**Compte rendu : TD-TPS\_3&4**

**Stitou Ftima**

**Groupe 1**

**S6**

**Exercice 1 :**

import java.util.Scanner;

public class Livre {

private String titre;

private String auteur;

private double prix;

public Livre(String titre, String auteur, double prix) {

this.titre = titre;

this.auteur = auteur;

this.prix = prix;

}

//Getters

public String getTitre() {

return titre;

}

public String getAuteur() {

return auteur;

}

public double getPrix() {

return prix;

}

//Setters

public void setTitre(String titre) {

this.titre = titre;}

public void setAuteur(String auteur) {

this.auteur = auteur;

}

public void setPrix(double prix) {

this.prix = prix;

}

//constructeur surchargé

public Livre() {

Scanner input = new Scanner(System.in);

System.out.print("Titre : ");

this.titre = input.nextLine();

System.out.print("Auteur : ");

this.auteur = input.nextLine();

System.out.print("Prix : ");

this.prix = input.nextDouble();

}

//la méthode Afficher ( ) permettant d’afficher les informations du livre en cours

public void afficher() {

System.out.println("Titre : " + titre);

System.out.println("Auteur : " + auteur);

System.out.println("Prix : " + prix + "€");

}

public static void main(String[] args) {

Livre livre1 = new Livre("Le Petit Prince", "Antoine de Saint-Exupéry", 12.99);

Livre livre2 = new Livre();

System.out.println("Livre 1 :");

livre1.afficher();

System.out.println("Livre 2 :");

livre2.afficher();

}}

Le resultat après le teste :



**Exercice 2 :**

import java.util.Calendar;

import java.util.Scanner;

public class Employe {

private String matricule;

private String nom;

private String prenom;

private int anneeNaissance;

private int anneeEmbauche;

private double salaire;

// Constructeur par défaut

public Employe() {}

// Constructeur surchargé avec saisie utilisateur

public Employe(String matricule, String nom, String prenom, int anneeNaissance, int anneeEmbauche, double salaire) {

this.matricule = matricule;

this.nom = nom;

this.prenom = prenom;

this.anneeNaissance = anneeNaissance;

this.anneeEmbauche = anneeEmbauche;

this.salaire = salaire;

}

// Accesseurs et Mutateurs

public String getMatricule() {

return matricule;

}

public void setMatricule(String matricule) {

this.matricule = matricule;

}

public String getNom() {

return nom;

}

public void setNom(String nom) {

this.nom = nom;

}

public String getPrenom() {

return prenom;

}

public void setPrenom(String prenom) {

this.prenom = prenom;

}

public int getAnneeNaissance() {

return anneeNaissance;

}

public void setAnneeNaissance(int anneeNaissance) {

this.anneeNaissance = anneeNaissance;

}

public int getAnneeEmbauche() {

return anneeEmbauche;

}

public void setAnneeEmbauche(int anneeEmbauche) {

this.anneeEmbauche = anneeEmbauche;

}

public double getSalaire() {

return salaire;

}

public void setSalaire(double salaire) {

this.salaire = salaire;

}

// Méthode pour calculer l'âge de l'employé

public int getAge() {

return (Calendar.getInstance().get(Calendar.YEAR) - anneeNaissance);

}

// Méthode pour calculer l'ancienneté de l'employé

public int getAnciennete() {

return (Calendar.getInstance().get(Calendar.YEAR) - anneeEmbauche);

}

// Méthode pour augmenter le salaire en fonction de l'ancienneté

public void AugmentationDuSalaire() {

int anciennete = getAnciennete();

if (anciennete < 5) {

salaire \*= 1.02;

} else if (anciennete < 10) {

salaire \*= 1.05;

} else {

salaire \*= 1.1;

}

}

// Méthode pour afficher les informations de l'employé

public void AfficherEmploye() {

System.out.println("Matricule: " + matricule);

System.out.println("Nom: " + nom);

System.out.println("Prénom: " + prenom);

System.out.println("Année de naissance: " + anneeNaissance);

System.out.println("Année d'embauche: " + anneeEmbauche);

System.out.println("Salaire: " + salaire);

System.out.println("Age : " + getAge() + " ans");

System.out.println("Ancienneté : " + getAnciennete() + " ans");}

//le test

public static void main(String[] args) {

Employe emp1 = new Employe("001", "Durand", "Jean", 1985, 2010, 2500.0);

emp1.AfficherEmploye();

System.out.println("Age : " + emp1.getAge());

System.out.println("Ancienneté : " + emp1.getAnciennete() + " ans");

emp1.AugmentationDuSalaire();

emp1.AfficherEmploye();

}}

Le resultat après le teste :



**Exercice 3 :**

public class Document{

static private int numeroSuivant = 0;

static private int getID() { return ++numeroSuivant; }

private int numero;

private String titre;

public Document(String titre) {

this.numero = getID();

this.titre = titre; }

public int getNumero(){ return numero; }

public String getTitre(){ return titre; } }

// la calsse Dictionnaire hérité de Document

public class Dictionnaire extends Document {

private int nbTomes;

private String langue;

public Dictionnaire(String titre, int nbTomes, String langue)

{ super(titre);

this.nbTomes = nbTomes;

this.langue = langue; }

public int getNombreDeTomes() { return nbTomes; }

public String getLangue() { return langue; } }

// la calsse livre hérité de Document

public class Livre extends Document{

private String auteur;

private int nbPages;

//constructeur surchargé

public Livre(String titre, String auteur, int nbPages) {

super(titre);

this.nbPages = nbPages;

this.auteur = auteur; }

// Getters & Setterrs

public String getAuteur() { return auteur; }

public int getNombreDePages() { return nbPages; } }

//la calasse Bibliotheque

public class Bibliotheque{

public static void main(String[] argv){

Document d = new Document("Configurer son compte utilisateur");

Livre l = new Livre("La communication sous unix","J-M Rifflet",799);

Dictionnaire di = new Dictionnaire("Man pages",4,"Anglais");

System.out.println(d.getTitre()) ;

System.out.println(l.getTitre()) ;

System.out.println(di.getTitre()) ; } }

import java.util.Enumeration; // utile pour l’ajout de la question suivante

import java.util.Vector;

//classe ListeDeDocument

public class ListeDeDocuments {

private Vector lesDocuments;

//constructeur surchargé

public ListeDeDocuments() { lesDocuments = new Vector(); }

public void ajouteDocument(Document unDocument) { lesDocuments.add(unDocument); } }

public void tousLesAuteurs() {

Enumeration e = lesDocuments.elements();

Document unDocument;

while( e.hasMoreElements() ) { unDocument = (Document)e.nextElement(); System.out.print(unDocument.getNumero());

if (unDocument instanceof Livre)

System.out.println( " " + ((Livre)unDocument).getAuteur());

else

System.out.println() ; } }

//la methode toString() dans la classe Document

public String toString() { return ("numero: "+numero+" titre: "+titre); }

//Dans Livre :

public String toString(){ return (super.toString()+" auteur: "+getAuteur()+" pages: "+getNombreDePages()); }

//Dans Dictionnaire :

public String toString() { return (super.toString()+" tomes: "+getNombreDeTomes()+" langue: "+getLangue()); }

//Dans ListeDeDocuments :

public void tousLesDocuments() {

Enumeration e = lesDocuments.elements();

Document unDocument;

while (e.hasMoreElements()){

unDocument = (Document)e.nextElement();

System.out.println(unDocument); } }

//la classe Bibliotheque afin de tester la classe ListeDeDocuments:

public class Bibliotheque{

public static void main(String[] argv) {

Document d = new Document("Configurer son compte utilisateur");

Livre l = new Livre("La communication sous unix","J-M Rifflet" ,799);

Dictionnaire di = new Dictionnaire("Man pages",4,"Anglais");

System.out.println(d.getTitre());

System.out.println(l.getTitre()); System.out.println(di.getTitre());

ListeDeDocuments ld = new ListeDeDocuments();

ld.ajouteDocument(d);

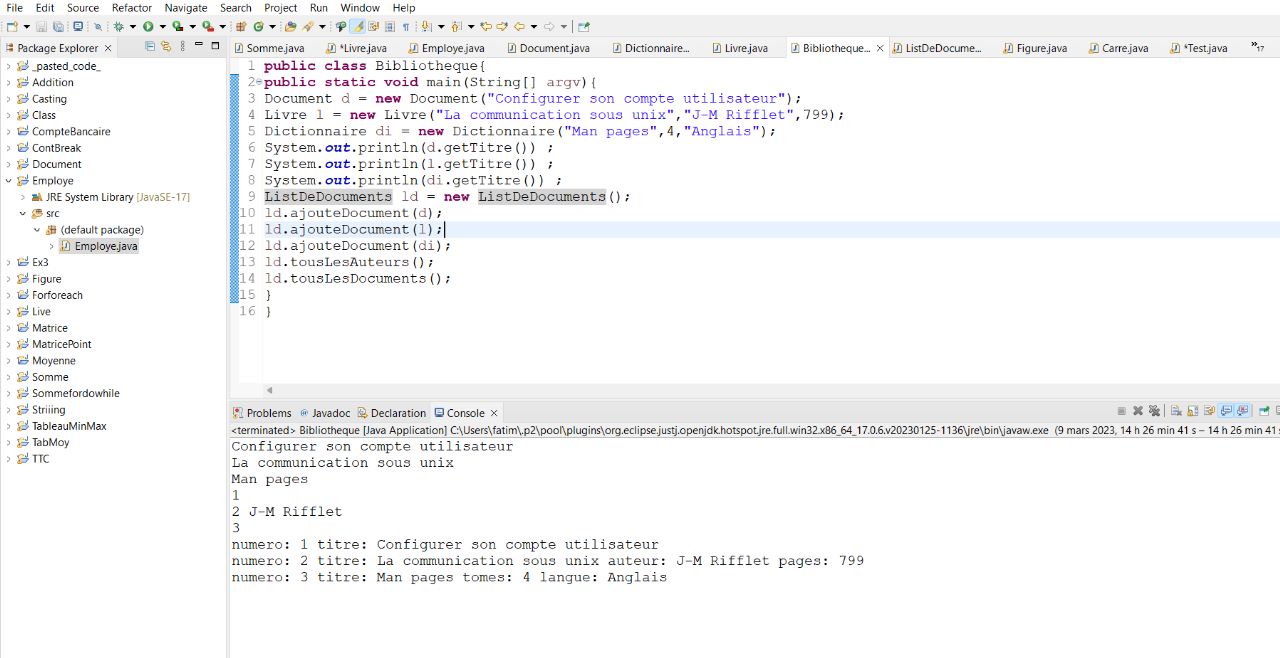
ld.ajouteDocument(l);

ld.ajouteDocument(di);

ld.tousLesAuteurs();

ld.tousLesDocuments(); } }

Le resultat après le teste :



**Exercice 4 :**

//Fichier Figure.java

import java.util.Vector ;

public class Figure {

private static Vector instances = new Vector();

private int abscisse;

private int ordonnee;

private int couleur;

public Figure( int abscisse , int ordonnee , int couleur )

{ this.abscisse = abscisse;

this.ordonnee = ordonnee;

this.couleur = couleur;

instances.add(this); }

public static Vector getInstances() { return instances ; }

public String toString(){ return (abscisse+" "+ordonnee+" "+couleur); } }

// la classe hérité de Figure

public class Carre extends Figure {

private int cote ;

//constructeur surchargé

public Carre(int abscisse, int ordonnee, int couleur, int cote) { super(abscisse,ordonnee,couleur);

this.cote = cote; }

public static Vector getInstances() {

int nCarre = 0;

Vector instancesCarre = new Vector();

Vector instances = Figure.getInstances();

Enumeration e = instances.elements();

Figure uneFigure;

while(e.hasMoreElements()) { uneFigure = (Figure)e.nextElement();

if (uneFigure instanceof Carre)

instancesCarre.add(uneFigure); }

return instancesCarre; }

public String toString(){

return (super.toString()+" "+cote); } }

//Fichier Test.java

import java.util.Vector ;

import java.util.Enumeration ;

//le test

public class Test {

public static void main(String[] argv) {

Figure f1 = new Figure(1,1,1);

Figure f2 = new Figure(2,2,2);

Carre c3 = new Carre(3,3,3,3);

Figure f4 = new Figure(4,4,4);

Carre c5 = new Carre(5,5,5,5);

System.out.println("Liste des figures") ;

Enumeration e = Figure.getInstances().elements() ;

while(e.hasMoreElements())

{ System.out.println(e.nextElement()) ; }

System.out.println("Liste des carres") ; e = Carre.getInstances().elements() ; while(e.hasMoreElements())

{ System.out.println(e.nextElement()) ; } } }

Le resultat après le teste :

