

TP 1 : Premiers pas

1 Hello World

Recopier, compiler (javac HelloWorldApp.java) et exécuter (java HelloWorldApp) le code suivant. On se rappellera qu'il faut donner comme nom au fichier le même nom que la classe principale décrite dans le fichier, avec l'extension java. Ici par exemple, ce sera HelloWorldApp.java :

```
/*
 * The HelloWorldApp class implements an application that
 * simply displays "Hello World!" to the standard output.
 */
class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!"); //Display the string.
    }
}
```

1. Comprendre ce programme.
2. Changer la classe `HelloWorldApp` pour qu'elle affiche votre prénom.
3. Définir une classe `Echo`, avec une méthode `main` qui affiche les arguments de la ligne de commande (contenus dans `String[]`)
4. Définir une classe `Somme`, avec une méthode `main`, qui affiche la somme des arguments (entiers) de la ligne de commande. On utilisera la procédure `Integer.parseInt(s)` qui transforme une chaîne de caractères `s` en un entier si cela est possible.

2 Objet comment ça marche

Fichier `CD.java` :

```
class CD{
    /* Attribut de class*/
    char sillon[];
    /* Constructeur*/
    public CD(char sillon[]){
        this.sillon=sillon;
    }
    private CD(){
    }
    /*accesseur*/
    public char lire(int t){
        if (sillon.length>t)
            return sillon[t];
        else
            return (char) 0;
    }
}
```

Fichier CDR.java :

```
class CDR extends CD{
    public CDR(int t){
        super(new char[t]);
    }
    public void ecrire(char data, int pos){
        if (sillon[pos]==0)
            sillon[pos]=data;
    }
}
```

Fichier LecteurGraveur.java

```
class LecteurGraveur{
    CD cd;
    public void inserer(CD cd){
        this.cd=cd;
    }
    public void ejecter(){
        cd=null;
    }
    public String lire(){
        String ret=null;
        if (cd==null){
            System.out.println("Veuillez insérer le CD S.V.P");
        }else{
            ret="";
            for(int a=0;cd.lire(a)!=0;a++)
                ret+=Character.toString(cd.lire(a));
        }
        return ret;
    }
    public void ajouter(String text){
        //on vérifie le type de CD
        if (cd instanceof CDR){
            int begin=0;
            CDR cdr=(CDR)cd;
            // on cherche le première case vide
            for(;cd.lire(begin)!=0;begin++);
            // on écrit à partir de cette case
            for(int a=0;a<text.length();a++){
                cdr.ecrire(text.charAt(a),a+begin);
            }
        }
    }
    public void Effacer(){
        /*A Compléter*/
    }
}
```

Fichier Humain.java :

```
class Humain{
    public static void main(String[] arg){
```

```

String text;
char data[]={'B','o','n','j','o','u','r'};

CD cd1=new CD(data); //On crée un CD avec bonjour écrit dessus.
CDR cd2=new CDR(75); //On crée un CDR de 75 caractères.
// on crée le graveur

LecteurGraveur LG=new LecteurGraveur();
//on insère le CD

LG.inserer(cd1);

//on le lit

text=LG.lire();
System.out.println(text);

//on insère le CDR (c'est magique pas besoin d'éjecter)

LG.inserer(cd2);
// on écrit dessus

LG.ajouter(text);
LG.ajouter(" tout le monde");
System.out.println(LG.lire());
}
}

```

Un CD est un support numérique que l'on presse directement avec les données. On ne peut que les lire. La classe CD est faite de cette façon, seul le constructeur peut y mettre des données. Le CDR est un CD où on peut ajouter des données, mais pas les supprimer. Le CDRW est un CDR où on peut tout effacer. Un Lecteur graveur peut lire, écrire et effacer le disque inséré s'il est du bon format.

1. Comprendre et compiler les classes ci-dessus.
2. Faire la classe CDRW qui hérite de CDR et qui permet l'effacement.
3. Compléter la classe LecteurGraveur.
4. Nous remarquons que le LecteurGraveur ne vérifie pas si le CD est plein lors de l'écriture, corriger cela.

3 Distributeur de boisson

Le but de cet exercice est de vous familiariser avec les objets.

3.1 La version simple.

Pour commencer, nous allons faire la classe Distributeur. Pour le moment le distributeur possède un bouton servant à obtenir un produit. Le produit est aussi un objet. La classe Main va jouer le rôle d'interface entre vous et l'objet distributeur. elle va aussi brancher le distributeur en instanciant la classe.

1. Voici le code de la structure comprenez le et complétez le au fur et à mesure :

Fichier *Main.java* :

```
class Main{
    public static void main(String args[]){
        Produit produit;
        System.out.println("Branchons le Distributeur");
        Distributeur distributeur=new Distributeur();
        System.out.println("Demande d'un produit");
        produit=distributeur.getProduit();
        if (produit!=null)
            System.out.println(produit+" obtenu");
    }
}
```

Fichier *Distributeur.java* :

```
class Distributeur{
    public Produit getProduit(){
        return new Produit();
    }
}
```

Fichier *Produit.java* :

```
class Produit{
    private String nom="produit";
    Produit(){
    }
    /* Constructeur donnant un nom au produit*/
    Produit(String nom){
        this.nom=nom;
    }
    public String toString(){
        return nom;
    }
}
```

2. Nous remarquons que le distributeur génère un produit à la demande. Ajouter les fonctions `boolean loadProduit(Produit produit)` qui prend un produit et le met dans le distributeur et renvoie "true" si elle a été insérée et "false" si il n'y a plus de place.

Vous allez donc ajouter dans le distributeur un tableau du nombre de places. `getProduit()` renverra "null" lorsqu'il n'y aura plus de produit (Utilisez des méthodes efficaces). C'est la fonction `main` de la classe `Main` qui va charger les produits dans le distributeur.

Références

Pour vous aider, vous pourrez utiliser les deux références ci-dessous. La première contient un tutorial sur le langage et la seconde contient les spécifications de l'API Java (toutes les procédures et les classes déjà définies).

<http://java.sun.com/docs/books/tutorial/>

<http://java.sun.com/j2se/1.5.0/docs/api/>