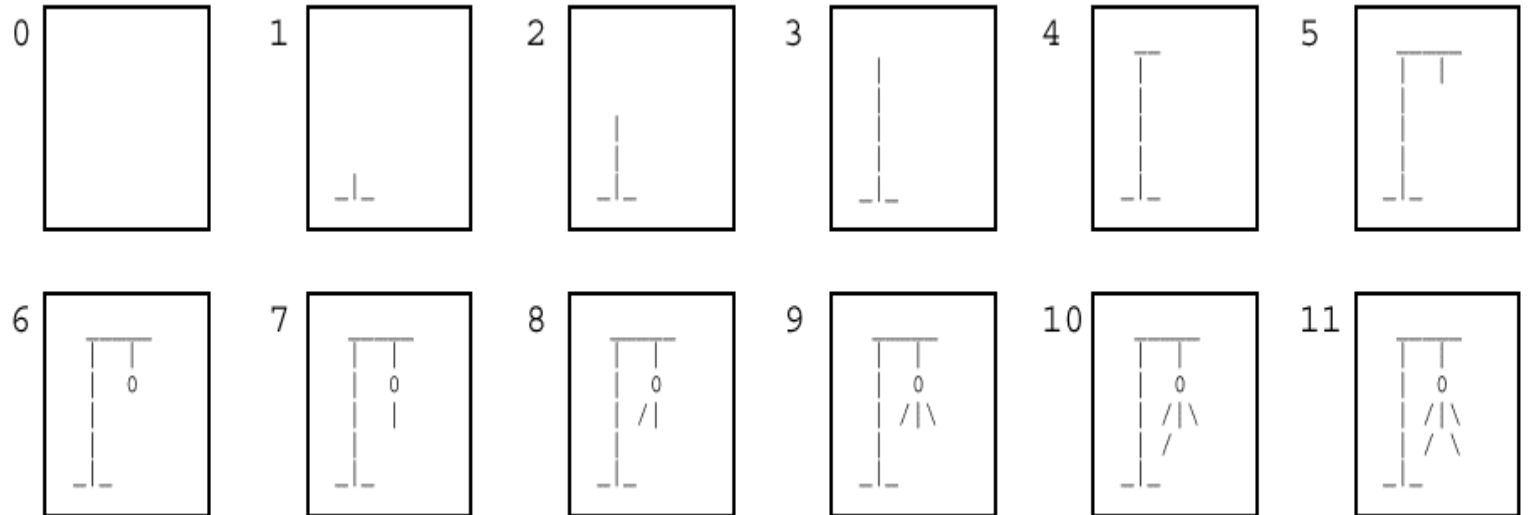


# TP 4: Le Pendu

## 1 Dessiner le Pendu

Voici l'avancement du dessin du pendu en fonction du nombre d'erreurs du joueur :



Voici le tableau de caractères `tabPendu` représentant le dessin complet du pendu :

`tabPendu`

`tabLimite` (exercice 1)

	0	1	2	3	4	5
0		-	-	-	-	-
1						
2					o	
3				/		\
4				/		\
5	-		-			

	0	1	2	3	4	5
0						
1						
2						
3						
4						
5						

NB: - Le caractère \ s'écrit : '\\'  
 - Les cases vides contiennent le caractère espace (' ').

**Exercice 1:** Etablir les valeurs du tableau d'entiers `tabLimite`, de dimension 6x6, vérifiant :  
`tabLimite(i,j) = n` ssi le caractère `tabPendu(i,j)` apparaît pour la première fois au nombre d'erreurs n (par exemple on a: `tabLimite(3,4) = 7` , c'est le corps du pendu).

**Exercice 2:** Ecrire la classe `Pendu` , qui doit contenir :

- les constantes (déclarées 'final static') : **tabPendu** et **tabLimite**. Ces deux tableaux doivent naturellement être initialisés au moment de leurs déclarations, on utilisera pour cela les valeurs données dans l'énoncé pour tabPendu et les valeurs établies à l'exercice 1 pour tabLimite.
- la variable d'instance : **nbErreur**. Cette variable contiendra le nombre d'erreurs commis par le joueur.
- les méthodes : **incrementErreur()** et **aPerdu()**. La méthode incrementErreur() permettra d'incrémenter le nombre d'erreurs et la méthode aPerdu() doit retourner la valeur true si le joueur a perdu.
- un constructeur : sans argument, qui initialise simplement nbErreur à 0.

**Exercice 3:** Rajouter à la classe Pendu une méthode **dessinePendu()** qui affiche à l'écran le dessin complet du pendu.

**Exercice 4:** Rajouter à la classe Pendu une méthode de signature:

**public char caractere(int i, int j)**, qui retourne le caractère à afficher à la position (i,j) en fonction du nombre d'erreurs courant.

**Exercice 5:** Rajouter à la classe Pendu une méthode **dessine()** qui affiche à l'écran le dessin du pendu correspondant à la valeur du nombre d'erreurs courant.

**Exercice 6:** Rajouter une fonction **main()** qui utilise une instance de la classe Pendu, et qui affiche successivement toutes les étapes du dessin du pendu.

## 2 Le jeu du Pendu

**Exercice 7:** Modifier la classe Pendu de la façon suivante :

- ajouter les attributs : **'String solution'** et **'boolean[] trouve'**.
- adapter le constructeur : en ajoutant un paramètre de type String qui servira à initialiser la chaîne solution, et qui initialise le tableau 'trouve' à la dimension solution.length() et le remplit avec la valeur false.
- ajouter les méthodes : **afficheSolution()** et **affiche()**. La méthode afficheSolution() doit afficher à l'écran la chaîne solution avec un espace entre chaque caractère, affiche() fait la même chose mais si trouve[i]==false, le caractère '\_' sera affiché à la place du caractère se trouvant à la position i dans la solution.

**Exercice 8:** Rajouter la méthode : **public boolean choix(char c)**, qui retourne:

- 'true' si c apparaît dans la chaîne solution, et dans ce cas affecte la valeur 'true' à trouve[i] pour toutes les positions i de c dans la solution.
- 'false' si c n'apparaît pas dans la solution.

NB: La classe String possède la méthode : **public int indexOf(char c, int deb)**,

cette méthode retourne la position de la prochaine occurrence du caractère c dans la chaîne à partir de la position 'deb' (inclusive). Si c n'apparaît plus, la valeur -1 est retournée.

**Exercice 9:** Rajouter la méthode : **public boolean aGagner()**, qui retourne: 'true' si le joueur a gagné (si toutes les lettres de la solution ont été trouvées).

**Exercice 10:** Modifier la fonction **main()** de sorte à réaliser le jeu du pendu.

Voici l'affiche à fournir à l'utilisateur :

```

> java Pendu
Le jeu commence...

-----
Votre choix : J
*
  _|_
-----
Votre choix : E
*
  _|_
----- E _
Votre choix : F

```

```

*
  _|_
----- E _
Votre choix : A
*
  _|_
----- A _ E _
Votre choix : I
*

```

[...]

[...en cas de défaite]

```

*
  _|_
  |O
  / \
  / \
  _|_
----- A H I E R
Votre choix : K
*
  _|_
  |O
  / \
  / \
  _|_
-----
Perdu !
Solution: C A H I E R

```

[...en cas de victoire]

```

*
  _|_
  |O
  _|_
----- A H I E R
Votre choix : C
*
  _|_
  |O
  _|_
----- C A H I E R
Vous avez gagné !!!

```

**Infos :**  
La classe String possède la méthode : String toUpperCase(), qui retourne la valeur de la chaîne en majuscule.