

POO & Java

L3 - TP1

Exercice 1

Placez le code suivant dans un fichier `Personne.java`, compilez et exécutez le.

```
1 public class Personne{
2     String nom;
3     int age;
4     Personne(String nom, int age){
5         this.nom = nom;
6         this.age = age;
7     }
8     String getNom(){
9         return this.nom;
10    }
11    int getAge(){
12        return this.age;
13    }
14    void setAge(int a){
15        this.age = a;
16    }
17 }
```

Pour qu'il se passe quelque chose, ajoutez une méthode principale qui crée 2 personnes, affiche leurs noms et âges, ajoute 1 à leurs âges, et réaffiche leurs âges.

Exercice 2

- Écrire une classe `Liste` qui représente une liste chaînée d'objets. Une liste chaînée contient une valeur (de type `Object` ici) et une "queue", qui est le reste de la liste. Cette classe contiendra au moins les méthodes suivantes :
 - créer une liste vide,
 - ajouter un objet à la fin de la liste,
 - ajouter un objet en tête de liste,
 - connaître la position d'un objet donné dans la liste,
 - récupérer l'objet situé à une position donnée.
- Dans une méthode principale, créer une liste de chaînes de caractères aléatoires, la parcourir et afficher les chaînes. Voici une fonction qui génère une chaîne de caractères alphabétiques minuscules aléatoires :

```
1 static String randomString(int length){
2     char[] t = new char[length];
3     for(int i=0; i<length; i++) t[i] = (char) ('a' + (int) (Math.random()*26));
4     return new String(t);
5 }
```

- Créer une liste de listes de chaînes et afficher les chaînes.
- Créer une classe `Crypteur` dont chaque instance peut crypter un texte selon le principe du code de César, ou décalage circulaire : à chaque lettre correspond la lettre située `n` places plus loin dans l'alphabet. Un objet de type `Crypteur` doit opérer sur deux listes de caractères, la première contient les lettres de `a` à `z`, la seconde les lettres décalées. Exemple de listes de cryptages pour un décalage de 3 :

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c

5. Dans une méthode principale, créer deux crypteurs avec des décalages différents et leur faire crypter la même chaîne de caractères.

Exercice 3

On veut simuler le comportement d'une foule. 3 classes sont déjà écrites :

- une classe `Displayable` dont les instances peuvent s'afficher sur une interface graphique,
- une classe `CrowdFrame` qui affiche des `Displayable` dans une fenêtre,
- une classe `CrowdGame`, **incomplète**, qui fait tourner la simulation

Toutes les personnes qu'on veut afficher, en plus d'être des `Displayable` doivent avoir une direction et une vitesse. De plus, les personnes s'affichent sous forme d'un rond complété par un petit trait qui indique la direction vers où la personne se déplace. Toutes les personnes ont également une méthode `void move()` qui déplace la personne en fonction de sa direction et de sa vitesse.

1. Sans rien changer aux classes `Displayable` et `CrowdFrame`, écrire une classe qui représente les personnes.
Il y a au moins deux types de personnes (vous pouvez en ajouter d'autres ...) :
 - Les *travailleurs* qui ont une couleur grise et marchent toujours dans la même direction à une vitesse plus élevée que les autres personnes.
 - Les *touristes* qui changent de direction aléatoirement à chaque appel de la méthode `move`, ou avec une certaine périodicité et sont de toutes les couleurs.
2. Écrire les classes représentant les travailleurs et les touristes
3. Compléter la classe `CrowdGame` pour générer des personnes des deux types et faire tourner la simulation (mettre la ligne 22 de `CrowdFrame.java` en commentaire pour afficher les positions successives).

