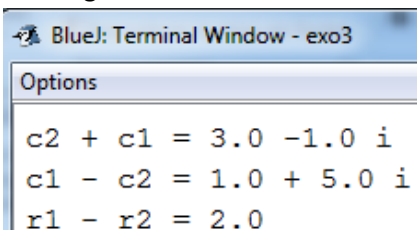
    ...
    ...
    ...
    ...
}
```

4. Redéfinissez dans la classe Reel, la méthode toString() de la classe Object de telle façon qu'elle retourne une chaîne de caractères sous la forme : x (Exemple : "5")

5. Ecrivez la classe "Test" (principale) qui permettra de :

- Créez deux objets "Complexe" : $c1 = 2 + 2i$ (partie réelle = 2, partie imaginaire = 2) et $c2 = 1 - 3i$
- Créez deux objets "Reel" : $r1 = 5$ et $r2 = 3$
- Additionnez $c1$ et $c2$ puis affichez le résultat
- Soustrayez $c2$ de $c1$ puis affichez le résultat
- Soustrayez $r2$ de $r1$ puis affichez le résultat

Affichage attendu



```
Blue: Terminal Window - exo3
Options
c2 + c1 = 3.0 -1.0 i
c1 - c2 = 1.0 + 5.0 i
r1 - r2 = 2.0
```

Bon courage !

Exercice 1 (05 points)

Choisissez la (ou les) réponse(s) correctes.

1d	2(c et/ou d)	(3a ou 3ac)	4b	5c	6c	7c	8a	9a	10c
----	--------------	-------------	----	----	----	----	----	----	-----

Exercice 2 (05 points)

```
package a;
public class A {
    int a; // « protected ou public int a » est aussi accepté (1 pt)
    public A(int a){this.a=a;}
    public int getA(){return a;}
    public void m(){System.out.println(a); }
}
```

0.5

0.5

```
package b;
import a.*;
public class B extends A {
    int b;
    public B (int a, int b){super(a);this.b = b;}
    public void m(){ System.out.println(getA() + b); }
}
```

0.5

0.5

0.5+0.5

```
package b;
public class C extends B {
    public C(int a, int b){super(a,b);}
    public void m(){ System.out.println(b);}
}
```

0.5

```
package c;
import a.*;
import b.*;
public class Test {
    public static void main(String args[]){
        1. A a=new A(5);
        2. B b=new B(10,20);
        3. b=a; //downcast implicite
        4. ((A)b).m();
        5. Object d1 = (B) a; // ClassCastException
        6. a=b;
        7. a.m();
        8. Object d2 = (B) a;
        9. ((B) d2).m();
    }
}
```

0.5

0.5

Affichage : 30

30

30

0.5

Exercice 10 (10 points)

Classe Complexe : (03.25 points)

```
package nombres;
import operations.*;
public class Complexe 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 toString() {
        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+"";
    }
}
```

Classe Reel : (03.25 points)

```
package nombres;
import operations.*;
public class Reel 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 toString() {
        return (x+"");
    }
}
```

Classe Test (03.50 points)

```
package nombres;
public class Test {
    public static void main (String[] args){
        Complexe c1 = new Complexe(2,2) ;
        Complexe c2 = new Complexe(1,-3) ;
        Complexe c3,c4 ;
        Reel r1=new Reel(5);
        Reel r2=new Reel(3);
        Reel r3;
        c3 =(Complexe)(c1.plus(c2));
        c4=(Complexe)(c1.moins(c2));
        System.out.println("c2 + c1 = " +c3);
        System.out.println("c1 - c2 = " +c4);
        r3=(Reel)(r1.moins(r2));
        System.out.println("r1 - r2 = " +r3);
    }
}
```

0.25

0.25

0.25

0.25

0.25

0.5

0.5

0.25

0.25

0.5

0.25