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	20
Réponse(s)																				

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

1.
```
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

2. Soient la classe paramétrée: `class X <T>{public T val; X(T x){val=x;}}` et la méthode:
`public static void somme(X<Number> x){System.out.println(x.val.doubleValue());}`
Le code: `X<Integer> x= new X<Integer>(2); somme(x);`

   (a) provoque une erreur à la compilation ou à l'exécution

   (b) affiche i=2.0

3.
```
class JA<B>{
  B b;
  JA(B b){this.b=b;}
  B get() {return b; }
}
```

   Le code: `JA<? super Integer> vsi=new JA<Number>(2); System.out.println(vsi.get());`

   (a) provoque une erreur à la compilation

   (b) provoque une erreur à l'exécution

   (c) affiche "2"

4. On considère la classe:

```
class JA<B>{
  B b;
  JA(B b){this.b=b;}
  B get() {return b; }
}
```

   Le code: `JA<?> vj= new JA<Integer>(1); System.out.println(vj.get());`

   (a) provoque une erreur à la compilation

   (b) provoque une erreur à l'exécution

   (c) affiche "1"

5. Soient la classe paramétrée: `class X <T>{public T val; X(T x){val=x;}}` et la méthode:
`public static void somme(X<? extends Number> x){System.out.println(x.val.doubleValue());}`
Le code: `X<Integer> x= new X<Integer>(2); somme(x);`

   (a) provoque une erreur à la compilation ou à l'exécution

   (b) affiche i=2.0

6. Soient: class X <T>{public T val; X(T x){val=x;}} et:
public static void somme(X<? super Number> x){System.out.println(x.val.doubleValue());}
Le code: X<Integer> x= new X<Integer>(2); somme(x);

   (a) provoque une erreur à la compilation ou à l'exécution

   (b) affiche i=2.0

7. L'interface: 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) affiche bravo 2

   (c) affiche bravo 0

8. On considère la classe:

```
class JA<B>{
  B b;
  JA(B b){this.b=b;}
  void put(B b){this.b=b; }
}
```

   Le code: JA<?> vj= new JA<Integer>(1); vj.put(new Integer(1));

   (a) provoque une erreur

   (b) ne provoque pas d'erreur

9. 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

10. Les classes:

```
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) affichent 1 3

   (c) provoquent une erreur à l'exécution

11. On considère la classe:

```
class JA<B>{
  B b;
  JA(B b){this.b=b;}
  B get() {return b; }
}
```

   Le code: JA<Integer> vj= new JA<Integer>(1); System.out.println(vj.get());

   (a) provoque une erreur à la compilation

   (b) provoque une erreur à l'exécution

   (c) affiche "1"

12. On considère la classe:

```
class JA<B>{
  B b;
  JA(B b){this.b=b;}
  B get() {return b; }
}
```

Le code: `JA<Integer> vi=new JA<Integer>(3);JA<Number> vn=vi; System.out.println(vn.get());`

(a) provoque une erreur à la compilation

(b) provoque une erreur à l'exécution

(c) affiche "3"

13. Soient la classe paramétrée: `class X <T>{public T val; X(T x){val=x;}}`
et la méthode: `public static void somme(X<?> x){System.out.println(x.val.doubleValue());}`
Le code: `X<Number> x= new X<Number>(2); somme(x);`

(a) provoque une erreur à la compilation ou à l'exécution

(b) affiche i=2.0

14. Soient la classe paramétrée: `class X <T>{public T val; X(T x){val=x;}}`
et la méthode: `public static void somme(X<Number> x){System.out.println(x.val.doubleValue());}`
Le code: `X<Number> x= new X<Number>(new Integer(2)); somme(x);`

(a) provoque une erreur à la compilation ou à l'exécution

(b) affiche i=2.0

15. On considère la classe:

```
class JA<B>{
  B b;
  JA(B b){this.b=b;}
  B get() {return b; }
}
```

Le code:
`JA<Integer> vi=new JA<Integer>(3);JA<? super Integer> vsi=vi; System.out.println(vsi.get());`

(a) provoque une erreur à la compilation

(b) provoque une erreur à l'exécution

(c) affiche "3"

16. On considère la classe:

```
class JA<B>{
  B b;
  JA(B b){this.b=b;}
  void put(B b){this.b=b;}
}
```

Le code:
`JA<Integer> vi=new JA<Integer>(3); vi.put(new Integer(2));`

(a) provoque une erreur

(b) ne provoque pas d'erreur

17. On considère la classe:

```
class JA<B>{
   B b;
   JA(B b){this.b=b;}
   B get() {return b; }
}
```

Le code: `JA v=new JA<Integer>(1);; JA<? super Number> vsn=v; System.out.println(v.get());`

(a) provoque une erreur à la compilation

(b) provoque une erreur à l'exécution

(c) affiche "1"

18. On considère les classes:

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

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) affiche 1 3

(c) provoquent une erreur à l'exécution

19. L'interface: `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) provoque une erreur à la compilation

(b) affiche bravo 1

20.
```
class XX {void f(){}; }
class YY extends XX{void f() throws Exception{throw new Exception();}}
```

(a) Ces classes peuvent être compilées

(b) Ces classes ne peuvent pas être compilées

Answer Key for Exam A

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

1. class E1 extends Exception{}
   class E2 extends E1{}
   class ZZ {void f() throws E2{}}
   class TT extends ZZ{void f(){}}

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

2. Soient la classe paramétrée: class X <T>{public T val; X(T x){val=x;}} et la méthode:
   public static void somme(X<Number> x){System.out.println(x.val.doubleValue());}
   Le code: X<Integer> x= new X<Integer>(2); somme(x);

   - [x] (a) provoque une erreur à la compilation ou à l'exécution
   - [ ] (b) affiche i=2.0

3. class JA<B>{
   B b;
   JA(B b){this.b=b;}
   B get() {return b; }
   }

   Le code: JA<? super Integer> vsi=new JA<Number>(2); System.out.println(vsi.get());

   - [ ] (a) provoque une erreur à la compilation
   - [ ] (b) provoque une erreur à l'exécution
   - [x] (c) affiche "2"

4. On considère la classe:

   class JA<B>{
   B b;
   JA(B b){this.b=b;}
   B get() {return b; }
   }

   Le code: JA<?> vj= new JA<Integer>(1); System.out.println(vj.get());

   - [ ] (a) provoque une erreur à la compilation
   - [ ] (b) provoque une erreur à l'exécution
   - [x] (c) affiche "1"

5. Soient la classe paramétrée: class X <T>{public T val; X(T x){val=x;}} et la méthode:
   public static void somme(X<? extends Number> x){System.out.println(x.val.doubleValue());}
   Le code: X<Integer> x= new X<Integer>(2); somme(x);

   - [ ] (a) provoque une erreur à la compilation ou à l'exécution
   - [x] (b) affiche i=2.0

6. Soient: class X <T>{public T val; X(T x){val=x;}} et:
   public static void somme(X<? super Number> x){System.out.println(x.val.doubleValue());}
   Le code: X<Integer> x= new X<Integer>(2); somme(x);

   - [x] (a) provoque une erreur à la compilation ou à l'exécution
   - [ ] (b) affiche i=2.0

7. L'interface: 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
   - [x] (b) affiche bravo 2
   - [ ] (c) affiche bravo 0

8. On considère la classe:

```
class JA<B>{
  B b;
  JA(B b){this.b=b;}
  void put(B b){this.b=b; }
}
```

Le code: `JA<?> vj= new JA<Integer>(1); vj.put(new Integer(1));`

- [x] (a) provoque une erreur
- [ ] (b) ne provoque pas d'erreur

9.
```
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
- [x] (b) Ces classes ne peuvent pas être compilées

10. Les classes:

```
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) affichent 1 3
- [x] (c) provoquent une erreur à l'exécution

11. On considère la classe:

```
class JA<B>{
  B b;
  JA(B b){this.b=b;}
  B get() {return b; }
}
```

Le code: `JA<Integer> vj= new JA<Integer>(1); System.out.println(vj.get());`

- [ ] (a) provoque une erreur à la compilation
- [ ] (b) provoque une erreur à l'exécution
- [x] (c) affiche "1"

12. On considère la classe:

```
class JA<B>{
  B b;
  JA(B b){this.b=b;}
  B get() {return b; }
}
```

Le code: `JA<Integer> vi=new JA<Integer>(3);JA<Number> vn=vi; System.out.println(vn.get());`

- [x] (a) provoque une erreur à la compilation
- [ ] (b) provoque une erreur à l'exécution
- [ ] (c) affiche "3"

13. Soient la classe paramétrée: class X <T>{public T val; X(T x){val=x;}}
et la méthode: public static void somme(X<?> x){System.out.println(x.val.doubleValue());}
Le code: X<Number> x= new X<Number>(2); somme(x);

[(a)] provoque une erreur à la compilation ou à l'exécution
(b) affiche i=2.0

14. Soient la classe paramétrée: class X <T>{public T val; X(T x){val=x;}}
et la méthode: public static void somme(X<Number> x){System.out.println(x.val.doubleValue());}
Le code: X<Number> x= new X<Number>(new Integer(2)); somme(x);

(a) provoque une erreur à la compilation ou à l'exécution
[(b)] affiche i=2.0

15. On considère la classe:

```
class JA<B>{
  B b;
  JA(B b){this.b=b;}
  B get() {return b; }
}
```

Le code:
JA<Integer> vi=new JA<Integer>(3);JA<? super Integer> vsi=vi; System.out.println(vsi.get());

(a) provoque une erreur à la compilation
(b) provoque une erreur à l'exécution
[(c)] affiche "3"

16. On considère la classe:

```
class JA<B>{
  B b;
  JA(B b){this.b=b;}
  void put(B b){this.b=b;}
}
```

Le code:
JA<Integer> vi=new JA<Integer>(3); vi.put(new Integer(2));

(a) provoque une erreur
[(b)] ne provoque pas d'erreur

17. On considère la classe:

```
class JA<B>{
  B b;
  JA(B b){this.b=b;}
  B get() {return b; }
}
```

Le code: JA v=new JA<Integer>(1);; JA<? super Number> vsn=v; System.out.println(v.get());

(a) provoque une erreur à la compilation
(b) provoque une erreur à l'exécution
[(c)] affiche "1"

18. On considère les classes:

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

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)] affiche 1 3
(c) provoquent une erreur à l'exécution

19. L'interface: `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)] provoque une erreur à la compilation
   (b) affiche bravo 1

20. `class XX {void f(){}; }`
`class YY extends XX{void f() throws Exception{throw new Exception();}}`

   (a) Ces classes peuvent être compilées
   [(b)] Ces classes ne peuvent pas être compilées