

Ecole Supérieure Polytechnique

Département Génie Informatique

DUT II INFO/DST II INFO /DUT 2 TR /DST 2TR

Contrôle Continu

POO/JAVA

Durée 2 Heures

Exercice 1 : Quizz (9 points)

Ce QUIZZ permet de tester vos connaissances en programmation Java (1 point par question avec un malus de 0,5 point par question en cas de mauvaise réponse)

Question 1

Lors de la compilation d'un programme Java, le compilateur convertit le code source lisible par l'homme en code indépendant de la plateforme d'exécution qu'une Machine virtuelle Java peut comprendre. Comment s'appelle ce code indépendant de la plateforme d'exécution ?

- a) code machine
- b) code binaire
- c) code JVM (byte code)
- d) code java

Question 2

Quelle est la sortie de ce code Java ?

```
String[] tab = new String[4];
tab[0]="UN"; tab[1]="DEUX";
for(String s : tab){System.out.println(s);}
```

- a) Ce code affiche : UN DEUX null null
- b) Ce code affiche : UN DEUX 0 0
- c) Ce code affiche : UN DEUX

Question 3

Soit le code Java suivant :

```
public class Exemple
{
    ????? int var_x;
}

public class C
{
    public void traitement(Exemple ex)
    {
        ex.var_x = 100;
    }
}
```

Pour que ce code soit correct, il faut remplacer ????? par :

- a) private
- b) public
- c) protected

Question 4 : Quelle est la bonne façon de créer un objet appelé *unObjet* d'une classe nommée *UneClass* ?

- a) new unObjet = UneClass() ;
- b) class unObjet = null ;
- c) class unObjet = new class() ;
-) *UneClass unObjet = new UneClass() ;*

Question 5 : Comment peut-on connaître les paramètres à fournir pour créer un nouvel objet ?
En consultant :

- a) *les constructeurs*
- b) *les méthodes*
- c) les packages
- d) les librairies

Question 6

Quelle instruction parmi les suivantes est utilisée pour arrêter une boucle ?

- a) stop
- b) return
- c) exit
- d) *break*

Question 7

Qu'est-ce qui décrit le mieux le mot-clé new ?

- a) Crée une copie d'un objet existant et le traite comme un nouvel objet.
- b) *Instancie un nouvel objet.*
- c) Compare une référence d'objet à une nouvelle.
- d) Crée une nouvelle primitive.

Question 8 : Un attribut privé (private) d'une classe A peut être directement utilisé par une méthode d'une autre classe B qui appartient au même package que la classe A.

- a) Vrai
- b) *Faux*

Question 9 : Quelle est la sortie de ce code Java ?

```
public class MaClasse {  
    public static void main(String[] args) {  
        int x = 5;  
        System.out.println(x++);  
        System.out.println(++x);  
    }  
}
```

- a) 5 puis 7
- b) 6 puis 7
- c) 5, puis 6
- d) 6 puis 6

Exercice 2 : Sortie programme (4 Points)

Donner la valeur de v après chacune des 4 dernières instructions, dans le code suivant :

```
int a=5, b=4;  
String v;
```

v="7 6 "+a+b;	7 6 54
v="(53) "+(a+b);	(53) 9
v=" "+'a'+b;	a4
v=a+b+" 3 2 1";	9 3 2 1

Exercice 3: Classe (7 points)

On considère un système d'information pour la gestion d'articles dans un magasin. Dans ce système, chaque article est décrit par son nom, son prix et sa quantité en stock.

Dans ce système nous souhaitons afficher les détails de chaque article, permettre d'effectuer des ventes et d'approvisionner les articles qui sont en rupture de stock.

1. Ecrire en Java la classe Article en prenant le soin d'implémenter les accesseurs (get et set) et le constructeur sans paramètre (2 points)

```
public class Article {  
    private String nom;  
    private double prix;  
    private int stock;  
  
    // Constructeur sans parametre  
    public Article() {  
        this.nom = "";
```

```

        this.prix = 0.0;
        this.stock = 0;
    }

    // Accesseurs
    public String getNom() {
        return nom;
    }

    public void setNom(String nom) {
        this.nom = nom;
    }

    public double getPrix() {
        return prix;
    }

    public void setPrix(double prix) {
        this.prix = prix;
    }

    public int getStock() {
        return stock;
    }

    public void setStock(int stock) {
        this.stock = stock;
    }
}

```

2. Ecrire le corps de la méthode ***public static void afficherDetails(Article a)*** qui permet d'afficher les détails d'un article a déjà initialisé (2 points)

```

public static void afficherDetails(Article a) {
    System.out.println("Details de l'article :");
    System.out.println("Nom : " + a.getNom());
    System.out.println("Prix : " + a.getPrix());
    System.out.println("Quantite en stock : " + a.getStock());
}

```

3. Ecrire le corps de la méthode ***public static void vendre (Article a, int quantite)*** qui permet de vendre un certain nombre d'unités de l'article a en tenant compte de la quantité passée en paramètre et du stock (1.5 points)

```

public static void vendre(Article a, int quantite) {
    if (quantite <= 0) {

```

```

        System.out.println("La quantite doit etre superieure a 0.");
        return;
    }

    if (quantite > a.getStock()) {
        System.out.println("Quantite insuffisante en stock.");
        return;
    }

    a.setStock(a.getStock() - quantite);
    System.out.println("Vente effectuee. Nouvelle quantite en stock : " +
a.getStock());
}

```

4. Ecrire le corps de la méthode ***public static approvisionner (Article a, int quantite)*** qui permet d'approvisionner le stock de l'article a (1.5 points)

```

public static void approvisionner(Article a, int quantite) {
    if (quantite <= 0) {
        System.out.println("La quantite doit etre superieure a 0.");
        return;
    }

    a.setStock(a.getStock() + quantite);
    System.out.println("Approvisionnement effectue. Nouvelle quantite en
stock : " + a.getStock());
}
}

```