

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.

```
public double getValue() {  
    return value;  
}  
public String toString() {  
    return value + "/" + MAXIMUM_GRADE;  
}  
public static Grade averageGrade(List<Grade> grades){  
    double sumOfGrades = 0.;  
    for(Grade grade : grades){  
        sumOfGrades += grade.getValue();  
    }  
    return new Grade(sumOfGrades/grades.size());  
}
```

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".

```
public Grade getGrade() {  
    return grade;  
}  
public String toString() {  
    return teachingUnitName + " : " + grade;  
}
```

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.

```
public void addResult(String teachingUnitName, Grade grade){
    results.add(new TeachingUnitResult(teachingUnitName, grade));
}
public List<Grade> getGrades(){
    List<Grade> grades = new ArrayList<>();
    for(TeachingUnitResult result : results)
        grades.add(result.getGrade());
    return grades;
}
public String toString() {
    return firstName + " " + lastName;
}
public Grade averageGrade() {
    return Grade.averageGrade(getGrades());
}
public void printResults(){
    printName();
    for (TeachingUnitResult result : results)
        System.out.println(result);
    printAverageGrade();
}
private void printName() {
    System.out.println(this);
}
private void printAverageGrade() {
    System.out.println("Note moyenne : " + averageGrade());
}
}
```

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.

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

```

public List<Student> getStudents(){
    return List.copyOf(students);
}
public String toString() {
    return name;
}
public void printStudentsResults(){
    printName();
    System.out.println();
    for(Student student : students){
        student.printResults();
        System.out.println();
    }
}
private void printName(){
    System.out.println(this);
}

```

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.

```

public static void main(String[] args){
    Student student1 = new Student("Jean-Michel", "Bruitage");
    Student student2 = new Student("David", "Goodenough");
    student1.addResult("Programmation 2", new Grade(20));
    student2.addResult("Programmation 2", new Grade(0));
    student1.addResult("Structures discrètes", new Grade(20));
    student2.addResult("Structures discrètes", new Grade(0));
    Cohort cohort = new Cohort("L2 informatique");
    cohort.addStudent(student1);
    cohort.addStudent(student2);
    cohort.printStudentsResults();
}

```