

UEF 7.3. Patrons De Conception

TD/TP 5

Modèle - Vue - Contrôleur

Exercice 1 :

Dans le cadre de la gestion du cadastre d'une ville, les géographes ont besoin de découper chaque zone en sous-zones et en parcelles. Pour simplifier le problème, on partira du postulat que chaque parcelle est un rectangle et que les rectangles se suivent en un pavement parfait. Dans chaque zone, on stockera ses sous-zones et/ou ses parcelles. Dans chaque parcelle, on aura la position de son coin haut gauche, sa longueur, sa largeur et sa surface. Les zones peuvent également donner leur surface. Les géographes veulent avoir deux types d'affichage des informations sur les parcelles. Le premier affichage est graphique : une représentation des parcelles est faite avec une certaine échelle. Le second type d'affichage est textuel : pour chaque parcelle, les coordonnées, dimensions et surface sont indiquées.

1. Construire le modèle UML complet correspondant à cette gestion du cadastre en se basant sur MVC et les patrons de conception appropriés.

Note: Exercice inspiré de l'examen Master 2, Ingénierie des Logiciels, USTHB 2010/2011

Exercice 2 : Soit le code suivant

```
import java.util.*;

public class main {

    // Un annuaire en format liste [] des dictionnaires {}
    private static Personne [] annuaire = {
        new Personne("Ahmed", "Mahdi", "077878"),
        new Personne("Mohamed", "Rabehi", "06678"),
        new Personne("Mounir", "Mahdi", "0556644"),
        new Personne("Noui", "Brahimi", "067879")};

    // Ce programme permet de rechercher le num d'une personne dans l'annuaire
    // par son nom et l'afficher, lire des données a partir du clavier
    public static void main (String [] arg) {
        System.out.println("Recherche d'un telephone");
        System.out.println("Introduire Un nom");
        Scanner s= new Scanner(System.in);
        String nom = s.nextLine();
        // nombre d'elements trouvés
        int nb_found = 0;
        // parcours des personnes
        for (Personne p: annuaire){
            //afficher toutes les personnes qui ont le nom donné
            if (p.getNom().equals(nom)){
                System.out.println(p.getNom()+" " + p.getPrenom()+" " + p.getTel());
                nb_found += 1;
            }
        }
    }
}
```

UEF 7.3. Patrons De Conception

```
    if (nb_found==0)
        System.out.println( "ce nom "+nom+" n'existe pas ");
    }
}

public class Personne {
    private String prenom, nom, tel;
    public Personne (String nom, String prenom, String tel){
        this.nom=nom;
        this.prenom=prenom;
        this.tel=tel;
    }
    public String getPrenom() { return prenom; }
    public String getNom() { return nom; }
    public String getTel() { return tel; }
}
```

1. Nous souhaitons reformuler le code précédant en séparant l'aspect affichage de l'aspect données. Donnez la nouvelle conception et le code correspondant.
2. Quels sont les avantages et les inconvénients de cette restructuration ?

Note : Exercice inspiré du TD de Taha Zerrouki, Master 1, Université de Bouira.