

Java – TP1

Le jeu des allumettes

Nicolas Baudru
Carine Guivier-Curien
Laurent Vallet

Année 2008-2009

Avant propos : Chaque classe écrite doit être enregistrée dans un fichier du même nom. Par exemple une classe *Chien* doit être enregistrée dans un fichier *Chien.java*. Chaque classe écrite doit immédiatement être testée. Par exemple vous créez une classe *TestChien* qui contiendra uniquement une méthode *main* à partir de laquelle vous testerez toutes les méthodes de votre classe *Chien*.

Compilation et exécution : Nous vous rappelons que pour utiliser un programme, il faut d’abord le compiler. Pour cela, on utilise la commande *javac* suivie du nom de votre fichier. Exemple : *javac Chien.java* et *javac TestChien.java*. Une fois que le compilateur ne trouve plus d’erreur, c’est-à-dire que la compilation réussie, vous obtenez un fichier avec une extension *.class*. (*Chien.class* et *TestChien.class* dans notre exemples). Si votre classe contenait une méthode *main*, vous pouvez alors l’exécuter en utilisant la commande *java*. Dans notre exemple seule *TestChien.class* est exécutable par la commande *java TestChien*.

Le jeu des allumettes : Nous vous proposons de programmer le jeu des allumettes. Les règles de ce jeu sont très simples : un joueur (vous) et l’ordinateur doivent prendre à tour de rôle 1, 2 ou 3 allumettes. Celui qui prend la dernière a gagné.

La conception de ce programme est minimaliste. En effet le programme est constitué de 4 classes, *Jeu*, *Joueur*, *Ordinateur* et *TestJeu*. La classe *TestJeu* sert à tester votre programme. Elle vous est déjà fournie :

```
class TestJeu {
    public static void main (String [] args ){
        Jeu a = new Jeu();
        a.menuJeu();
    }
}
```

Question 1 : Nous allons commencer par écrire la classe *Ordinateur*. Cette classe permet de créer un objet qui jouera le rôle de l'ordinateur. L'ordinateur doit simplement proposer un nombre d'allumettes à retirer. Ce nombre doit être compris entre un et trois. Créez la classe *Ordinateur* contenant une seule méthode appelée *propositionOrdinateur*. Cette méthode ne prend aucun argument et retourne un entier. Son comportement est le suivant :

- elle choisit aléatoirement un nombre entier i compris entre 1 et 3;
- elle affiche à l'écran ce nombre dans un message du type : « l'ordinateur a choisi le nombre i » ;
- enfin elle retourne le nombre i .

Le squelette de cette classe est donné ci-dessous. Complétez-la, compilez-la puis testez-la.

```
class Ordinateur {  
    public int propositionOrdinateur() { ... }  
}
```

Question 2 : La seconde classe que nous allons écrire est celle du *Joueur*. La méthode *propositionJoueur* de cette classe vous permet d'entrer au clavier un nombre entier compris entre 1 et 3. Ce nombre correspondra au nombre d'allumettes que vous souhaitez enlever. Son comportement est le suivant :

- la méthode vous demande d'abord de taper au clavier un nombre entier compris entre 1 et 3;
- elle récupère ensuite le nombre que vous avez saisi. Vous pouvez utiliser pour cela la méthode *lireChaine* de la classe *LectureChaine* vue en cours avec le constructeur de la classe *Integer* (rechercher sur internet comment s'utilise cette classe) ;
- enfin elle retourne le nombre choisi.

Le squelette de cette classe est donné ci-dessous. Complétez-la, compilez-la puis testez-la.

```
class Joueur {  
    public int propositionJoueur() { ... }  
}
```

Question 3 : La méthode *propositionJoueur* que vous venez d'écrire a un défaut. Elle ne vérifie pas si l'utilisateur saisit bien un nombre compris entre 1 et 3. Modifiez la méthode *propositionJoueur* de sorte qu'elle redemande au joueur de saisir un nombre tant que ce dernier n'est pas compris entre 1 et 3.

Nous allons maintenant écrire la dernière classe. La classe *Jeu* contient un attribut *nbAllumettes* de type entier représentant le nombre d'allumettes restant (initialisé à 17), et trois méthodes, *afficherAllumettes*, *commencerJeu* et *menuJeu*. Ces trois méthodes sont publiques (mot clé *public*, ne retournent rien (mot clé *void*) et ne prennent aucun argument (il n'y a rien entre les parenthèses).

```

class Jeu {
    int nbAllumettes = 17;

    public void afficherAllumettes() { ... }
    public void commencerJeu(){ ... }
    public void menuJeu() { ... }
}

```

Question 4 : Intéressons-nous d’abord à la méthode *afficherAllumettes*. Celle-ci a un comportement très simple : elle affiche à l’écran autant de symbole « | » (appelé pipe) que d’allumettes restantes. En d’autres termes, un pipe représentera une allumette. Ecrivez cette méthode, puis compilez la classe *Jeu* et testez-la.

La méthode *commencerJeu* est plus compliquée à mettre en oeuvre. Nous allons l’aborder en deux temps.

Question 5 : Dans un premier temps, vous devez concevoir cette méthode pour qu’elle se comporte comme suit :

- elle crée un objet de type joueur ;
- tant qu’il reste des allumettes :
 - elle affiche le nombre d’allumettes restantes sous forme de pipe,
 - puis elle demande au joueur de proposer un nombre *nb* (c’est ici qu’il faut utiliser la méthode *propositionJoueur*),
 - et enfin elle retire *nb* allumettes du jeu,

Ecrivez cette méthode, puis compilez la classe *Jeu* et testez-la.

Question 6 : Modifiez la méthode précédente de sorte que le joueur et l’ordinateur jouent chacun leur tour. Compilez de nouveau la classe *Jeu* et testez-la.

Question 7 : Modifiez la méthode précédente de sorte que le vainqueur du jeu (c’est à dire celui qui prend la dernière allumette) s’affiche à l’écran. Compilez de nouveau la classe *Jeu* et testez-la.

Question 8 : Il nous reste en fin la méthode *menuJeu*. Nous vous laissons le choix de son comportement. L’idée étant que cette méthode vous permettent de configurer certains aspects du jeu, comme par exemple :

- si oui ou non vous voulez lancer une autre partie,
- le nombre d’allumettes en début de partie,
- le nombre de joueurs,
- etc

Vous pouvez rajouter des attributs, méthodes ou même des classes si vous en avez l’utilité. N’oubliez pas de (re)compilez toutes les classes modifiées ou nouvellement créée et de les testez souvent. Cela vous évitera bien des désagréments.