

Programmation Orientée Objet  
qcm3, 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	14	15	16	17	18	19
Réponse(s)																			

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

```

1. interface I{
    void f();
    int f(int i);
}

interface J extends I{
    void f(double d);
    void f();
}

class A implements J{
    public void f(){;}
    public int f(int i){return 0;}
    public void f(double d){;}
    void f(char c){;}
}
    
```

Trouver la ou les bonnes réponses:

- (a) ce code ne peut pas être compilé
- (b) le code suivant "I i=new A(); J j=(J)i; j.f(3.2);" est correct (pas d'erreur ni à la compilation ni à l'exécution)
- (c) le code " I ii= new J(); ii.f();" est correct (pas d'erreur ni à la compilation ni à l'exécution)

```

2. class A{
    static int i=0;int j=10;
    void f(){System.out.println("i="+i);System.out.println(" j="+j);}
    static void g(){this.f();}
}
    
```

avec le morceau de code: (new A()).g();

- (a) provoque une erreur à la compilation
- (b) a che i=0 j=10
- (c) provoque une erreur à l'exécution

```

3. class D {public int val; public D(int i){val=i;}}
class E extends D{public E(int i){super(i);}}
    
```

et le code :

```
D[] t1= new D[4]; t1[0]=new D(1); t1[1]=new E(3); System.out.println(t1[0].val+" "+t1[1].val);
```

- (a) provoque une erreur à la compilation
- (b) a che 1 3
- (c) provoquent une erreur à l'exécution

```

4. class XX {void f(){System.out.println("XX");}; }
class YY extends XX{void f() throws Exception{throw new Exception();}}
Et le code "((XX).(new YY()).f());"
    
```

- (a) Provoquent la levée d'une exception
- (b) A chent XX
- (c) Les classes XX et YY ne peuvent pas être compilées

5. class X implements Cloneable {  
 public int i = 8; public int[] ti = new int[3];  
 public Object clone() {X tmp = null;  
 try {tmp = (X) super.clone();}catch (Exception e) {}  
 return tmp; }  
 }

Le code suivant: "X x1= new X(); x1.ti[0] = 1;X x3 = (X) x1.clone(); x3.i = 3;x3.ti[0] = 3;  
 System.out.println(x1.ti[0] + " " + x1.i); " :

- (a) provoque une erreur à la compilation ou à l'exécution
- (b) a che 3 3
- (c) a che 3 8
- (d) a che 1 3
- (e) a che 1 8

6. interface I{int i;void f();} avec le morceau de code:  
 (new I(){public void f(){i= 1; System.out.println("bravo "+ i);}}).f();

- (a) provoquent une erreur à la compilation
- (b) a chent bravo 1

7. class A{  
 static int i=0; void f(){System.out.print(" i="+i);}  
 }  
 class B extends A{  
 static int i=5; static void f() {System.out.println(" i="+i);}  
 }

avec le morceau de code: A a=new B();a.f();

- (a) provoquent une erreur à l'exécution
- (b) provoquent une erreur à la compilation
- (c) a chent i=0
- (d) a chent i=5

8. class D {public int val; public D(int i){val=i;}}  
 class E extends D{public E(int i){super(i);}}

avec le code :

D[] t1= new E[4]; t1[0]=new D(1); t1[1]=new E(3); System.out.println(t1[0].val+" "+t1[1].val);

- (a) provoquent une erreur à la compilation
- (b) a chent 1 3
- (c) provoquent une erreur à l'exécution

9. class A1 {void f(){System.out.println(i);}int i=0;}  
 class B1 extends A1{public void f(){System.out.println(i);}int i=1;}  
 class B2 extends A1{public void f(){System.out.println(i);}int i=2;}

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) a che 1
- (d) a che 2
- (e) a che 0

10. `class A{int i=10; void f(){System.out.print(" i="+i);}}`  
`class B extends A{static int i=0; void f(){System.out.print("i="+i);super.f(); }}`  
 avec le morceau de code: `A a=new B();a.f();`

- (a) provoque une erreur à l'exécution
- (b) provoque une erreur à la compilation
- (c) a che i=10 i=10
- (d) a che i=0 i=10

11. `class X {  
     public int i = 8; public int[] ti = new int[3];  
     public Object clone() {X tmp = null;  
         try {tmp = (X) super.clone();tmp.ti = ti.clone();}catch (Exception e) {}  
         return tmp; }  
 }`

Le code suivant: `"X x1= new X(); x1.ti[0] = 1;X x3 = (X) x1.clone(); x3.i = 3;x3.ti[0] = 3;  
 System.out.println(x1.ti[0] + " " + x1.i); "` :

- (a) provoque une erreur à la compilation ou à l'exécution
- (b) a che 3 3
- (c) a che 3 8
- (d) a che 1 3
- (e) a che 1 8

12. `class A1 {void f(){System.out.println(i);}int i=0;}`  
`class B1 extends A1{public void f(){System.out.println(i);}int i=1;}`  
`class B2 extends A1{public void f(){System.out.println(i);}int i=2;}`

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) a che 1
- (d) a che 2
- (e) a che 0

13. `class X implements Cloneable {  
     public int i = 8; public int[] ti = new int[3];  
     public Object clone() {X tmp = null;  
         try {tmp = (X) super.clone();tmp.ti = ti.clone();}catch (Exception e) {}  
         return tmp; }  
 }`

Le code suivant: `"X x1= new X(); x1.ti[0] = 1;X x3 = (X) x1.clone(); x3.i = 3;x3.ti[0] = 3;  
 System.out.println(x1.ti[0] + " " + x1.i); "` :

- (a) provoque une erreur à la compilation ou à l'exécution
- (b) a che 3 3
- (c) a che 3 8
- (d) a che 1 3
- (e) a che 1 8

14. `class E1 extends Exception{}`  
`class E2 extends E1{}`  
`class ZZ {void f() throws E2{}}`  
`class UU extends ZZ{void f() throws E1{}}`

- (a) Ces classes peuvent être compilées
- (b) Ces classes ne peuvent pas être compilées

15. `class A{static int i=0;int j=10;void f(){System.out.print(" i="+i);System.out.print(" j="+j);}}`  
`class B extends A{int i=15; void f(){super.f();System.out.print(" i="+i);System.out.println(" j="+j);}}`  
avec le morceau de code: `A a=new B();a.f();`
- (a) provoque une erreur à l'exécution
  - (b) provoque une erreur à la compilation
  - (c) a che i=0 j=10 i=15 j=10
  - (d) a che i=0 j=10
16. `class E1 extends Exception{}`  
`class E2 extends E1{}`  
`class ZZ {void f() throws E2{}}`  
`class TT extends ZZ{void f(){}}`
- (a) Ces classes peuvent être compilées
  - (b) Ces classes ne peuvent pas être compilées
17. `interface I{void f();}`  
`class A implements I{public void f(){System.out.println("A");}}`  
`class B implements I{public void f(){System.out.println("B");}}`  
et le code suivant: `"I[] tabl=new I[2]; tabl[0]= new A();tabl[1]= new B(); tabl[0].f(); "`
- (a) provoquent une erreur à la compilation
  - (b) provoquent une erreur à l'exécution
  - (c) a chent A
18. `interface I{static int i=0; void f();}` avec le morceau de code:  
`(new I(){int i=2; public void f(){System.out.println("bravo"+ i);}}).f();`
- (a) provoque une erreur à la compilation
  - (b) a che bravo 2
  - (c) a che bravo 0
19. `interface I{void f();}`  
`class A implements I{public void f(){System.out.println("A");}}`  
`class B implements I{public void f(){System.out.println("B");}}`  
Le code suivant: `"I[] tabl=new I[2]; tabl[0]= new A();tabl[1]= new B(); tabl[0].f(); "`
- (a) provoque une erreur à la compilation
  - (b) provoque une erreur à l'exécution
  - (c) a che A

# Answer Key for Exam | | |---| | A | |---|

1. (b)
2. (a)
3. (b)
4. (c)
5. (c)
6. (a)
7. (b)
8. (c)
9. (b)
10. (d)
11. (a)
12. (b)
13. (e)
14. (b)
15. (c)
16. (a)
17. (c)
18. (b)
19. (b)