

Situation didactique

Cours : Programmation orientée objet Java (POO)

UE :

Enseignant : Msc. Ing. Paul AYANG

Crédit :

Année académique : 2022-2023

Programmation orientée objet Java (POO)

Etude de cas : Gestion des ventes de jus naturels

La SONAJUS est une société nationale de fabrication de jus naturels. Les jus fabriqués sont : jus d'orange, d'ananas, de mangue, de mandarine et d'oseille.

La société dispose de plusieurs agences de dépôts dans tout le pays. Chaque agence est constituée de plusieurs services (administratif, technique, commercial, informatique) et des vendeurs. Les vendeurs sont payés en fonction du nombre de jus en bouteille ou en cartons vendus, et ils sont divisés en deux catégories, les vendeurs à l'agence et les vendeurs ambulants. Au niveau de l'agence, chaque vendeur a un numéro matricule, et à la fin de chaque journée on enregistre le nombre de jus qu'il a vendu. Les jus sont vendus aux clients sur demande. Une demande a un numéro, une date et la quantité. Le nombre de jus acheté peut servir également au sondage.

La SONAJUS dispose également d'un parc automobile permettant d'assurer le transport de la boisson et du personnel. Le parc est constitué de plusieurs véhicules. Chaque véhicule est caractérisé par son numéro d'immatriculation, sa marque, son type. Pour le bon fonctionnement du parc, la société a créé un service qui emploie des agents d'entretien, des mécaniciens et des chauffeurs.

Pour cette nouvelle année, la direction générale de la SONAJUS a en projet, l'organisation d'un sondage pour connaître la popularité de ses produits à savoir le jus d'orange, d'ananas, de mangue, de mandarines et d'oseille. Pour ce sondage les produits sont numérotés 1, 2, 3, 4 et 5 respectivement. Les personnes interrogées doivent fournir pour réponse au questionnaire leur âge, leur activité (travailleur,

étudiant, chômeur) le numéro de deux produits préférés choisis dans la liste de cinq produits de l'entreprise. On suppose que toute personne interrogée s'intéresse soit à deux de ces produits, soit à aucun.

Le manque d'intérêt pour les produits proposés est représenté par le choix de numéro nul.

TAF :

Situation Didactique 1 : (Chapitre I : Généralités)

1. Qu'est-ce qu'un programme?
2. Donner l'historique de la programmation.
3. En quoi consiste la programmation de la machine analytique ?
4. Expliquer clairement la notion de compilation.
5. Donner les différentes étapes de la compilation.
6. Expliquer la notion d'interprète
7. Quelles différences entre langages compilés et interprétés ?
8. Quels sont les différents paradigmes de programmation ?
9. Citer les différents styles de programmation.
10. Donner quelques exemples de programmation déclarative.
11. C'est quoi la programmation impérative ?
12. Donner l'historique de la machine de Turing.
13. Qu'est qu'un langage langages machine ?
14. Présenter les langages d'assemblage (ou assembleurs)
15. Qu'est-ce qu'une programmation structurée ?
16. Qu'est-ce qu'une programmation procédurale ?
17. Quels sont les intérêts de la programmation procédurale ?
18. Qu'est-ce la programmation orientée objet ?
19. Ecrire un code objet pour afficher le message « bonjour TI »

Situation Didactique 2 : (Chapitre II : Langages de programmation)

1. Définir : Langage de programmation.
2. Qu'est-ce qu'un langage ?
3. C'est quoi la programmation ?
4. Présenter Le langage java.
5. Présenter le langage python.
6. Présenter le langage C++.

7. Présenter le langage C.
8. Présenter le langage JavaScript.
9. Présenter le langage C# .
10. Présenter le langage Ruby.
11. Présenter le langage PHP.
12. Le langage Objective-C.
13. Faites un résumé des paradigmes de programmation.
14. Comparer tous ces langages dans un tableau.
15. Ecrire un code pour afficher le menu suivant :

```

-----
|                MENU PRINCIPAL                |
|                1- Menu 1                        |
|                2- Menu 2                        |
|                3- Menu 3                        |
|                4- Menu 4                        |
|                5- Menu 5                        |
|                6- Faites votre choix            |
|                6- Exit                          |
-----

```

16. Ecrire le code pour exécuter le menu 1.
17. Ecrire le code pour exécuter le menu 2.

Situation Didactique 3 (Chapitre III : Les méthodes de programmation et bonnes pratiques de programmation)

1. Présenter les différentes étapes de développement d'un logiciel .
2. Quels sont les différents paradigmes de programmation que nous pouvons utiliser dans le cadre de ce cours? expliquer vos choix.
3. Méthodologie de développement logiciel.
4. Quelles sont les bonnes pratiques pour la lisibilité, utilisabilité d'un programme ?
5. Faites le tableau comparatif de la compilation vs interprétation.
6. Ecrire un code qui crée une classe article avec les attributs : référence, désignation, prix unitaire et méthodes : prix achat, prix de vente.
7. Proposer deux classes similaires à la classe article.
8. Proposer un code qui calcule le périmètre du cercle, rectangle, carré et du rectangle.

9. Proposer une classe mère pour ces figures géométriques.

Situation Didactique 4 (Chapitre IV : Généralités du langage Java.)

1. Quelles sont les différentes caractéristiques du langage Java ?
2. Quels sont les différents styles de programmation ?
3. Donner les avantages de la programmation objet.
4. C'est quoi la programmation modulaire ?
5. Expliquer les concepts d'abstraction et de spécialisation.
6. Ecrire un code objet qui effectuer les 4 opérations arithmétiques des nombres réels.
(en TP)
7. Proposer un code pour manipuler les nombres complexes.(en TP)

Situation Didactique 5 (Chapitre IV : Généralités du langage Java.)

1. C'est quoi le langage Java ?
2. Etablir clairement le lien entre la compilation, bytecode et JVM.
3. Définir : Le bytecode
4. Qu'est qu'une machine virtuelle (JVM).
5. Présenter les différentes plateformes java.
6. Expliquer le concept de variables en java.
7. Expliquer les notions de classes et objets en java.
8. Donner la structure d'une classe java.
9. Proposer un code pour résoudre une équation de 1^{er} degré à une inconnue. (en TP)
10. Proposer un code pour résoudre une équation de 2^{eme} degré à une inconnue. (en TP)

Situation Didactique 6 (Chapitre V : Structure générale d'un programme Java)

1. Quelles sont les différentes parties d'un programme java ?
2. Comment crée-on un paquetage ?
3. Comment insère-t-on les directives d'importation ?
4. Comment crée-t-on une classe ?
5. Quels sont les membres d'une classe ?
6. Comment reconnaître les blocs et les variables locales ?
7. Comment déclare-t-on une variable locale constante ? proposer un exemple de code.
8. Quel rapport y a-t-il entre les champs et les variables?
9. Expliquer clairement le concept d'accessibilité en POO.
10. Qu'est qu'un modificateur ?
11. Quelle est l'importance du concept d'accessibilité ?
12. Donner le rôle des champs et méthodes dans une classe java.

13. Comment se fait l'accès aux champs et aux méthodes ? illustrer par des exemples.
14. Comment se fait la création d'instances de classes ? illustrer par des exemples.
15. Implémenter le code suivant :

```
public class LocalVariable {
    private static double sum(double[] values) {
        double sum=0.0;
        for(double v:values) {
            sum+=v;
        } // v n'est plus accessible
        return sum;
    } // values et sum pas accessible
    public static void main(String[] args) {
        sum(new double[]{2,3,4,5});
    }
}
```

Expliquer les commentaires.

Situation Didactique 7 (Chapitre V : Structure générale d'un programme Java)

Soit le code java suivant :

```
public class Pixel {
    private int x;
    private int y;
    public Pixel(int x, int y) {
        this.x = x;
        this.y = y;
    }
    public boolean sameAs(Pixel p) {
        return (this.x==p.x) && (this.y==p.y);
    }
    public static void main(String[] args) {
        Pixel p0 = new Pixel(0,0);
        Pixel p1 = new Pixel(1,3);
        boolean b = p0.sameAs(p1); // false
    }
}

class OtherClass {
    public static void main(String[] args) {
        Pixel p0 = new Pixel(0,0);
        p0.sameAs(p); // true
        p0.x = 1; // error: x has private
                // access in Pixel
    }
}
```

1. identifier le Constructeur.
2. Comment crée-t-on un constructeur ?
3. Comment crée-t-on un autre constructeur dans une même classe ?
4. Ajouter un Champ constant dans cette classe.
5. Comment faire une surcharge de méthodes ?
6. Corriger l'erreur générée ci-dessus.

```

public class Pixel {
    private final int x;
    private int y;
    public Pixel(int x, int y) {
        this.x = x;
        this.y = y;
    }
    public Pixel() {
        // error: final field x may not
        // have been initialized
    }
    public Pixel(int v) {
        this(v, v);
    }
    public static void main(String[] a){
        Pixel p = new Pixel(1);
        p.x = 0; // error: final field x
        // cannot be assigned
    }
}

```

7. Corriger l'erreur ci-dessus.
8. Comment se passe la création d'instance de classe ?
9. Qu'est-ce que la surcharge de méthode ?
10. Quand est-ce qu'il faut utiliser la surcharge ?
11. Quand est-ce qu'il y a ambiguïté de surcharge ?
12. Qu'est-ce qui caractérisent les membres statiques ?
13. Pourquoi les messages ci-dessous sont affichés ?

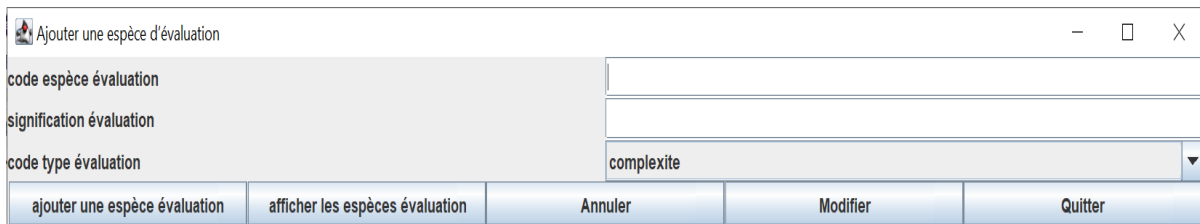
```

public static void main(String[] args) {
    System.out.println("Hi les geeks!"); // Hi les geeks!
        void java.io.PrintStream.println(String x)
    System.out.println('c'); // c
        void java.io.PrintStream.println(char x)
    System.out.println(2.5f); // 2.5
        void java.io.PrintStream.println(float x)
    Object o = new Object();
    System.out.println(o); // java.lang.Object@7a9664a1
        void java.io.PrintStream.println(Object x)
    int[] tab = new int[5];
    System.out.println(tab); // [I@27a8c4e7
        void java.io.PrintStream.println(Object x)
    Pixel p1 = new Pixel(1,3);
    System.out.println(p1); // Pixel@20cf2c80
        void java.io.PrintStream.println(Object x)
}

```

14. Comment crée-t-on la méthode exécutable dans une classe ?
15. Que fait la méthode println (Object x)

Situation Didactique 8 (Chapitre VI : Interfaces graphiques : Contrôle et propriétés, Contrôle et évènements.)



1. Créer cette fenêtre.
2. Que fait la classe JFrame ?
3. Identifier les différents objets graphiques sur cette interface.
4. Quel est le rôle des écouteurs d'événements ?
5. Qu'est-ce qu'une Classes anonymes ?
7. Créer les classes de paramétrage.
8. Créer les classes d'activités.
9. Faire interagir les différentes classes.

Situation Didactique 9 (Chapitre VI : Interfaces graphiques : Contrôle et propriétés, Contrôle et évènements.)

1. Quelles sont les instructions java permettant la connexion aux bases de données ?
2. Quelles sont les directives obligatoires à importer ?
3. Créer la base de données sous MYSQL pour la gestion des ventes des jus naturels.
4. Connecter les différentes interfaces graphiques à la base de données.
5. Donner la procédure pour la gestion des Fichiers en java.
6. Qu'est-ce que la Serialization ?
7. Qu'est-ce que le JDBC ?

Situation Didactique 10 (Chapitre VI : Interfaces graphiques : Contrôle et propriétés, Contrôle et évènements.)

1. Quelles sont les instructions qui permettent la création des menus en java?
2. Créer le Menu principal pour la gestion des ventes des jus naturel?
3. Créer les Sous menus pour chaque menu principal.
4. Connecter les différentes interfaces aux différents menus.
5. Ecrire le code java complet pour la gestion des ventes des jus naturel.