RÉPUBLIQUE TUNISIENNE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

DOSSIER DE CANDIDATURE

Soumis à

La commission Nationale de Recrutement des Maîtres assistants

Dans la discipline

Informatique

Travaux Dirigés Corrigés
Programmation Orienté Objet

Elaboré par :
Dr. Ghassen HAMDI
Laboratoire MARS, Université de Sousse

République Tunisienne
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale d'Ingénieurs de Sousse

Travaux Dirigés

Programmation Orienté Objet

Niveau : 3ème Génie Mécanique et Productique

Préparé Par : *HAMDI Ghassen*

Docteur en Sciences Informatique.

Année universitaire: 2021/2022

Email **: hamdighassan@gmail.com**

Travaux dirigés
« Programmation Orienté Objet »

Objectif(s) :

- S'habituer à travailler avec les notions de base de l'orienté objet à savoir les classes, les objets, instanciation, héritage, polymorphisme, surcharge…
- Manipuler des structures de données comme les tableaux, les chaines de caractères, les collections etc, en utilisant le langage Java

Bibliographie :

- www.cours-gratuit.com.
- firmforme.be.
- jmdoudoux.developpez.com.
- www.commentcamarche.net
- esmanick.unblog.fr

Sommaire

Travaux Dirigés 1

Eléments de base de la programmation JAVA

Exercice 1

Créer une classe Java dite « Exo1 » qui contient une méthode main() permettant de :

- Saluer une personne par son prénom, par exemple « Hello Ahmed ».
- Afficher son poids (nombre réel en kg) et sa taille (nombre réel en mètre).

***NB** : Toutes les valeurs sont à lire sur la ligne de commandes comme des arguments de la méthode main().*

Solution :

```
public class Exo1 {

    public static void main(String args[])
    {

        System.out.println("Bonjour "+args[0]);
        System.out.println("Votre poids est "+args[1]+" Kg");
        System.out.println("Vous mesurez "+args[2]+ " m");

    }

}
```

Exercice 2

1. Créer une classe Java appelée « Exo2 » qui calcule le maximum de deux valeurs réelles (Double) passées en arguments du main.
2. Utiliser l'opérateur conditionnel.
3. Le programme doit vérifier que l'utilisateur a saisi exactement deux arguments, sinon :

- Il affiche un message qui indique la syntaxe exacte d'appel du programme.

Solution :

```
import java.util.Scanner;

public class Exo2 {

    public static void main(String[] args) {

        if(args.length==2)
        {
            double a = Double.parseDouble(args[0]);
            double b = Double.parseDouble(args[1]);
            /*if(a>=b)
                System.out.print(a);
            else System.out.print(b);
            */
            double max= a>=b ? a : b;
            System.out.print(max);
            }
        else
            {
            System.out.println("Il faut exactement deux arguments selon la syntaxe suivante : ");

            System.out.println("java Ex02 val1 val2");

            }

    }

}
```

Exercice 3

1. Ecrire une classe Java « Exo3 » permettant d'appliquer une opération arithmétique sur deux valeurs numériques.
2. L'opérateur ainsi que les valeurs doivent être saisis à partir de la ligne de commandes selon le format suivant :
    - java nom_app operateur val1 val2
3. Les opérateurs à considérer sont ceux de l'addition, la soustraction, la multiplication et la division.
4. Considérer ces opérateurs sous forme de caractères ou de chaines de caractères.

Solution :

```java
public class Exo3 {

    public static void main(String[] args) {

        // Ici, on considère l'opérateur comme un caractère
        // chose plus pratique
        char operateur = args[0].charAt(0);
        int a = Integer.parseInt(args[1]);
        int b = Integer.parseInt(args[2]);

        switch(operateur) {
            case '+' -> System.out.println("Résultat  = "+ (a+b));
            case '-' -> System.out.println("Résultat  = "+ (a-b));
            case 'x' -> System.out.println("Résultat  = "+ a*b);
            case '/' -> System.out.println("Résultat  = "+ (float)a/b);
            default -> System.out.println("Opérateur invalide");
            }

        // Ici, on considère l'opérateur comme une chaine de caractères
        // moins pratique

            /*String operateur = args[0];
            int a = Integer.parseInt(args[1]);
            int b = Integer.parseInt(args[2]);

            switch(operateur) {
                case "plus" -> System.out.println("Résultat  = "+ (a+b));
                case "moins" -> System.out.println("Résultat  = "+ (a-b));
                case "fois" -> System.out.println("Résultat  = "+ a*b);
                case "sur" -> System.out.println("Résultat  = "+ (float)a/b);
                default -> System.out.println("Opérateur invalide");
            }*/
    }

}
```

Exercice 4

Créer une classe Java dite « Exo4 » qui contient une méthode main () qui a le même rôle que celle de l'exercice 1, mais toutes les valeurs sont à lire à partir du clavier à l'aide d'un objet Scanner.

1. Demander à l'utilisateur de saisir des informations (le prénom, le poids et la taille).
2. Calculer l'indice de masse corporelle (IMC) qui estime le poids optimal en corrélation avec la taille. (IMC = poids / taille²)
    - Le poids est en kg et la taille est en mètre
3. Afficher un message selon l'IMC trouvé :
    - IMC < 18,5 : insuffisance pondérale
    - 18,5 <= IMC < 25 : poids normal
    - 25 <= IMC < 30 : surpoids
    - IMC > =30 : obésité

Solution :

```java
import java.util.Scanner;

public class Exo4 {

    public static void main(String[] args) {
        Scanner clavier = new Scanner(System.in);

        System.out.print("Entrer votre nom : ");
        String prenom = clavier.nextLine();
        System.out.print("Entrer votre poids : ");
        double poids=clavier.nextDouble();
        System.out.print("Entrer votre taille : ");
        double taille=clavier.nextDouble();

        System.out.println("Bonjour "+prenom +" "+poids+" Kg "+taille+" metre");

        // Calcul de l'IMC
        double imc = poids/(taille*taille);
        //  double imc2 = poids/(Math.pow(taille, 2));

        System.out.println("IMC = "+imc);
        // Résultat selon l'IMC

        if(imc<18.5) System.out.println("Insuffisance pondérale");
            else if(imc<25) System.out.println("Poids normal");
                else if(imc<30) System.out.println("Surpoids");
                        else System.out.println("Obésité");

        clavier.close();

    }

}
```

Travaux Dirigés 2

Chaîne de caractère, tableau et collection

Exercice 1

Une chaîne de caractères chaîne est **carrée** s'il existe une chaîne ch telle que chaîne=chch, par exemple chercher et bonbon sont des chaînes carrées.

Une chaîne de caractères chaîne est dite **palindrome** si elle s'écrit d'un sens à l'autre, que ce soit de gauche à droite ou de droite à gauche. A titre d'exemple, elle et radar sont des palindromes.

1. Donnez une classe Java « **Exo1TD2** » qui nous permet de vérifier si une chaîne saisie par l'utilisateur est carrée ou palindrome.
    - Pour cet exercice, développer les méthodes booléennes « EstCarrée » et « EstPalindrome ».

Solution :

```
import java.util.Scanner;
public class Exo1TD2 {
public static boolean EstPalindrome(String mot){

        String motinverse="";
        for(int i = mot.length()-1; i>=0; i--)
        {
            motinverse=motinverse+mot.charAt(i);
        }

        if(mot.equals(motinverse)) return true;
            else return false;
    }

public static boolean EstCarree(String mot){

    if(mot.length()%2==0)
    {
        String moitie = mot.substring(0, (mot.length()/2));
        if(mot.equals(moitie+moitie)) return true;
    }
    return false;
}
public static void main(String[] args) {

    Scanner scanner = new Scanner( System.in );
    System.out.println("Donnez une chaine de caractères :");
    String mot = scanner.next();
    if(EstCarree(mot)) System.out.println("Ce mot est carré...");
    else if(EstPalindrome(mot)) System.out.println("Ce mot est palindrôme...");
        else System.out.println("Ce mot n'est ni carré ni palindrome !!!");

    scanner.close();
}
}
```

Exercice 2

Quelquefois, retrouver un mot de passe à partir d'une phrase clé est plus simple que retenir le mot de passe lui-même. D'où, l'une des techniques de génération de mots de passe est la technique de la phrase clé :

- Sélectionner un dicton, le titre d'un film, de chanson ou de livre, etc. Cette phrase servira de référence pour la création d'un mot de passe.
- Le mot de passe est dérivé de la phrase clé à l'aide d'une méthode que nous sélectionnons.

La procédure sélectionnée dans notre situation est la suivante : générer le mot de passe en extrayant les premières lettres des mots constituant la phrase, puis en leur concaténant les longueurs de ces mots.

- A titre d'exemples : « La vie en rose » donne le mot de passe suivant : « Lver2324 »
- « The Java Programming Language » donne le mot de passe suivant : « TJPL34118 »

- Ecrire une classe « **Exo2TD2** » qui, en lisant une phrase clé, génère un mot de passe en utilisant la méthode susmentionnée.

NB : *Utiliser la méthode prédéfinie « split » pour décomposer la phrase en chaines de caractères. Elle retourne un tableau composé des chaines contenues dans cette phrase.*

Solution :

```java
import java.util.Scanner;

public class Exo2TD2 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Veuillez saisir votre phrase clé : ");
        String phrase = sc.nextLine();

        String[] mots= phrase.split(" ");
        String initiales = "";
        String lestailles="";
        for(String mot : mots) {

            initiales= initiales+mot.charAt(0);
            lestailles=lestailles+mot.length();

        }

        System.out.println("Votre mot de passe est : "+initiales+lestailles);


    }

}
```

Exercice 3

1. Ecrire une classe Java « **Exo3TD2** » permettant de remplir un tableau de chaines de caractères, dont la taille est à saisir par l'utilisateur.

2. Puis, une fois rempli, il doit le parcourir, vérifier la présence de chaines palindromes et créer un autre tableau de booléens qui reflète exactement le contenu du premier tableau.

À titre d'illustration, si le tableau initial est comme suit :

exercice	java	radar	salut	elle

Le deuxième tableau sera :

false	false	true	false	true

Solution :

```
import java.util.Scanner;

public class Exo3TD2 {

    public static boolean palindrome(String mot){

        String motinverse="";
        for(int i = mot.length()-1; i>=0; i--)
        {
            motinverse=motinverse+mot.charAt(i);
        }

        if(mot.equals(motinverse)) return true;
            else return false;

    }

    public static void main(String[] args){

        Scanner scanner = new Scanner( System.in );
        System.out.print("Combien de mots allez-vous saisir ? ");
        int nb = scanner.nextInt();

        String tab_mots[] = new String[nb];
        boolean tab_bool[] = new boolean[nb];

        System.out.println("Saisissez vos "+nb+" mots : ");

        for(int i=0; i<=nb-1; i++)
        {
            tab_mots[i]=scanner.next();
        }

      for(int i=0; i<=nb-1; i++)
      {
          if(palindrome(tab_mots[i]))
              tab_bool[i]=true;
          else tab_bool[i]=false;
      }

      System.out.println("Tableau des mots tableau des booléens");
      for(int i=0; i<=nb-1; i++)
      {
          System.out.println(tab_mots[i]+" "+tab_bool[i]);
      }
      scanner.close();
  }

  }
```

Exercice 4

1. Ecrire une classe Java « **Exo4TD2** » qui demande à l'utilisateur de remplir un tableau de 5 entiers.
2. Ensuite, il procédera à l'affichage du contenu de ce même tableau dans l'ordre inverse.
3. Puis, il demandera de nouveau à l'utilisateur de remplir une liste (ArrayList) avec des chaines de caractères au choix.
4. Le programme arrête d'accepter des saisies dès qu'il lit la chaine « Stop ».

5. Enfin, il affichera aussi le contenu de cette liste dans le sens inverse.

Solution :

```java
import java.util.ArrayList;
import java.util.Scanner;

public class Exo4TD2 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        int tab[]=new int[5];

        System.out.println("Remplissez un tableau de 5 entiers :");

        for(int i=0; i<tab.length; i++)
            tab[i]=sc.nextInt();

        System.out.println("Le tableau inverse est :");
        for(int i=tab.length-1; i>-1; i--)
            System.out.println(tab[i]);

        System.out.println("Remplissez une liste de chaines de caractères :");
        System.out.println("Pour arrêter la saisie, écrivez STOP...");

        ArrayList<String> liste = new ArrayList<String>();
        String entree = sc.next();
        while(!entree.equals("STOP"))
        {
            liste.add(entree);
            entree=sc.next();
        }

        System.out.println("La liste inverse est :");
        for(int i =liste.size()-1; i>-1; i--)
            System.out.println(liste.get(i));
    }

}
```

Travaux Dirigés 3

La Programmation Orienté Objet avec JAVA

Exercice 1

1. Essayer de développer une classe **Etudiant** qui a :

   - Attributs publics de type String : **nom** et **dnaiss** : « jj/mm/aaaa ».
   - un constructeur public sans paramètres.
   - Un deuxième constructeur acceptant deux paramètres servant à initialiser les attributs d'un étudiant ;
   - Une méthode publique ***travailler()*** affichant un message qui signale que l'étudiant s'est mis au travail.
   - Une méthode publique ***seReposer()*** affichant un message qui signale que l'étudiant se repose.
   - Une méthode publique ***calculerAge()*** qui sert à trouver l'âge d'un étudiant à partir de sa date de naissance et qui se base sur l'année actuelle.

NB : récupérer l'année à partir de la date de naissance, la convertir en entier puis la soustraire de l'année courante, à extraire aussi à partir de la date courante obtenue à l'aide de l'instruction : Date d = new Date() ;

2. Développer une classe de test appelé **TestEtudiant** qui contient une méthode main() qui :

   - crée un étudiant avec ses 2 informations ;
   - invoque la méthode travailler() de l'étudiant ;
   - invoque la méthode seReposer() de l'étudiant ;
   - affiche l'âge de cet étudiant.

Solution :

```java
import java.util.Date;

public class Etudiant {

    public String nom;
    public String dnaiss;

    public Etudiant()
    {
        nom="";
        dnaiss="";
    }

    public Etudiant(String n, String d)
    {
        nom=n;
        dnaiss=d;
    }
    // OU
/*  public Etudiant(String nom, String dnaiss)
    {
        this.nom=nom;
        this.dnaiss=dnaiss;
    }
    */
    public void travailler()
    {
        System.out.println("L'étudiant "+nom +" s'est mis à travailler");
    }

    public void seReposer()
    {
        System.out.println("L'étudiant "+nom +" se repose");
    }


    public int calculerAge()
    {

        // On décompose la date de naissance en trois chaines selon le séparateur "/"
        String[] dateDecomposee = dnaiss.split("/");
        // L'année est la troisième chaine, donc d'indice 2
        int anneeNaissance = Integer.parseInt(dateDecomposee[2]);

        Date d=new Date();
        // la date d aura la forme suivante "Mon Jan 18 17:03:37 GMT+01:00 2021"
        // où l'année est aussi mise en fin de la chaine

        // Donc, on décompose cette date (après l'avoir convertie en chaine de caractères) sur la base
        // du séparateur ESPACE (" ")
        String[] dateActuelleDecomposee = d.toString().split(" ");
        int anneeCourante = Integer.parseInt(dateActuelleDecomposee[dateActuelleDecomposee.length-1]);

        return anneeCourante-anneeNaissance;

    }


}
```

```java
public class TestEtudiant {

    public static void main(String[] args) {

        // utilisation du 1er constructeur
        Etudiant x = new Etudiant();
        x.nom="Yacouba";
        x.dnaiss="12/03/2005";

        // utilisation du 2ème constructeur
        Etudiant y = new Etudiant("Florence", "25/08/2020");

        // appeler la méthode travailler sur x
        x.travailler();

        // appeler la méthode travailler sur y
        y.travailler();

        // appeler la méthode seReposer sur x
        x.seReposer();

        // appeler la méthode seReposer sur y
        y.seReposer();

        System.out.println("L'age de "+x.nom+ " est : "+x.calculerAge());
        System.out.println("L'age de "+y.nom+ " est : "+y.calculerAge());

    }

}
```

Exercice 2

1. Reprendre les mêmes classes de l'exercice 1 et transformer les attributs publics en attributs privés.
    - Des erreurs vont apparaitre dans la classe **TestEtudiant**. Ceci est dû au fait que cette dernière n'a pas le droit d'accéder aux attributs privés de la classe **Etudiant**.
2. Pour pallier ce problème, ajouter dans la classe **Etudiant** un accesseur et un mutateur pour chaque attribut.
    - Le mutateur de ***dnaiss*** doit veiller que le nouvel âge est valable pour un étudiant (on suppose qu'un étudiant doit avoir entre 18 et 30 ans).
3. Utiliser ces derniers pour manipuler les attributs privés dans **TestEtudiant**.
4. Permettre à **TestEtudiant** de demander une nouvelle valeur pour la date de naissance de l'étudiant et de trouver le nouvel âge ou de refuser l'opération.

Solution :

```
public class TestStudent {

public static void main(String[] args) {

   // utilisation du 1er constructeur
     Student x = new Student();
     x.setNom("Yacouba");
     x.setDnaiss("12/03/1998");
  // utilisation du 2ème constructeur
     Student y = new Student("Florence", "25/08/2000");
  // appeler la méthode travailler sur x
     x.travailler();
  // appeler la méthode travailler sur y
     y.travailler();
  // appeler la méthode seReposer sur x
     x.seReposer();
  // appeler la méthode seReposer sur y
     y.seReposer();
    System.out.println("L'age de "+x.getNom()+ " est :
     "+x.calculerAge());
    System.out.println("L'age de "+y.getNom()+ " est :
     "+y.calculerAge());
     y.setDnaiss("12/03/2002");
    System.out.println("Le nouvel âge de "+y.getNom()+ " est :
     "+y.calculerAge());
     x.setDnaiss("02/10/1900");
    System.out.println("Le nouvel âge de "+x.getNom()+ " est :
     "+x.calculerAge());
}

}
```

Travaux Dirigés 4

Héritage avec JAVA

Exercice

1. Développer la classe Crayon caractérisée par :

   - deux attributs privés (double) ***épaisseur*** et ***longueur***,
   - un constructeur qui nous permet d'initialiser les attributs
   - les mutateurs et les accesseurs nécessaires
   - une méthode **affiche()** qui nous permet d'afficher les caractéristiques d'un crayon.

2. Développer la classe CrayonCouleur héritant de la classe Crayon et qui est caractérisée aussi par :

   - un attribut supplémentaire privé ***couleur*** (String).
   - un constructeur qui nous permet d'initialiser les attributs
   - le mutateur et l'accesseur nécessaires
   - une méthode appelée **changeCaracteristiques()** qui n'a pas de valeur de retour et qui en faisant appel aux autres méthodes modifie les valeurs de tous les attributs (longueur, épaisseur et couleur).
   - une méthode **affiche()** qui permet de présenter les caractéristiques d'un crayon de couleur en redéfinissant et utilisant la méthode d'affichage d'un crayon ordinaire.

3. Enfin, développer la classe TestCrayon qui contient une méthode main() pour tester ces classes :

   - créer dedans un crayon et crayon de couleur, changer l'épaisseur du premier et
   - la couleur du second puis afficher leurs informations.

Solution :

```
public class Crayon {

    private double epaisseur;
    private double longueur;

    public Crayon(double epaisseur, double longueur) {
        this.epaisseur=epaisseur;
        this.longueur=longueur;
    }

    public double getEpaisseur() {
        return epaisseur;
    }

    public double getLongueur() {
        return epaisseur;
    }

    public void setEpaisseur(double ep) {
        epaisseur=ep;
    }

    public void setLongueur(double lng) {
        longueur=lng;
    }

    public void affiche() {
        System.out.println("Epaisseur = "+epaisseur + " ; Longueur = "+ longueur);
    }

}
```

```
public class CrayonCouleur extends Crayon {

    private String couleur;

    public CrayonCouleur(double epaisseur, double longueur, String couleur) {
        super(epaisseur, longueur);
        this.couleur=couleur;
    }

    public String getCouleur() {
        return couleur;
    }

    public void setCouleur(String coul) {
        couleur=coul;
    }

    public void changeCaracteristiques(double newEp, double newLong, String newCoul) {
        setEpaisseur(newEp);
        setLongueur(newLong);
        setCouleur(newCoul);
    }

    public void affiche() {
        super.affiche();
        System.out.println("Couleur = "+ couleur);
    }
}
```

```java
public class TestCrayons {

    public static void main(String[] args) {

        Crayon cr = new Crayon(1, 10);
        CrayonCouleur crc1 = new CrayonCouleur(1.2, 11, "jaune");
        CrayonCouleur crc2 = new CrayonCouleur(0.8, 9, "rouge");

        cr.affiche();
        crc1.affiche();
        crc2.affiche();

        cr.setEpaisseur(0.9);
        crc1.setCouleur("mauve");
        crc2.changeCaracteristiques(0.5, 7, "marron");

        cr.affiche();
        crc1.affiche();
        crc2.affiche();

    }

}
```