555F

AVIS IMPOR

1. Chacune des feuilles de votre copie do
   l'endroit indiqué «coller ici votre cod
2. Une copie d'examen comporte une
   chacune de vos feuilles, le code à b
3. Cette feuille d'examen est strictement
   Elle doit être écrite en noir et/ou ble
4. Le non respect de l'une de ces recor

ANT AUX ETUDIANTS

comporter une étiquette code à barres placée à
barres».
ule «feuille principale» et des «feuilles suites». Sur
es est obligatoire.
rsonnelle. Elle ne doit comporter aucun signe distinctif.

ndations peut faire attribuer la note ZERO à l'épreuve.

NOTE

Coller ici votre code à barre

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20

00	25	50	75

Institut Supérieur d'Informatique
et de Mathématiques de Monastir

Année universitaire : 2025-2026

EXAMEN

SESSION PRINCIPALE (JANVIER 2026)

Filière : L2 Informatique
Matière : Programmation Java
Enseignant : Akil ELKAMEL

Nombre de pages : 8
Durée : 1h30mn
Date : 07/01/2026

Documents non autorisés

Répondre directement sur la feuille d'examen.

Nom, prénom et signature de l'enseignant correcteur	Nombre de copies

Exercice 1 : (7 points)

1. Indiquer si l'affirmation est correcte ou non *(2 points)*

	Vrai	Faux
1. Une classe abstraite peut être instanciée.	☐	☐
2. Un champ **static** indique que le champ est commun à toutes les instances de la classe.	☐	☐
3. On peut définir un ou plusieurs constructeurs dans une classe abstraite.	☐	☐
4. Une interface peut hériter une ou plusieurs autres interfaces	☐	☐

2. Considérer les deux classes **`Animal`** et **`Chien`** où la méthode **`manger()`** a été redéfinie. Indiquer la/les réponse(s) correcte(s) si l'instruction proposée est insérée à l'emplacement précisé dans le programme principal *(1 point)*

```
class Animal {
    public void manger() {
        System.out.println("L'animal mange");
    }
}

class Chien extends Animal {
    public void manger() {
        System.out.println("Le chien mange");
    }
}

class Test {
    public static void main(String[] args) {
        Animal a = new Chien();
        Chien c = new Chien();
        /* Insérer l'instruction proposée ici */
    }
}
```

Instruction proposée	Erreur (Compilation)	Affiche **`L'animal mange`**	Affiche **`Le chien mange`**
`a.manger();`	☐	☐	☐
`d.manger();`	☐	☐	☐

3. Considérer les classes suivantes et indiquer la/les réponse(s) correcte(s) *(1 point)*

```
class Parent {
    public Parent() {
        System.out.print("A");
    }
}

class Enfant extends Parent {
    public Enfant(int x) {
        System.out.print("B");
    }
    public Enfant() {
        this(123);
        System.out.print("C");
    }
}

class Test {
    public static void main(String[] args) {
        new Enfant();
    }
}
```

☐ Une ou des erreurs de compilation ou d'exécution sont détectées

☐ Aucune erreur n'est détectée et le programme affiche : **ACB**

☐ Aucune erreur n'est détectée et le programme affiche : **ABC**

☐ Aucune erreur n'est détectée et le programme affiche : **BC**

☐ Aucune erreur n'est détectée et le programme affiche : **AC**

4. Considérer les classes suivantes et indiquer la/les réponses correcte(s) si l'instruction proposée est insérée à l'emplacement précisé dans le programme principal. *(1 point)*

```
class X {
    void do1() { }
}

class Y extends X {
    void do2() { }
}

class Test {
    public static void main(String[] args) {
        X x = new Y();
        /* Insérer l'instruction proposée ici */
    }
}
```

Instruction proposée	Erreur (Compilation)	Erreur (Exécution)	OK
`x.do2();`	☐	☐	☐
`((Y) x).do2();`	☐	☐	☐

5. Considérer les classes suivantes et indiquer la/les réponses correcte(s) si l'instruction proposée est insérée à l'emplacement précisé dans le programme principal. *(2 points)*

```
class A {
    void m() { }
}

class B extends A {
    void m() { }
}

class C {
    void m() { }
}
```

```
class Main {
    public static void main(String[] args) {
        A a = new A();
        B b = new B();
        C c = new C();
        /* Insérer l'instruction proposée ici */
    }
}
```

Instruction proposée	Erreur (Compilation)	Erreur (Exécution)	OK
`A x = a;`	☐	☐	☐
`A x = b;`	☐	☐	☐
`B x = a;`	☐	☐	☐
`B x = b;`	☐	☐	☐
`A x = c;`	☐	☐	☐
`A x = (A) b;`	☐	☐	☐
`B x = (B) a;`	☐	☐	☐
`C x = (C) b;`	☐	☐	☐

Exercice 2 : (13 points)

On veut créer une application en java pour manipuler des formes géométriques tout en respectant les principes de la programmation orientée objet.
On commence par fournir l'interface suivante :

```
interface FormeGeometrique {
        double perimetre();
        double surface();
}
```

Q1. Ecrire la classe **`Cercle`** qui possède un centre de type **`Point`** et un rayon de type **`double`**.
Supposons que la classe **`Point`** est implémentée conformément à la modélisation suivante :