Conservatoire National des Arts et Métiers
Centre Régional de Basse Normandie
Informatique Avancé, NFP121

Questionnaire à choix multiple sur Java 1.5.
Série 2

Durée : 30 minutes

Documents autorisés : Aucun

Préambule :
Prenez bien le temps de lire les énoncés. Les questions peuvent être traitées dans un ordre quelconque.
Vous devez entourer clairement les numéros correspondant à vos réponses directement sur le support qui doit être rendu pour la correction. Il n'est pas nécessaire d'argumenter vos choix.

---

1. Un programme comporte les deux fichiers suivants :

Fichier Sup.java :

```
package org.cnam.package1;

public class Sup {
    static{
        System.out.println("1");
    }
    {
        System.out.println("2");
    }
      public Sup(int i) {
        System.out.println("3");
      }
}
```

Fichier Test.java :

```
package org.cnam.package1;

public class Test extends Sup {
    static{
         System.out.println("4");
    }
    {
         System.out.println("5");
     }
     public Test(int i){
        super(i);
         System.out.println("6");
     }
     public static void main(String[] args) {
        Test t = new Test(1);
     }
}
```

Ce programme :

1. affiche : 123456
2. affiche : 142356
3. affiche : 124365
4. n'affiche rien
5. ne compile pas
6. lance une Runtime error

2. Le programme suivant :

Fichier Test.java :

```
package org.cnam.package1;

public class Test {
     public Test(int i){
       super();
       System.out.println("1");
     }
     public static void main(String[] args) {
       Test t = new Test(1);
     }
}
```

1. affiche : 1
2. n'affiche rien
3. ne compile pas
4. lance une Runtime error

3. Un programme comporte le fichier suivant :

Fichier Test.java :

```
package org.cnam.package1;

public class Sup {
      public Sup() {
        System.out.println("1");
      }
}

public class Test extends Sup {
     public Test(){
       super();
         System.out.println("3");
     }
     public static void main(String[] args) {
       Test t = new Test();
     }
}
```

1. affiche : 13
2. affiche : 31
3. n'affiche rien
4. ne compile pas
5. lance une Runtime error

4. Un programme comporte le fichier suivant :

Fichier Test.java :

```
package org.cnam.package1;

class Sup {
      public Sup() {
        System.out.println("1");
      }
}

public class Test extends Sup {
     public Test(){
       super();
         System.out.println("3");
     }
     public static void main(String[] args) {
       Test t = new Test();
     }
}
```

1. affiche : 13.
2. affiche : 31
3. n'affiche rien
4. ne compile pas
5. lance une Runtime error

5. Le programme suivant :

Fichier Test.java :

```
package org.cnam.package1;

public class Test {
      public void method() {
            System.out.println("1");
      }
      static public void main(String[] args) {
            Test t = new Test() {
                  public void method() {
                        System.out.println("2");
                  }
            };
            t.method();
      }
}
```

1. affiche : 1
2. affiche : 2
3. n'affiche rien
4. ne compile pas
5. lance une Runtime error

6. Le programme suivant :

Fichier Test.java :

```
class Test {
	static public void main(String[] args) {
		Object t = new Object () {
			public void method() {
				System.out.println("2");
			}
		};
		t.method();
	}
}
```

1. affiche : 2
2. n'affiche rien
3. ne compile pas
4. lance une Runtime error

7. Le programme suivant :

Fichier Test.java :

```
package org.cnam.package1;

public class Test {
	static public void main(String[] args) {
		Object t = new Object () {
			public String toString() {
				return "2";
			}
		};
		System.out.println(t);
	}
}
```

1. affiche : 2
2. n'affiche rien
3. ne compile pas
4. lance une Runtime error

8. Le programme suivant :

Fichier Test.java :

```
package org.cnam.package1;

class Test1 {

	static public class Interne {
		public void method() {
			System.out.println("1");
		}
	}
}
public class Test {
	static public void main(String[] args) {
		Test1.Interne i =
		   new Test1.Interne();
		i.method();
	}
}
```

1. affiche : 1
2. ne compile pas
3. lance une Runtime error

9. Le programme suivant :

Fichier Test.java :

```
package org.cnam.package1;

public class Test {
	private int attribut;

	class Interne {
		public void method2() {
			System.out.println(attribut);
		}
	}
	public void method() {
		attribut = 1;
		new Interne().method2();
	}
	static public void main(String[] args) {
		Test t = new Test();
		t.method();
	}
}
```

1. affiche : 1
2. ne compile pas
3. lance une Runtime error

10. Le programme suivant :

Fichier Test.java :

```
package org.cnam.package1;

public class Test {
	private int attribut;

	static class Interne {
		public void method2() {
			System.out.println(attribut);
		}
	}
	public void method() {
		attribut = 1;
		new Interne().method2();
	}
	static public void main(String[] args) {
		Test t = new Test();
		t.method();
	}
}
```

1. affiche : 1
2. ne compile pas
3. lance une Runtime error

11. Le programme suivant :

Fichier Test.java :

```
package org.cnam.package1;

public class Test{
   public static void main(String[] args){
       Integer myint1 = new Integer(10);
       int myint2 = myint1;
       System.out.println(myint2);
   }
}
```

1. affiche : 10
2. n'affiche rien
3. ne compile pas
4. lance une Runtime error

12. Le programme suivant :

Fichier Test.java :

```
package org.cnam.package1;

public class Test{
   public static void main(String[] args){
       int myint1 = 10;
       Integer myint2 = myint1;
       System.out.println(myint2);
   }
}
```

1. affiche : 10
2. n'affiche rien
3. ne compile
4. lance une Runtime error

13. Parmi les instructions suivantes lesquels sont valides :

    1. boolean b1 = 3 instanceof Integer;
    2. boolean b2 = new Integer(3) instanceof Integer;
    3. boolean b3 = Integer instanceof class;
    4. boolean b4 = Integer instanceof Class;

14. Le programme suivant :

Fichier Test.java :

```
package org.cnam.package1;

public class Test {
	public static void main(String[] args) {
		Integer myint1 = null;
		int myint2 = myint1;
		System.out.println(myint2);
	}
}
```

    1. affiche : 0
    2. n'affiche rien
    3. ne compile
    4. lance une Runtime error

15. Le programme suivant :

Fichier Test.java :

```
package org.cnam.package1;

public class Test {
	public static void main(String[] args) {
		int i = 10;
		double d1 = i;
		Double d2 = i;
		System.out.print(1) ;
	}
}
```

    1. affiche : 1
    2. n'affiche rien
    3. ne compile car la ligne `double d1 = i;` est invalide
    4. ne compile car la ligne `Double d2 = i;` est invalide
    5. lance une Runtime error

16. Le programme suivant :

Fichier Test.java :

```
package org.cnam.package1;

public class Test{
   public static void main(String[] args){
       Object myobj1 = new Object();
       Test myobj2 = (Test) myobj1;
       System.out.println("ok");
   }
}
```

    1. affiche : ok
    2. n'affiche rien
    3. ne compile pas
    4. lance une Runtime error

17. Le programme suivant :

Fichier Test.java :

```
package org.cnam.package1;

public class Test{
   public static void main(String[] args){
       Object myobj1 = new Object();
       Test myobj2 = myobj1;
       System.out.println("ok");
   }
}
```

1. affiche : ok
2. n'affiche rien
3. ne compile pas
4. lance une Runtime error

18. Le programme suivant :

Fichier Test.java :

```
package org.cnam.package1;

public class Test{
   public static void main(String[] args){
       Test myobj1 = new Test();
       Object myobj2 = myobj1;
       System.out.println("ok");
   }
}
```

1. affiche : ok
2. n'affiche rien
3. ne compile pas
4. lance une Runtime error

19. Parmi les affirmations suivantes, lesquelles sont valides :

1. Si deux méthodes d'une même classe possèdent un nom semblable mais différentes signatures, le nom de ces méthodes est dit surchargé.
2. Les méthodes surchargées peuvent provenir de l'héritage d'une superclasse.
3. float method(int i) et int method(int i) est une surcharge légale
4. int method(int i) et float method(float f) est une surcharge légale.

20. Le programme suivant :

Fichier Test.java :

```
package org.cnam.package1;

class Test1 {
      public int method(int i) {
            return 1;
      }
}

public class Test extends Test1 {
      public float method(int i) {
            return 1;
      }
}
```

1. compile
2. ne compile pas

21. Le programme suivant :

Fichier Test.java :

```
package org.cnam.package1;

class Test1 {
	public void method() {
		System.out.println("1");
	}
}
public class Test extends Test1 {
	public void method() {
		System.out.println("2");
	}
	static public void main(String[] args) {
		Test1 t = new Test();
		t.method();
	}
}
```

1. affiche : 1
2. affiche : 2
3. n’affiche rien
4. ne compile pas
5. lance une Runtime error