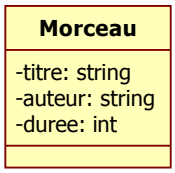
Travaux pratiques
séance n°7 - corrigé

Classes

Exercice 1

Question 1 : on souhaite représenter un ensemble de morceaux musicaux. A partir du diagramme de classe qui suit, définir le schéma de classe java correspondant :



Question 2 : la compléter avec les " getters" et les " setters" (accesseurs aux variables d'instance).

Question 3 : ajouter un constructeur permettant d'initialiser toutes les variables d'instance de cette classe

Question 4 : on introduit maintenant 4 constantes publiques pour représenter 4 genres musicaux (JAZZ, REGGAE, METAL, SKA). Ajouter les " getters" et les " setters" pour la nouvelle variable d'instance `genre`.

Solution

```
public class Morceau {
    public static final int REGGAE=1,JAZZ=2,SKA=3,METAL=4;
    private String titre;
    private String auteur;
    private int duree;
    private int genre;

    public Morceau(String titre,String auteur,int duree, int genre){
        this.titre=titre;
        this.auteur=auteur;
        this.duree=duree;
        this.genre=genre;
    }

    public String getTitre(){
        return titre;
    }

    public String getAuteur(){
        return auteur;
    }

    public int getDuree(){
        return duree;
    }
```

```
    public int getGenre(){
        return genre;
    }

    public void setTitre(String titre){
        this.titre = titre;
    }

    public void setAuteur(String auteur){
        this.auteur = auteur;
    }

    public void setDuree(int duree){
        this.duree = duree;
    }

    public void setGenre(int genre){
        this.genre = genre;
    }
}
```

Exercice 2

Question 1 : une " playlist" rassemble une suite de morceaux sélectionnés (voir diagramme UML). La classe java qui les représente stocke les morceaux sous la forme d'une instance de la classe `java.util.ArrayList` (consulter la documentation en ligne).
On peut ajouter un nouveau morceau, supprimer un morceau.

Question 2 : ajouter une variable de classe pour comptabiliser à tout moment le nombre d'instances de " playlists" existantes.

Solution

```
import java.util.ArrayList;

public class PlayList {
    private ArrayList<Morceau> liste;
    private static int nbListes = 0;
    private int taille = 0;

    public PlayList(){
        liste = new ArrayList<Morceau>();
        nbListes++;
    }

    public ArrayList<Morceau> getListe() {
        return liste;
    }

    public int getTaille(){
        return taille;
```

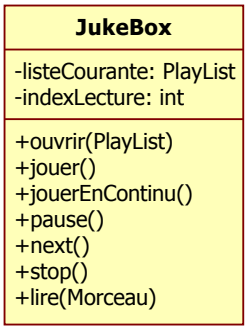
```
    }

    public void ajouter(Morceau morceau) {
        liste.add(morceau);
        taille++;
    }

    public void supprimer(Morceau morceau) {
       liste.remove(morceau);
       taille--;
    }

    public static int getNbListes() {
        return nbListes;
    }
}
```

Exercice 3

Question 1 : Un ″ juke box″ est associé à une ″ playlist″ . Il permet l'écoute de tout ou partie d'une ″ playlist″ .
La variable d'instance `indexLecture` représente l'index dans la ″ playlist ″ du morceau en lecture.
Les opérations disponibles sont :

- `jouer` un morceau. Notons que lorsque la fin de la liste est atteinte, c'est le premier morceau qui est joué.
- `jouer` les morceaux en continu.
- `lire` un morceau consiste en l'affichage du titre du morceau courant,
- `next` joue le morceau suivant (circulairement)
- `pause` affiche le message : ″ `---pause---`″ .
- `stop remet l'index de lecture à 0;`

Son modèle est représenté par le diagramme UML :



Implanter la classe `JukeBox`.

Solution

```
public class JukeBox {
    private PlayList listeCourante;
    private int indexLecture = 0;

    public void jouer() {
        if(indexLecture+1>listeCourante.getTaille())
            indexLecture = 0;
        lire((listeCourante.getListe()).get(indexLecture));
```

```
    }

    public void jouerEnContinu() {
        indexLecture = 0;
        Morceau morceau = null;
        Iterator<Morceau> iter = (listeCourante.getListe()).iterator();
        while( iter.hasNext() ){
            morceau = iter.next();
            lire(morceau);
            indexLecture++;
        }
    }

    public void lire(Morceau morceau){
        out.println("morceau en cours ....  "+morceau.getTitre());
    }

    public void pause() {
        out.println("---pause---");
    }

    public void next() {
        if(indexLecture+1>listeCourante.getTaille())
            indexLecture=0;
        else
            indexLecture++;
        jouer();
    }

    public void stop() {
        indexLecture = 0;
    }

    public void ouvrir(PlayList playList) {
        listeCourante = playList;
    }
}
```

Question 2 : créer le package `musik` contenant ces 3 classes.

Solution

ajouter la clause : **`package`** `musik;` au début de chaque fichier source.

Exercice 4

Il s'agit maintenant d'écrire un programme (classe contenant la méthode`main`) dont l'algorithme est le suivant :

1. importer le package `musik`
2. construire 4 instances de Morceau
3. construire un ″ playlist″
4. ajouter les 4 morceaux à la ″ playlist″
5. construire un ″ jukebox″ .
6. ouvrir la ″ playlist″
7. jouer la ″ playlist″ en continu
8. appuyer sur pause
9. jouer le morceau courant
10. jouer le morceau suivant

11. jouer le morceau courant
12. supprimer un morceau
13. jouer la ″ playlist″ en continu
14. stop
15. construire une nouvelle ″ playlist″
16. y ajouter 2 morceaux
17. ouvrir la ″ playlist″
18. jouer le morceau courant
19. passer au suivant, puis au suivant et encore au suivant.
20. afficher le nombre d'instances de ″ playlists″

Solution

```
import musik.*;
public class Main{
   public static void main( String[] args){
        Morceau morceau1 = new Morceau("aa","aa",12,Morceau.JAZZ);
        Morceau morceau2 = new Morceau("bb","bb",15,Morceau.SKA);
        Morceau morceau3 = new Morceau("cc","cc",16,Morceau.REGGAE);
        Morceau morceau4 = new Morceau("dd","dd",22, Morceau.JAZZ);

        PlayList playListe = new PlayList();
        playListe.ajouter(morceau1);
        playListe.ajouter(morceau2);
        playListe.ajouter(morceau3);
        playListe.ajouter(morceau4);

        JukeBox jukeBox = new JukeBox();
        jukeBox.ouvrir(playListe);
        jukeBox.jouerEnContinu();
        jukeBox.pause();
        jukeBox.jouer();
        jukeBox.next();
        jukeBox.jouer();
        playListe.supprimer(morceau2);
        jukeBox.jouerEnContinu();
        jukeBox.stop();

        System.out.println("\n\n");
        PlayList playListe2 = new PlayList();
        playListe2.ajouter(morceau1);
        playListe2.ajouter(morceau2);
        jukeBox.ouvrir(playListe2);
        jukeBox.jouer();
        jukeBox.next();
        jukeBox.next();
        jukeBox.next();

         System.out.println(
             "\n\n- nombre d'instances : "+PlayList.getNbListes());

    }
}
```