# Fiche d’exercices – Introduction

Pour chaque exercice ci-dessous, fournir le code **Java**, ainsi que la **modélisation UML**. Vous pouvez utiliser pour cela *Visual Paradigm for NetBeans***1**.

1. Construire une application permettant la saisie d’un tableau d’entiers (utiliser un tableau, pas un ArralyList) et l’affichage des valeurs saisies, ainsi que de la moyenne et les valeurs maximale et minimale saisies.
2. Définir une classe Etudiant, représentant une personne qui étudie au collègue. Comme n’importe quelle personne, un étudiant au collègue a un nom, un prénom, un téléphone et une adresse de contact. Il a aussi 4 notes au cours de l’année. On doit pouvoir calculer la moyenne de l’étudiant. Si cette moyenne est supérieure à 10, l’étudiant est ADMIS (résultat="admis"), sinon il est NON ADMIS.

Réaliser une application qui demande à l’utilisateur final les notes d’un étudiant et qui affiche ensuite s’il est admis ou pas.

1. Etudier et comprendre l’exemple de Calculette extensible donné en cours. Implémenter le code. Créer une nouvelle opération « ^ » : x^y équivaut à xy.
2. Analyser le code ci-dessous : est-il conforme aux critères de qualités énoncés en cours ? Si non, critiquer et corriger le code. Justifier vos positions.

|  |  |
| --- | --- |
| 12345678910111213141516171819202122232425262728293031323334 | public class Employe { String nom; String prenom; float salaire; float primes[]; public Employe(String nom, String prenom) { this.nom=nom; this.prenom=prenom; this.salaire = (float)1430.22; //smic en 1/1/2013 primes = null; } public void addnom(String val) { this.nom=val; } public String getnom() { return this.nom; } public void setnom(String nom) { this.nom=nom; } public String getprenom() { return this.prenom; } public float[] getPrimes() { return this.primes; }  public void setPrimes(float[] primes) { this.primes = primes; };  public void setSalaire(float salaire) { this.salaire = salaire; }  public float getSalaire() { return this.salaire; } public float getSalaireFinal() { float salfinal = this.salaire; for (int i=0; i<this.primes.length; i++) { salfinal += this.primes[i]; }  return salfinal; }  public int compareTo(Object o) { if (o != null) { String nom = ((Employe)o).nom; return this.nom.compareTo(nom) ; }  else return 0;  }}  |

**1 Observations**:

* Le modèle UML fourni doit correspondre au code Java présenté
* Lorsqu’il vous sera demandé de rendre l’exercice, vous devez compacter la totalité du projet NetBeans (y compris le projet Visual Paradigm) dans un fichier zip portant votre nom et le numéro de l’exercice. Exemple : Nom-FicheIntro-Exercice1.zip