

Java DUT 1 Feuille TD9
Université Paris-Est Marne-la-Vallée

Comment faire un menu en Java ? En Java, on peut utiliser les Map pour écrire des menus. Cette solution permet d'ajouter des options dans les menus plus facilement qu'avec un bloc `switch`.

Exercice 1.—

Les classes utilisées typiquement seront les suivantes :

- a) Une classe `Main` avec une méthode `main` qui lance une méthode `start` de la classe `TextMenu` sur un `TextMenu`.

```
public class Main {
    public static void main(String[] args){
        new TextMenu().start();
    }
}
```

- b) Une classe `Menu` qui contient une Map qui associe à chaque String un objet d'une classe implémentant l'interface `Action` contenant une méthode `doIt`. La classe `Menu` est :

```
public class Menu {
    private final Map<String,Action> map;

    public Menu() {
        map = new HashMap<String,Action>();
        map.put("a", new AAction());
        map.put("b", new BAction());
        map.put("q", new QuitAction());
    }

    public Map<String,Action> getMap(){
        return map;
    }
}
```

- b) Une classe `TextMenu` qui dérive de `Menu` et comprend un champ statique `text` de type `String` correspondant au texte du Menu et une méthode `start` qui affiche le texte, attend le choix de l'utilisateur (une `String`) et effectue l'action correspondante.

```
public class TextMenu extends Menu{
    private static String text =
        "Entrez votre choix \n" +
        "a : choix a\n" +
        "b : choix b\n" +
        "q : quitter\n";
}
```

```

public TextMenu(){
    super();
}

public void start() {
    Scanner sc = new Scanner(System.in);
    System.out.print(text);
    boolean notQuit = true;
    String choice;
    while (notQuit) {
        if (sc.hasNext()) {
            choice = sc.next();
            if (getMap().containsKey(choice)){
                getMap().get(choice).doIt();
                notQuit = !(choice.equals("q"));
            } else {
                System.out.println("Choix non valide");    }
        }
    }
    sc.close();
}
}

```

c) Une interface Action avec une méthode doIt qui effectuera l'action voulue.

```

public interface Action {
    void doIt();
}

```

d) Des classes AAction, BAction..., et QuitAction qui implémentent Action et correspondent au choix de la chaîne "a", "b"..., "q", dans le menu. implémentant

```

public class AAction implements Action{
    public void doIt() {
        System.out.println("Action correspondant au choix a");
    }
}

```

```

public class QuitAction implements Action{
    public void doIt() {
        System.out.println("On sort");
    }
}

```

Faites un test de ces classes.

Exercice 2.—

On reprend la feuille TD8 qui permet de gérer une Company contenant des Employee. L'interface Company est

```
interface Company {
    // add the Employee e to the Company
    boolean add(Employee e);
    // remove the Employee e to the Compagny
    Employee remove(Employee e);
    // returns the number of Employee of the compagny
    int size();
}
```

a) Créer un menu pour manipuler une Company avec les choix suivants :

```
public class TextMenu {
    private static String text =
        "Entrez votre choix \n" +
        "create : creer la Company \n" +
        "add firstName lastName : ajouter un Employee \n" +
        "print : afficher la liste des Employee \n"+
        "q : quitter \n";
    // ...
}
```

b) Que doit-on changer pour ajouter un nouveau choix d'action sur le menu ?