

Programmation Orientée Objets et Evénementielle

DUT SRC IUT Meaux année 2008-Module M2.23

TD 1 et 2 Java : La prise en main de Java et Eclipse, Eléments du langage

Exercice 1 : Objectifs : Prise en main d'Eclipse. Durée 1H

Ecrire en Java un programme qui édite sur la console « Bonjour Monde ». A cette fin on utilisera Eclipse pour créer un projet «PremierProjet» et dans ce projet un package «TP1_Exercice1» qui contiendra la classe « BonjourMonde ».

Exercice 2 : Recupérer la doc jdk-6 et commencer à s'en servir

A cette occasion on importera la documentation jdk-6-doc à l'adresse :

<http://java.sun.com/javase/6/download.jsp#docs>

Exercice 3 : Etude détaillée d'une classe

Durée : 1H

Le programme suivant définit la classe Personne comme étant structurée par les propriétés (ou variables d'instances) nom, annee_naissance et salaire, et ayant pour méthodes Personne(), affiche() et calcul_age().

La méthode Personne () - qui porte le nom de la classe - a un rôle particulier : contribuer à la construction des objets.

La méthode affiche() doit permettre à chaque objet d'afficher son propre état.

La méthode calcul_age() doit faire en sorte que chaque objet calcule lui-même son âge.

Les objets implémentés dans la classe person1 qui sert de classe de test, sont les suivants :

p1 avec nom= "dupont", annee_naissance=1950, salaire=1500

p2 avec nom= "mercier", annee_naissance=1962, salaire=2300

The screenshot shows the Eclipse IDE with the following components:

- Package Explorer:** Shows a project structure with a package 'tp1et2_Ex3_Classe' containing a class 'person1.java'.
- Code Editor:** Displays the source code for 'person1.java'. The code defines a 'Personne' class with private attributes 'nom', 'annee_naissance', and 'salaire'. It includes a constructor 'Personne(String n, int a, int s)', a method 'affiche()' that prints the object's state, and a method 'calcul_age()' that calculates the age based on the current year (2008) and the birth year. A 'person1' class contains a 'main' method that creates two 'Personne' objects (p1 and p2) and calls their 'affiche()' and 'calcul_age()' methods.
- Outline:** Shows the class hierarchy and methods for 'Personne' and 'person1'.
- Console:** Shows the output of the program execution: 'dupont 1950 1500', 'mercier 1962 2300', 'age = 58', and 'age = 46'.

- 1) Recopier, compiler et exécuter ce programme.
- 2) Expliquer la structuration du programme et les différents aspects de la classe Personne.
- 3) Modifier ce programme pour qu'il crée d'autres objets p3, p4....

Exercice 4 : Débogage d'un programme

Durée : 1H

Soit le programme suivant qui définit une classe Salarie

```
package tplet2_Ex4_Salarie;
import java.io.* ;
class Salarie {
    String nom ;
    string entreprise;
    int salaire;

    public void Salarie(string n, string e, int s) {
        Nom=n;
        Entreprise = e;
        Salaire = s;
    }

    public void affiche() {
        System.out.println(nom+"      "+entreprise+"      "+salaire);
    }
}

public class salarieTest{
    public void main(string args[]){
        Salarie s1 = new Salarie("Martin", "EDF", 1500);
        P1.affiche();
    }
}
```

- 1) Quelles variables structurent cette classe ?
- 2) Cette classe possède également un constructeur et une méthode. Découvrez les.
- 3) Il y a des erreurs dans le code ci-dessous de ce programme Trouvez les (sans l'aide de Eclipse dans un premier temps) et corrigez les.
- 4) Comment apparaissent les erreurs sous l'environnement Eclipse. Utilisez l'aide pour corriger les erreurs. Comment faites-vous ?
- 5) Commentez les erreurs et faites marcher ce programme.

Exercice 5 : Calculs élémentaires. Variables locales et variables d'instance

Durée : 1H

Soit un ensemble d'étudiants. Ils sont caractérisés par les 7 informations suivantes : nom, 4 notes, la moyenne de ces notes, leur résultat (admis ou non admis selon que la moyenne est $<$ ou $>$ à 10).

- 1) Définissez la classe Etudiant, en la dotant des méthodes nécessaires pour construire les objets « étudiant » et leur donner les moyens de calculer leurs moyennes et leurs résultats.
- 2) Déclarez une variable moyenne dans la méthode qui calcule la moyenne. Que se passe-t-il à l'exécution ? Pourquoi ?

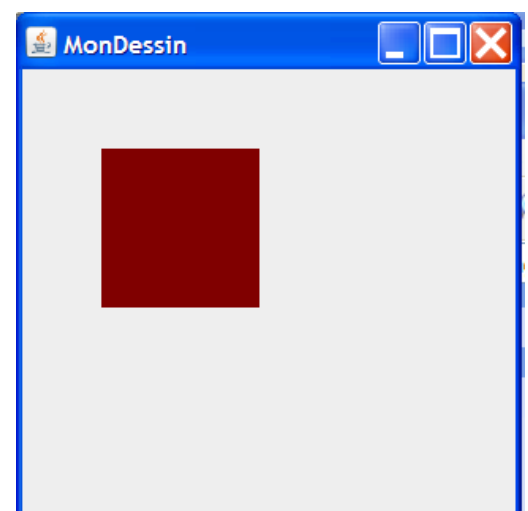
Exercice 6 : Dessiner. La classe Graphics, signification du terme « this », un peu de Swing

Durée : 1H

1) Le programme Java ci-dessous donne une base de dessin 2D dans une fenêtre carrée de `taille` pixels. Tapez le et exécutez le.

```
package tplet2_Ex6_dessinBase;
import javax.swing.*;
import java.awt.*;

class cadreDessin extends JFrame {
    private int taille = 320;
    //constructeur
    public cadreDessin() {
        this.setTitle("MonDessin");
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setBounds(0,0,taille,taille);
        // le panneau de dessin à installer dans la fenêtre
        this.getContentPane().add(new ZoneDessin(taille));
        this.setVisible(true);
    } // fin constructeur
} // fin de la classe cadreDessin
```

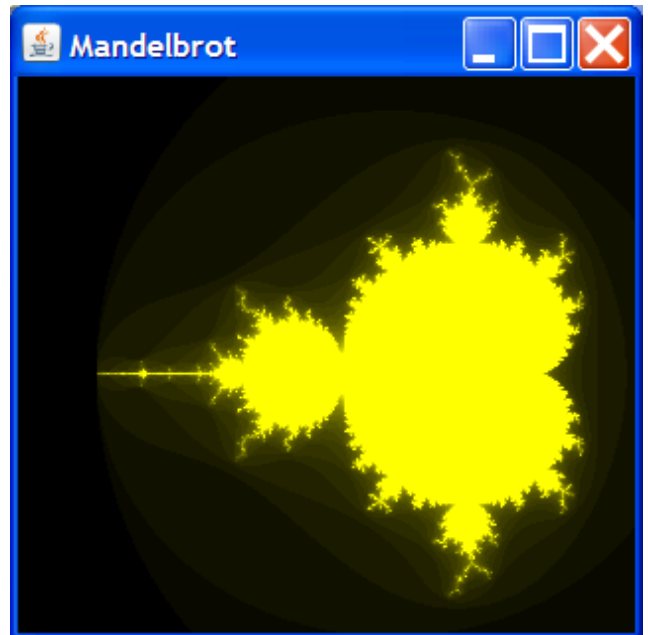


```

class ZoneDessin extends JPanel {
private int taille;
public ZoneDessin(int taille) {
this.taille= taille;
}
public void paintComponent(Graphics g) {
//Faites votre dessin avec le pinceau g et éventuellement d'autres paramètres à votre choix
monDessin(50, 100, g);
}
// Méthode de dessin à réaliser
public void monDessin(int x, int l, Graphics g) {
// dessine un carré avec une couleur rouge
g.setColor(new Color(0.5f, 0.0f, 0));
g.fillRect(x,x,l,l);
}
}

public class baseDessin {
public static void main(String args []){
JFrame FenetreDessin = new cadreDessin();
}
}

```



2) Commentez simplement ce programme. Ses caractéristiques seront revues en détail lors des TP sur Swing.

Recherchez la classe Graphics , la classe Color dans le package java.awt. Examinez les méthodes disponibles dans la classe Graphics. Expliquez ce que fait la méthode fillRect(...).

Expliquer ce que fait la classe cadreDessin. Quelle est l'interprétation du terme this ?

3) Dessin du fractal de Mandelbrot :

L'ensemble de Mandelbrot M est un fractal obtenu par le processus itératif suivant. En chaque point C du plan complexe on définit la suite : $Z_{n+1}=Z_n^2+C$ avec $Z_0=0$. Deux cas se rencontrent :

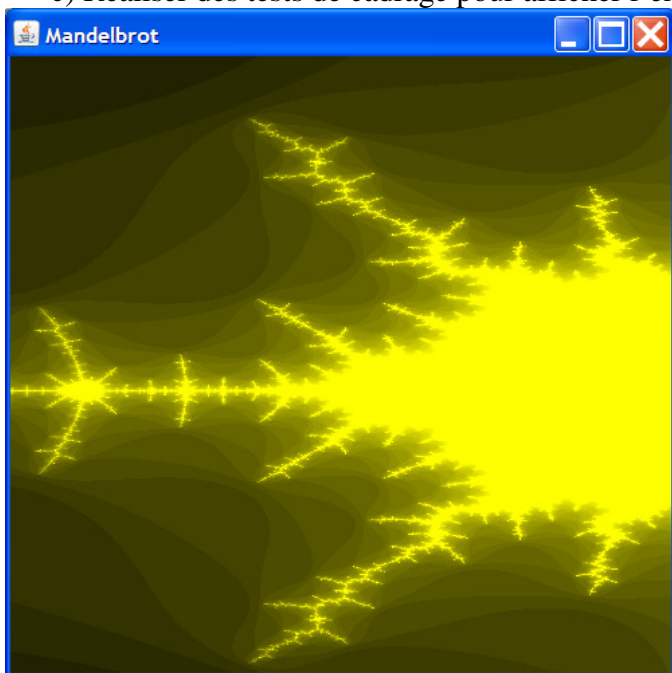
(a) $|Z_n| < \infty \forall n$: les éléments de la suite Z_n restent à distance finie de l'origine : le point C appartient à l'ensemble de Mandelbrot M.

(b) $|Z_n| \rightarrow \infty$: ce qui signifie que les éléments de la suite Z_n s'éloignent à l'infini : C n'appartient pas à l'ensemble de Mandelbrot M

a) Détailler le pseudo-code de la fonction dessinant le fractal de Mandelbrot (on considérera que l'infini commence à $n=30$).

b) Ecrire cette méthode en Java et l'inclure dans le programme de dessin précédent en lieu et place de la méthode monDessin().

c) Réaliser des tests de cadrage pour afficher l'ensemble M complet ou pour agrandir un détail.



d) Comment interprétez vous le dégradé de jaune ?

Exercice 7 : Utilisation de la console pour lire et écrire, chaînes de caractères

1) Le programme suivant permet de lire une ligne à la console. On souhaite éditer cette ligne en séparant chaque caractère par une barre verticale « | ». C'est l'objectif de la méthode `edition(String line)`;

Pour écrire cette méthode, on dispose dans la classe `String` de deux méthodes intéressantes : la méthode `charAt()` et la méthode `length()`

Aller chercher dans la documentation ces deux méthodes et utilisez les pour lire caractère par caractère la chaîne `line`. Editer alors chaque caractère précédé du caractère « | » dans la méthode `edition()`.

```
package tp1et2_Ex7_code0;
import java.io.*;

public class LireConsole {
    public static void main(String[] args) throws IOException {
        // Get set up to read lines of text from the user
        BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
        System.out.println("Taper le texte à encoder... quit pour quitter");
        for(;;){ // Boucle sans fin
            System.out.print("> "); // affiche le prompt
            String line = in.readLine(); // ligne une ligne
            if ((line == null) || line.equals("quit")) // Si EOF ou "quit" alors...
                break; // ... quitter la boucle
            edition(line);
        }
    }
    public static void edition(String line){
        ...
    }
}
```

Résultat à obtenir :

```
Taper le texte à encoder... quit pour quitter
> texte à traiter
texte encodé >|t|e|x|t|e| |à| |t|r|a|i|i|t|e|r
>
```

2) On souhaite à présent réaliser une édition codée. C'est à dire que chaque lettre de l'alphabet lu doit être codée et éditée par la procédure d'édition. On propose d'utiliser la méthode `Rot13` qui substitue à chaque lettre de l'alphabet latin (les autres, les blancs ou autres caractères de ponctuation ne sont pas modifiés) par la même lettre « tournée » de 13 places dans l'alphabet.

Par exemple le « a » est remplacé par « n », le « p » est remplacé par « c »...

Puisque l'alphabet Latin a 26 lettres, cette méthode encode et décode également! Vérifiez le lors de vos tests.

Ecrire cette méthode `Rot13(char c)` ; qui retourne le caractère modifié.

Résultat à obtenir :

```
Taper le texte à encoder... quit pour quitter
> texte à coder!
texte encodé >grkgr à pbqre! ← Texte encodé
> grkgr à pbqre!
texte encodé >texte à coder! ← Texte encodé une deuxième fois... donc décodé
> quit
```

Exercice 8 : Algorithme de tri par insertion

Durée 1H

Objectifs : Algorithmique de base sur les tris. Langage Java sur les tableaux.

Créer une classe `Trieur` qui contient une méthode `trie()` avec en argument un tableau d'entiers. Proposer un tri par insertion ; ce tri consiste à récupérer à chaque itération la plus petite valeur à partir d'un indice `i`, puis à réaliser une insertion par échange de valeurs en `i` pour mettre la plus petite valeur (`i` étant incrémentée après chaque itération).

Exemple :

Tableau à trier : 3 1 2

Itération 1: 132, c'est la valeur 1 qui est la plus petite, on l'échange avec 3.

Itération 2 : 123, c'est la valeur 2 qui est la plus petite à partir de la position 2, on l'échange avec 3.