Université Mohammed V – Agdal
Faculté des Sciences – Rabat
Département Informatique

Pr. S. ZITI
Drt. A. ALAOUI
2012/2013

Travaux pratiques 2
Programmation orientée objet (Java)
SMA5

Exercice 1:

Un monôme est une application $\mathbf{x \rightarrow ax^n, x \in R, a \in R, n \in N}$; a est appelé le coefficient du monôme et n son degré.

Le but de cet exercice est de modéliser un monôme sous la forme d'un objet défini par son degré et son coefficient.

1. a. Coder une classe **Monome** qui encapsulera le degré et le coefficient.

b.Coder également le(s) constructeur(s) nécessaire(s). Une exception **ArithmeticException** devra être levée si on essaye de construire un monôme avec un degré négatif.

c.Coder les méthodes **GetDegre** et **GetCoefficient** qui permettront une encapsulation efficace.

2. Coder une méthode appelée calcul permettant de calculer la valeur du monôme pour une valeur donnée en paramètre de x.

3. Surcharger les méthodes

- **String toString() :**permettant de représenter le monôme sous la forme de la chaîne $\mathbf{ax^n}$ (en remplaçant a et n par leurs valeurs)
- **boolean equals(Monome m)** permettant de comparer l'instance courante à un autre monôme (2 monômes sont égaux si leurs coefficients et leurs degrés sont identiques)
- Expliquer pourquoi il est utile de surcharger la méthode **equals**

4. Coder une méthode statique appelée **derivee** qui crée et retourne un nouveau monôme qui est la dérivée du monôme passé en paramètre.

Rappel : la dérivée de ax^n est anx^{n-1} si $n>0$; la dérivée de ax^0 est $0x^0$.

5. Coder la méthode **main** permettant de saisir au clavier le coefficient et le degré d'un monôme qui sera ensuite construit. Répéter la saisie d'un nombre x et l'affichage de la valeur du monôme pour ce paramètre x jusqu'à ce qu'on saisisse la valeur 0.

Exercice 2:

Le but de cet exercice à modéliser un polynôme comme une collection de Monomes

Reprendre la classe **Monome** de l'exercice précédent

1. Ecrire une classe **Polynome** qui sera composée d'une **ArrayList** de monomes
2. Ecrire la méthode **Monome getMonome(int i)** qui retourne le monôme à l'indice i de la liste
3. Ecrire la méthode **public int Ajoute (Monome m)** qui ajoute un monôme à la liste et qui retourne le nombre de monôme dans la liste. Si un monôme de même degré est déjà présent dans la liste, on le remplacera par un monôme de ce degré et dont le coefficient est la somme des coefficients du monôme ajouté et du monôme déjà présent, sauf si cette somme est nulle.
4. Ecrire une méthode **public String toString()** qui présente la liste sous la forme d'une chaîne. On veillera à obtenir une présentation élégante du style **P(x)=3x^2-2x+1** en évitant d'avoir plusieurs signes dans le cas d'un monôme négatif.
5. Ecrire une méthode **public double calcul(double x)** qui calcule la valeur du polynôme pour une valeur de x, en faisant la somme de toutes les valeurs des monômes de la liste pour cette valeur de x
6. Cette méthode n'est pas optimale, car on effectue de nombreuses fois le calcul de $\mathbf{x^n}$.
7. On cherche à mettre en œuvre le schéma de Horner qui écrit le polynome
   $P(x) = a_4 x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_o$
   sous la forme $P(x) = ((a_4 x + a_3) x + a_2) x + a_1) x + a_0$
8. Ecrire une méthode **public double calculHorner(double x)** qui effectue le calcul avec cette méthode.

   NB : il sera judicieux de trier préalablement la liste des monômes.