

Tâche 1 : Votre but est de **compléter** les instructions des méthodes suivantes :

- `double getValue()` : retourne la valeur (`value`) de la note.
- `String toString()` : retourne une représentation de la note sous forme de chaîne de caractères. Pour une note ayant une valeur 12, cette méthode devra retourner la chaîne de caractères : `"12.0/20"`.
- `static Grade averageGrade(List<Grade> grades)` : calcule et renvoie la moyenne d'une liste de notes.

```
1 public double getValue() {
2     return value;
3 }
4 public String toString() {
5     return value + "/" + MAXIMUM_GRADE;
6 }
7 public static Grade averageGrade(List<Grade> grades){
8     double sumOfGrades = 0.;
9     for(Grade grade : grades){
10        sumOfGrades += grade.getValue();
11    }
12    return new Grade(sumOfGrades/grades.size());
13 }
```

Tâche 2 : Votre but est de **compléter** les instructions des méthodes suivantes :

- `Grade getGrade()` : retourne la note associée au résultat.
- `String toString()` : renvoie le nom de l'unité d'enseignement suivi de `" : "` suivi de la représentation en chaîne de caractère de la note. Par exemple, un résultat d'une UE de Programmation 2 avec une note de 20 devra renvoyer la chaîne de caractères suivante : `"Programmation 2 : 20.0/20"`.

```
1 public Grade getGrade() {
2     return grade;
3 }
4 public String toString() {
5     return teachingUnitName + " : " + grade;
6 }
```

Tâche 3 : Votre but est de compléter les instructions des méthodes suivantes :

- `void addResult(String teachingUnitName, Grade grade)` : ajoute un nouveau résultat à partir du nom de l'UE et d'une note.
- `List<Grade> getGrades()` : renvoie la liste des notes associées aux résultats de l'étudiant.
- `String toString()` : renvoie le nom de l'étudiant, c'est-à-dire son prénom, suivi d'un espace, suivi de son nom.
- `Grade averageGrade()` : renvoie la moyenne des notes associées aux résultats de l'étudiant.
- `void printResults()` : affiche les résultats de l'étudiant en sortie standard.

```
1 public void addResult(String teachingUnitName, Grade grade){
2     results.add(new TeachingUnitResult(teachingUnitName, grade));
3 }
4 public List<Grade> getGrades(){
5     List<Grade> grades = new ArrayList<>();
6     for(TeachingUnitResult result : results)
7         grades.add(result.getGrade());
8     return grades;
9 }
10 public String toString() {
11     return firstName + " " + lastName;
12 }
13 public Grade averageGrade() {
14     return Grade.averageGrade(getGrades());
15 }
16 public void printResults(){
17     printName();
18     for (TeachingUnitResult result : results)
19         System.out.println(result);
20     printAverageGrade();
21 }
22 private void printName() {
23     System.out.println(this);
24 }
25 private void printAverageGrade() {
26     System.out.println("Note moyenne : " + averageGrade());
27 }
```

Tâche 4 : Votre but est de compléter les instructions des méthodes suivantes :

- `public void addStudent(Student student)` : ajoute un étudiant à la promotion.
- `public List<Student> getStudents()` : renvoie la liste des étudiants de la promotion.
- `String toString()` : retourne une représentation de la promotion correspondant à son nom.
- `public void printStudentsResults()` : affiche les résultats de l'étudiant en sortie standard.

```
1 public void addStudent(Student student){
2     students.add(student);
```

```

3   }
4   public List<Student> getStudents(){
5       return List.copyOf(students);
6   }
7   public String toString() {
8       return name;
9   }
10  public void printStudentsResults(){
11      printName();
12      System.out.println();
13      for(Student student : students){
14          student.printResults();
15          System.out.println();
16      }
17  }
18  private void printName(){
19      System.out.println(this);
20  }

```

Tâche 5 : Ajoutez dans votre classe `Main` le code d'une méthode `public static void main(String[] args)` qui :

1. crée des instances de `Student` ayant les nom et prénom du membre du projet plus un étudiant fictif,
2. ajoute à ces étudiants les notes en "Programmation 2" et "Structures discrètes" que vous aimeriez avoir,
3. crée une promotion (instance de `Cohort`) ayant nommée "L2 informatique",
4. ajoute les étudiants créés à la promotion et
5. affiche les résultats de la promotion.

```

1   public static void main(String[] args){
2       Student student1 = new Student("Jean-Michel", "Bruitage");
3       Student student2 = new Student("David", "Goodenough");
4       student1.addResult("Programmation 2", new Grade(20));
5       student2.addResult("Programmation 2", new Grade(0));
6       student1.addResult("Structures discrètes", new Grade(20));
7       student2.addResult("Structures discrètes", new Grade(0));
8       Cohort cohort = new Cohort("L2 informatique");
9       cohort.addStudent(student1);
10      cohort.addStudent(student2);
11      cohort.printStudentsResults();
12  }

```