**TP : Les expressions lambda et streams**

***Exercice 1***

Écrire la méthode affichage\_selectif afin que le programme suivant affiche les éléments positifs du tableau tab :  
public class Affichage  
{ public static void main (String [] args)  
{ int [] tab = {1, 4, -2, 9, -3, 5, -3 } ;  
System.out.print ("-- Les positifs de tab : ") ;  
affichage\_selectif (tab, ee -> ee > 0) ;  
}  
}

***Exercice 2***

Compléter la dernière solution de l’exercice précédent, de manière que le programme affiche, en plus des nombres positifs du tableau tab, les nombres négatifs, puis les nombres pairs :  
-- Les positifs :1 \* 4 \* 9 \* 5 \* 12 \* 7 \* 6 \*-- Les negatifs :-2 \* -3 \* -3 \* -11 \*-- Les pairs :4 \* -2 \* 12 \* 0 \* 6 \*On proposera tout d’abord une solution utilisant des expressions lambda, puis une solution utilisant des références à des méthodes qu’on écrira.

***Exercice 3***

On dispose de la classe Point définie ainsi :

class Point{ public Point (int x, int y) { this.x = x ; this.y = y ; }public int getX() { return x ; }public int getY() { return y ; }public void affiche (){ System.out.print(" [ " + x + ", " + y + "] ") ; }private int x, y ;}Écrire une méthode statique nommée traiteListe recevant en premier argument une liste de points sur laquelle elle réalise successivement trois opérations paramétrées par les trois arguments suivants :  
• une sélection des éléments réalisant une condition ;  
• un tri (suivant un critère variable) des éléments sélectionnés ;  
• un affichage des éléments ainsi triés.  
La méthode traiteListe s’utilisera ainsi (le type des arguments étant à préciser) : traiteListe (liste, selection, tri, affichage) ;On utilisera cette méthode :  
• pour sélectionner les points d’abscisse positive, les trier sur les valeurs de leurs abscisses et les afficher suivant cette forme :  
[ 2, 5] [ 2, 3] [ 6, -3]• pour sélectionner tous les éléments, les trier suivant la somme de leurs coordonnées et les afficher ainsi :  
(abs = -3, ord = 4) (abs = 6, ord = -3) (abs = 2, ord = 3)Ici, on évitera d’utiliser des streams.

***Exercice 4***

On dispose de la classe Personne suivante :  
class Personne{ public Personne (String prenom, String nom, int annee){ this.nom = nom ; this.prenom = prenom ; annee\_naissance =annee ; }public String getNom() { return nom ; }public String getPrenom() { return prenom ; }public int getAnnee() { return annee\_naissance ; }private String nom, prenom ;private int annee\_naissance ;}}Réaliser les opérations suivantes sur un tableau d’objets de type Personne, en utilisant un stream :  
• afficher le nom des personnes nées après 1985 ;  
• afficher le nom des personnes nées avant 2000, triés par ordre alphabétique sur leur nom, et afficher leur nombre ;  
• afficher tous les noms et prénoms, triés par ordre alphabétique sur leur nom et leur prénom.

Écrire un programme qui, à partir d’une liste de personnes (List<Personne>) utilise un stream pour afficher l’année de naissance de la plus jeune. On proposera deux formulations :  
• l’une ne recourant pas à un comparateur et affichant seulement l’année concernée ;  
• l’autre affichant le nom, prénom et année de naissance de la personne concernée (on peut, cette fois, utiliser un comparateur)