

Programmation Orientée Objet
qcm4, Version: A

Nom: _____

Carte d'étudiant: _____

Remplissez la table avec les lettres correspondant à vos réponses.

Questions	1	2	3	4	5	6	7	8	9	10	11	12	13
Réponse(s)													

Bonne réponse=2pt; mauvaise réponse ou réponse incomplète =-1pt; pas de réponse=0pt

Dans ce qui suit un “warning” du compilateur n’est pas considéré comme une erreur à la compilation.

1. Le code:

```
Number t1[] = new Integer[12]; t1[1]=10;
```

Provoque

- (a) Une erreur à la compilation
- (b) Une erreur à l'exécution
- (c) Ne provoque pas d'erreur

2.

```
public static double Sommeb(List<?> L) {  
    double s = 0.0; for (Object n : L) s += ((Number)n).doubleValue(); return s; }
```

Et le code:

```
List l = new ArrayList(10); for (int i = 0; i < 10; i++) l.add(i, i); System.out.println(Sommeb(l));
```

Provoquent

- (a) Une erreur à la compilation
- (b) Une erreur à l'exécution
- (c) Affiche 45.0

3.

```
public static double Somme(List<Number> L){  
    double s=0.0; for(Number n: L)s+=n.doubleValue(); return s;  
}
```

Et le code:

```
List l1 = new ArrayList(10); for (int i = 0; i < 10; i++) l1.add(i,"x"); System.out.println(Somme(l1));
```

Provoquent

- (a) Une erreur à la compilation
- (b) Une erreur à l'exécution

4.

```
public static double Somme(List<Number> L){  
    double s=0.0; for(Number n: L)s+=n.doubleValue(); return s;  
}
```

Et le code:

```
ArrayList l= new ArrayList(10); for( int i=0; i<10;i++)l.add(i,i); System.out.println(Somme(l));
```

Provoquent

- (a) Une erreur à la compilation
- (b) Une erreur à l'exécution
- (c) Affiche 45.0

5.

```
public static double Sommed(Number[] L){  
    double s = 0.0; for (Number n : L) s += n.doubleValue(); return s; }
```

Et le code:

```
Integer[] t=new Integer[10]; for(int i=0;i<10;i++)t[i]=i; System.out.println(Sommed(t));
```

Provoquent

- (a) Une erreur à la compilation
- (b) Une erreur à l'exécution
- (c) Affiche 45.0

6. `public static double Somme(List<Number> L){
 double s=0.0; for(Number n: L)s+=n.doubleValue(); return s;
}`

Et le code:

```
List<?> n= new ArrayList(10); for( int i=0; i<10;i++)n.add(i,i); System.out.println(Somme(n));
```

Provoquent

- (a) Une erreur à la compilation
- (b) Une erreur à l'exécution
- (c) Affiche 45.0

7. `public static double Sommeb(List<?> L) {
 double s = 0.0; for (Object n : L) s += ((Number)n).doubleValue(); return s; }`

Et le code:

```
ArrayList<Integer> m = new ArrayList<>(10); for (int i = 0; i < 10; i++) m.add(i, i);  
System.out.println(Sommeb(m));
```

Provoquent

- (a) Une erreur à la compilation
- (b) Une erreur à l'exécution
- (c) Affiche 45.0

8. `public static double Somme(List<Number> L){
 double s=0.0; for(Number n: L)s+=n.doubleValue(); return s;
}`

Et le code:

```
ArrayList<Integer> m= new ArrayList<>(10); for( int i=0; i<10;i++)m.add(i,i);  
System.out.println(Somme(m));
```

Provoquent

- (a) Une erreur à la compilation
- (b) Une erreur à l'exécution
- (c) Affiche 45.0

9. Le code:

```
List<Number> ln =new ArrayList<Integer>(1); ln.add(1,1);
```

Provoque

- (a) Une erreur à la compilation
- (b) Une erreur à l'exécution
- (c) Ne provoque pas d'erreur

10. `public static double Sommec(List<? extends Number> L) {
 double s = 0.0; for (Number n : L) s += n.doubleValue(); return s; }`

Et le code:

```
List<Integer> m = new ArrayList<>(10); for (int i = 0; i < 10; i++) m.add(i, i);  
System.out.println(Sommec(m));
```

Provoquent

- (a) Une erreur à la compilation
- (b) Une erreur à l'exécution
- (c) Affiche 45.0

11. `public static double Sommec(List<? extends Number> L) {
 double s = 0.0; for (Number n : L) s += n.doubleValue(); return s; }`

Et le code:

```
ArrayList<Integer> m = new ArrayList<>(10); for (int i = 0; i < 10; i++) m.add(i, i);  
System.out.println(Sommec(m));
```

Provoquent

- (a) Une erreur à la compilation
 - (b) Une erreur à l'exécution
 - (c) Affiche 45.0
12. `public static double Sommed(Number[] L){
 double s = 0.0; for (Number n : L) s += n.doubleValue(); return s; }`

Et le code:

```
Number[] t=new Integer[10]; for(int i=0;i<10;i++)t[i]=i; System.out.println(Sommed(t));
```

Provoquent

- (a) Une erreur à la compilation
 - (b) Une erreur à l'exécution
 - (c) Affiche 45.0
13. `public static double Somme(List<Number> L){
 double s=0.0; for(Number n: L)s+=n.doubleValue(); return s;
}`

Et le code:

```
List l= new ArrayList(10); for( int i=0; i<10;i++)l.add(i,i); System.out.println(Somme(l));
```

Provoquent

- (a) Une erreur à la compilation
- (b) Une erreur à l'exécution
- (c) Affiche 45.0

Answer Key for Exam A

Bonne réponse=2pt; mauvaise réponse ou réponse incomplète =-1pt; pas de réponse=0pt

Dans ce qui suit un “warning” du compilateur n’est pas considéré comme une erreur à la compilation.

1. Le code:

```
Number t1[] = new Integer[12]; t1[1]=10;
```

Provoque

- (a) Une erreur à la compilation
- (b) Une erreur à l'exécution
- ☒ (c) Ne provoque pas d'erreur

2.

```
public static double Sommeb(List<?> L) {  
    double s = 0.0; for (Object n : L) s += ((Number)n).doubleValue(); return s; }
```

Et le code:

```
List l = new ArrayList(10); for (int i = 0; i < 10; i++) l.add(i, i); System.out.println(Sommeb(l));
```

Provoquent

- (a) Une erreur à la compilation
- (b) Une erreur à l'exécution
- ☒ (c) Affiche 45.0

3.

```
public static double Somme(List<Number> L){  
    double s=0.0; for(Number n: L)s+=n.doubleValue(); return s;  
}
```

Et le code:

```
List l1 = new ArrayList(10); for (int i = 0; i < 10; i++) l1.add(i,"x"); System.out.println(Somme(l1));
```

Provoquent

- (a) Une erreur à la compilation
- ☒ (b) Une erreur à l'exécution

4.

```
public static double Somme(List<Number> L){  
    double s=0.0; for(Number n: L)s+=n.doubleValue(); return s;  
}
```

Et le code:

```
ArrayList l= new ArrayList(10); for( int i=0; i<10;i++)l.add(i,i); System.out.println(Somme(l));
```

Provoquent

- (a) Une erreur à la compilation
- (b) Une erreur à l'exécution
- ☒ (c) Affiche 45.0

5.

```
public static double Sommed(Number[] L){  
    double s = 0.0; for (Number n : L) s += n.doubleValue(); return s; }
```

Et le code:

```
Integer[] t=new Integer[10]; for(int i=0;i<10;i++)t[i]=i; System.out.println(Sommed(t));
```

Provoquent

- (a) Une erreur à la compilation
- (b) Une erreur à l'exécution
- ☒ (c) Affiche 45.0

6. `public static double Somme(List<Number> L){
 double s=0.0; for(Number n: L)s+=n.doubleValue(); return s;
}`

Et le code:

```
List<?> n= new ArrayList(10); for( int i=0; i<10;i++)n.add(i,i); System.out.println(Somme(n));
```

Provoquent

- ☒ (a) Une erreur à la compilation
- ☐ (b) Une erreur à l'exécution
- ☐ (c) Affiche 45.0

7. `public static double Sommeb(List<?> L) {
 double s = 0.0; for (Object n : L) s += ((Number)n).doubleValue(); return s; }`

Et le code:

```
ArrayList<Integer> m = new ArrayList<>(10); for (int i = 0; i < 10; i++) m.add(i, i);  
System.out.println(Sommeb(m));
```

Provoquent

- ☐ (a) Une erreur à la compilation
- ☐ (b) Une erreur à l'exécution
- ☒ (c) Affiche 45.0

8. `public static double Somme(List<Number> L){
 double s=0.0; for(Number n: L)s+=n.doubleValue(); return s;
}`

Et le code:

```
ArrayList<Integer> m= new ArrayList<>(10); for( int i=0; i<10;i++)m.add(i,i);  
System.out.println(Somme(m));
```

Provoquent

- ☒ (a) Une erreur à la compilation
- ☐ (b) Une erreur à l'exécution
- ☐ (c) Affiche 45.0

9. Le code:

```
List<Number> ln =new ArrayList<Integer>(1); ln.add(1,1);
```

Provoque

- ☒ (a) Une erreur à la compilation
- ☐ (b) Une erreur à l'exécution
- ☐ (c) Ne provoque pas d'erreur

10. `public static double Sommec(List<? extends Number> L) {
 double s = 0.0; for (Number n : L) s += n.doubleValue(); return s; }`

Et le code:

```
List<Integer> m = new ArrayList<>(10); for (int i = 0; i < 10; i++) m.add(i, i);  
System.out.println(Sommec(m));
```

Provoquent

- ☐ (a) Une erreur à la compilation
- ☐ (b) Une erreur à l'exécution
- ☒ (c) Affiche 45.0

11. `public static double Sommec(List<? extends Number> L) {
 double s = 0.0; for (Number n : L) s += n.doubleValue(); return s; }`

Et le code:

```
ArrayList<Integer> m = new ArrayList<>(10); for (int i = 0; i < 10; i++) m.add(i, i);  
System.out.println(Sommec(m));
```

Provoquent

- (a) Une erreur à la compilation
 - (b) Une erreur à l'exécution
 - ☒ (c) Affiche 45.0
12. `public static double Sommed(Number[] L){
 double s = 0.0; for (Number n : L) s += n.doubleValue(); return s; }`

Et le code:

```
Number[] t=new Integer[10]; for(int i=0;i<10;i++)t[i]=i; System.out.println(Sommed(t));
```

Provoquent

- (a) Une erreur à la compilation
 - (b) Une erreur à l'exécution
 - ☒ (c) Affiche 45.0
13. `public static double Somme(List<Number> L){
 double s=0.0; for(Number n: L)s+=n.doubleValue(); return s;
}`

Et le code:

```
List l= new ArrayList(10); for( int i=0; i<10;i++)l.add(i,i); System.out.println(Somme(l));
```

Provoquent

- (a) Une erreur à la compilation
- (b) Une erreur à l'exécution
- ☒ (c) Affiche 45.0