INFOJ

3

d'afficher les attributs d'un objet trinôme.

Exercice 8 : Méthodes de classe

Dans cet exercice, vous allez définir une « bibliothèque » de méthodes utilitaires pour la manipulation de nombres complexes. Pour cela, vous allez définir une classe Complexe ayant les attributs pr, pi, module (pr et pi désignant la partie réélle et la partie imaginaire d'un nombre complexe). Cette classe contiendra un constructeur et une méthode d'affichage. Vous définirez également une classe OperationComplexe contenant des méthodes utilitaires pour la manipulation de complexes. Les méthodes devant être implémentées sont : calcul du module. addition, multiplication...

Exercice 9: Attributs de classe

Implémentez une classe Personne qui aura deux attributs permettant de stocker le nom, le prénom d'une personne et son année de naissance. Vous ferez une méthode d'affichage qui affichera ces informations avec également le numéro de création de la personne. Vous ferez également une méthode permettant de calculer l'âge de chaque personne ainsi créée en fonction de la date de l'ordinateur. Pour cela vous utiliserez la classe Calendar (voir premier support de cours, transparent 29). Votre fonction d'affichage des informations associées à une personne sera alors sensible au fait que le champ age soit calculé ou nom.

Exercice 10:

Implémentez une classe Vecteur qui modélisera un objet vecteur de \mathbb{N}^3 . Votre classe vecteur aura une méthode permettant de construire un vecteur selon des valeurs données dans le code, une méthode d'affichage, une méthode permettant de calculer la norme d'un vecteur (la norme étant un attribut de votre classe), une méthode calculant la somme de deux vecteurs passés en paramètre et enfin une méthode réalisant le produit scalaire.

Exercice 11 : Les tableaux

En vous aidant de la classe Math (voir API), écrire une méthode permettant de calculer un nombre entier aléatoire compris entre deux valeurs. Vous écrirez ensuite une méthode qui retournera un tableau d'entiers pris au hasard entre deux valeurs et dont les arguments seront la taille du tableau et les deux valeurs de l'intervalle. Enfin, écrivez une méthode permettant l'affichage de ce tableau.

Exercice 12 : Les Cent Mille Milliards de Poèmes de Raymond Queneau

En 1961, Raymond Queneau (poète et co-fondateur de l'Oulipo) a publié un recueil de poèmes capable d'engendrer plus de 100 000 000 000 000 de poèmes! Chacun de ces poèmes a la forme d'un sonnet régulier (*i.e.* deux quatrains et deux tercets, soit au total quatorze vers). L'objectif de cet exercice est de réaliser un générateur de poèmes de Raymond Queneau. Pour cela, vous

allez télécharger le fichier Queneau, java à l'adresse suivante :

http://www-soc.lip6.fr/~marchett/Queneau.java (cliquez dessus)

A l'aide de vos connaissances sur les tableaux, implémentez ce générateur de poèmes en Java.

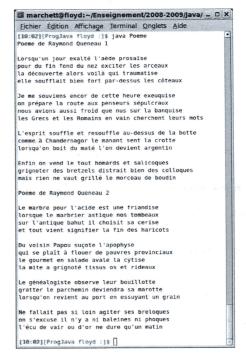


FIGURE 1 – Deux poèmes de Raymond Queneau.

TD1

Olivier Marchetti

Septembre 2023

Exercice 1:

Ecrire en Java un algorithme qui calcule e^x pour une valeur rentrée dans le programme.

Exercice 2: Nombres Parfaits

Un nombre est parfait s'îl est égal à la somme de ces diviseurs stricts (par exemple 6 = 1+2+3). Ecrire en Java un programme qui permet d'établir la liste des nombres parfaits inférieurs à 100000. Votre programme affichera la décomposition de chacun des nombres parfaits identifiés.

Exercice 3: Lignes de commandes

Ecrire un programme qui retourne les arguments présents sur la ligne de commande lors de l'invocation du programme.

Exercice 4 (D. Salvetti): Types

Les programmes suivants provoquent des erreurs lors de la compilation ou lors de l'exécution (plus exactement une exception). Expliquez pourquoi.

```
public class DivZero {
   public static void main(String args[]) {
      int i = 1, j = 0;
      double x = 1, y = 0;
      System.out.println(x / y);
      System.out.println(i / j);
   }
}
• public class Initialisation {
   public static void main(String args[]) {
      int i; final int j;
      System.out.println(i);
      j = 2;
      System.out.println(j);
      j = 3;
   }
}
```

Polytech-Sorbonne EI2I-4

```
• public class CastImplicite {
    public static void main(String args[]) {
        int i; double d; float f = 1f;
        i = 1.;
        f = 1.0;
        d = f;
    }
}
• public class Promotion {
    public static void main(String args[]) {
        for (byte i = 1; i < 10; i = i + 1) {
            System.out.println(i);
        }
    }
}</pre>
```

Exercice 5: Types et conversion

On considère les déclarations suivantes :

```
byte b1 = 10, b2 = 20;
short p = 200;
int n = 500;
long q = 100;
float x = 0.5f;
```

Donnez le type et la valeur des expressions arithmétiques suivantes :

```
1. b1 + b2

2. p + b1

3. b1 * b2

4. q + p * (b1 + b2)

5. x + q * n

6. b1 * q / x

7. b1 * q * 2. / x
```

Exercice 6:

Ecrire un programme affichant le code UNICODE (pour déterminer le nombre exact de caractères, on pourra consulter la classe Character). Pour temporiser l'affichage, vous utiliserer la méthode sleep() de la classe Thread.

Exercice 7:

Ecrire en Java un algorithme qui calcule les racinces d'un trinôme. Pour cela, vous définirez une classe trinôme avec les attributs appropriés et vous définirez les méthodes permettant le calcul des racinces d'un trinôme. Pour faciliter la programmation, vous ferez une méthode permettant