

```
1  /* -----
2  Laboratoire :      POO Labo 6 - Ecole
3  Fichier :        Main.java
4  Auteur(s) :      Allemano Enzo & Romano Malo
5  Date :           15.11.2021
6
7  But :            Prmet de tester les implémentations de nos classes
8
9  Remarque(s) :
10
11  Compilateur :     Java 1.8
12
13  ----- */
14  public class Main {
15
16      public static void main(String [] args) {
17          /**
18           * Déclration de 2 professeurs
19           */
20          Professeur Donini = new Professeur("Donini", "Pier", "PDO");
21          Professeur Evequoz = new Professeur("Evequoz", "Claude", "CEZ");
22
23          /**
24           * Déclaration de 5 leçons
25           */
26          Lecon POO1_1 = new Lecon("POO1", 2, 2, 2, "G02", Donini);
27          Lecon POO1_2 = new Lecon("POO1", 2, 5, 2, "H01", Donini);
28          Lecon POO1_3 = new Lecon("POO1", 3, 2, 2, "G03", Donini);
29
30          Lecon PLP_1 = new Lecon("PLP", 2, 7, 2, "H01", Evequoz);
31
32          Lecon TIC_1 = new Lecon("TIC", 1, 9, 1, "F06");
33
34          /**
35           * Assignment des leçons aux 2 professeurs
36           */
37          Donini.definirLecons(POO1_1, POO1_2, POO1_3);
38          Evequoz.definirLecons(PLP_1);
39
40          /**
41           * Déclration de 6 étudiants
42           */
43          Etudiant Lennon = new Etudiant("Lennon", "John", 1234);
44          Etudiant McCartney = new Etudiant("Mc Cartney", "Paul", 2341);
45          Etudiant Starr = new Etudiant("Starr", "Ringo", 3241);
46          Etudiant Harrison = new Etudiant("Harrison", "Georges", 4321);
47          Etudiant Waters = new Etudiant("Waters", "Roger", 1324);
48          Etudiant Gilmour = new Etudiant("Gilmour", "David", 4312);
49
50          /**
51           * Déclration de 2 groupes
52           */
53          Groupe IL6_1 = new Groupe(6, "IL", 1);
54          Groupe TS6_1 = new Groupe(6, "TS", 1);
55
56          /**
57           * Assignment des groupes aux étudiants
58           */
59          Lennon.setGroupe(IL6_1);
60          McCartney.setGroupe(IL6_1);
61          Starr.setGroupe(IL6_1);
62          Harrison.setGroupe(IL6_1);
63
64          Waters.setGroupe(TS6_1);
65          Gilmour.setGroupe(TS6_1);
66
67          /**
68           * Assignment des étudiants aux groupes
69           */
70          IL6_1.definirEtudiants(Lennon, McCartney, Starr, Harrison);
71          TS6_1.definirEtudiants(Waters, Gilmour);
72      }
```

```
73      /**
74       * Assignation des leçons aux groupes
75       */
76      IL6_1.definirLecons(POO1_1, POO1_2, POO1_3, PLP_1, TIC_1);
77      TS6_1.definirLecons(POO1_1, POO1_2, POO1_3);
78
79      /**
80       * Déclaration du tableau des membres de l'école
81       */
82      Personne membresEcole[] = new Personne[]{Donini, Evequoz, Lennon, McCartney,
83                                              Starr, Harrison, Waters, Gilmour};
84
85      /**
86       * Affichage des membres de l'école
87       */
88      System.out.println("-- Membres de l'ecole\n");
89      for(Personne membre : membresEcole){
90          System.out.println(membre);
91      }
92
93      /**
94       * Affichage des infos et de l'horaire du groupe IL6-1
95       */
96      System.out.println(IL6_1.horaire());
97
98      /**
99       * Affichage de l'horaire de Donini
100      */
101      System.out.println(Donini.horaire());
102
103  }
104 }
105
```

```
1  /* -----
2  Laboratoire :      POO Labo 6 - Ecole
3  Fichier :         Personne.java
4  Auteur(s) :       Allemano Enzo & Romano Malo
5  Date :            15.11.2021
6
7  But :             Definit la classe abstraite Personne, ses attributs
8                    et les méthodes qui lui sont possibles
9
10 Remarque(s) :
11
12 Compilateur :      Java 1.8
13
14 ----- */
15
16 public abstract class Personne {
17     private String nom;
18     private String prenom;
19
20     /**
21      * Créer une personne
22      * @param nom Nom de la personne
23      * @param prenom Prénom de la personne
24      */
25     public Personne(String nom, String prenom) {
26         this.nom = nom;
27         this.prenom = prenom;
28     }
29
30     /**
31      * Obtenir les infos d'une personne au format texte
32      * @return Infos de la personne au format [nom] [prenom]
33      */
34     public String toString(){
35         return this.prenom + " " + this.nom;
36     }
37 }
38
```

```
1  /* -----
2  Laboratoire :      POO Labo 6 - Ecole
3  Fichier :        Professeur.java
4  Auteur(s) :      Allemano Enzo & Romano Malo
5  Date :           15.11.2021
6
7  But :            Definit la classe Professeur, ses attributs et
8                  les méthodes qui lui sont possibles
9
10 Remarque(s) :
11
12 Compilateur :     Java 1.8
13
14 ----- */
15
16
17 public class Professeur extends Personne {
18     private String abreviation;
19     private Lecon[] lecons;
20
21     /**
22      * Créer un professeur
23      * @param nom Nom du professeur
24      * @param prenom Prénom du professeur
25      * @param abreviation Abréviation du professeur
26      */
27     public Professeur(String nom, String prenom, String abreviation) {
28         super(nom, prenom);
29         this.abreviation = abreviation;
30     }
31
32     /**
33      * Obtenir l'abréviation du professeur
34      * @return L'abréviation du professeur
35      */
36     public String abreviation() {
37         return this.abreviation;
38     }
39
40     /**
41      * Obtenir les infos du professeur au format texte
42      * @return Infos du professeur au format Prof. [prenom] [nom] ([abreviation])
43      */
44     public String toString() {
45         return "Prof. " + super.toString() + " (" + abreviation() + ")";
46     }
47
48     /**
49      * Définir les leçons du professeur
50      * @param lecons Tableau des leçons
51      */
52     public void definirLecons(Lecon ... lecons) {
53         this.lecons = lecons;
54     }
55
56
57     /**
58      * Obtenir l'horaire
59      * @return L'horaire du professeur en 2 dimensions au format texte
60      */
61     public String horaire() {
62         return "\n-- Horaire " + toString() + "\n\n" + Lecon.horaire(lecons);
63     }
64
65 }
66
```

```
1  /* -----
2  Laboratoire :      POO Labo 6 - Ecole
3  Fichier :        Etudiant.java
4  Auteur(s) :      Allemano Enzo & Romano Malo
5  Date :          15.11.2021
6
7  But :            Definit la classe Etudiant, ses attributs et les méthodes qui lui sont possibles
8
9  Remarque(s) :
10
11  Compilateur :     Java 1.8
12
13  ----- */
14
15  public class Etudiant extends Personne{
16      private int matricule;
17      private Groupe groupe;
18
19      /**
20       * Créer un étudiant
21       * @param nom Nom de l'étudiant
22       * @param prenom Prénom de l'étudiant
23       * @param matricule Matricule de l'étudiant
24       */
25      public Etudiant(String nom, String prenom, int matricule) {
26          super(nom, prenom);
27          this.matricule = matricule;
28      }
29
30      /**
31       * Définir le groupe d'un étudiant
32       * @param groupe Groupe
33       */
34      public void setGroupe(Groupe groupe) {
35          this.groupe = groupe;
36      }
37
38      /**
39       * Obtenir les infos de l'étudiant au format texte
40       * @return Infos de l'étudiant au format Etud. [prenom] [nom] ([matricule]) - [groupe]
41       */
42      public String toString(){
43          return "Etud. " + super.toString() + " (" + this.matricule + ") - " + groupe.nom() ;
44      }
45  }
46
```

```

1  /* -----
2  Laboratoire :      POO Labo 6 - Ecole
3  Fichier :        Lecon.java
4  Auteur(s) :      Allemano Enzo & Romano Malo
5  Date :           15.11.2021
6
7  But :             Definit la classe Lecon, ses attributs et les méthodes qui lui sont possibles.
8                    Définit également la méthode statique permettant de générer un horaire au
                      format texte.
9
10 Remarque(s) :
11
12 Compilateur :     Java 1.8
13
14 ----- */
15
16 public class Lecon {
17
18     private String matiere;
19     private int jourSemaine;
20     private int periodeDebut;
21     private int duree;
22     private String salle;
23     private Professeur professeur;
24
25     /**
26      * Créer une leçon sans professuer
27      * @param matiere Matière de la leçon
28      * @param jourSemaine Jour de la leçon
29      * @param periodeDebut Préiode de début de la leçon
30      * @param duree Durée de la leçon
31      * @param salle Salle de la leçon
32      */
33     public Lecon(String matiere, int jourSemaine, int periodeDebut,
34                  int duree, String salle) {
35         this.matiere = matiere;
36         this.jourSemaine = jourSemaine;
37         this.periodeDebut = periodeDebut;
38         this.duree = duree;
39         this.salle = salle;
40     }
41
42     /**
43      * Créer une leçon avec un professeur
44      * @param matiere Matière de la leçon
45      * @param jourSemaine Jour de la leçon
46      * @param periodeDebut Préiode de début de la leçon
47      * @param duree Durée de la leçon
48      * @param salle Salle de la leçon
49      * @param professeur Professeur de la leçon
50      */
51     public Lecon(String matiere, int jourSemaine, int periodeDebut,
52                  int duree, String salle, Professeur professeur){
53         this(matiere, jourSemaine, periodeDebut, duree, salle);
54         this.professeur = professeur;
55     }
56
57     /**
58      * Méthode statique de génération d'un calendrier au format texte d'un tableau de leçons.
59      * @param lecons Tableau de leçons
60      * @return Horaire des leçons au format texte
61      */
62     public static String horaire(Lecon ... lecons){
63         StringBuilder calendrier = new StringBuilder();
64         StringBuilder ligneCourante = new StringBuilder();
65         boolean afficherCelluleVide = true;
66         boolean afficheFinDePeriode = true;
67         String[] periode = new String[]{"8:30","9:15","10:25","11:15","12:00","13:15",
68                                         "14:00","14:55","15:45","16:35","17:20"};
69         calendrier.append(String.format("%76s","| Lun          | Mar          | Mer          " +
70                                         "| Jeu          | Ven          |"))
71         calendrier.append(String.format("%76s","|-----|-----|-----" +

```

```

72         "|-----|-----|)").append("\n");
73
74     for (int i = 0; i < 11;++i){
75         ligneCourante.append(String.format("%5s",periode[i]));
76         for (int j = 0; j < 5;++j){
77             ligneCourante.append("|");
78
79             for (Lecon l : lecons){
80                 if (l.jourSemaine == j && l.periodeDebut == i){
81                     ligneCourante.append(String.format("%-5s %s", l.matiere ,
82                         l.salle + (l.professeur == null ? " " : " "
83                             + l.professeur.abreviation())));
84                     afficherCelluleVide = false;
85                 }
86             }
87
88             if(afficherCelluleVide){
89                 ligneCourante.append(String.format("%13s", " "));
90             }
91
92             afficherCelluleVide = true;
93
94         }
95         ligneCourante.append("\n");
96         ligneCourante.append(String.format("%5s", " "));
97
98         for (int j = 0; j < 5;++j){
99             for (Lecon l : lecons){
100                 if (l.jourSemaine == j && (l.periodeDebut <= i &&
101                     l.periodeDebut + l.duree -1 > i)){
102                     ligneCourante.append(String.format("|%13s", " "));
103                     afficheFinDePeriode = false;
104                 }
105             }
106
107             if (afficheFinDePeriode){
108                 ligneCourante.append("|-----");
109             }
110
111             afficheFinDePeriode = true;
112         }
113
114         calendrier.append(ligneCourante).append("\n");
115         ligneCourante.setLength(0); //Efface le contenu
116     }
117
118     return calendrier.toString();
119 }
120 }
121

```

```

1  /* -----
2  Laboratoire :      POO Labo 6 - Ecole
3  Fichier :        Groupe.java
4  Auteur(s) :      Allemano Enzo & Romano Malo
5  Date :           15.11.2021
6
7  But :             Definit la classe Groupe, ses attributs et les méthodes qui lui sont possibles
8
9  Remarque(s) :
10
11 Compilateur :     Java 1.8
12
13 ----- */
14
15
16 public class Groupe {
17     private final int numero;
18     private final String orientation;
19     private final int trimestre;
20     private Lecon[] lecons;
21     private Etudiant[] etudiants;
22     private int nbEtudiants = 0;
23
24     /**
25      * Créer un groupe
26      * @param numero Numéro du groupe
27      * @param orientation Orientation du groupe
28      * @param trimestre Trimestre du groupe
29      */
30     public Groupe(int numero, String orientation, int trimestre) {
31         this.numero = numero;
32         this.orientation = orientation;
33         this.trimestre = trimestre;
34     }
35
36     /**
37      * Obtenir le nom du groupe
38      * @return Le nom du groupe au format [orientation][numero]-[trimestre], exemple : IL6-1
39      */
40     public String nom() {
41         return orientation + numero + "-" + trimestre;
42     }
43
44     /**
45      * Obtenir le nombre d'étudiants du groupe
46      * @return Le nombre d'étudiants
47      */
48     public int nombreEtudiants() {
49         return nbEtudiants;
50     }
51
52     /**
53      * Définir les lecons suivies par le groupe
54      * @param lecons Tableau des leçons suivies par le groupe
55      */
56     public void definirLecons(Lecon ... lecons) {
57         this.lecons = lecons;
58     }
59
60     /**
61      * Définir les étudiants membres du groupe
62      * @param etudiants Etudiants du groupe
63      */
64     public void definirEtudiants(Etudiant ... etudiants) {
65         this.etudiants = etudiants;
66         nbEtudiants = etudiants.length;
67     }
68
69     /**
70      * Obtenir l'horaire
71      * @return Horaire du groupe en 2 dimensions au format texte
72      */

```



```
73     public String horaire(){
74         return "\n-- Horaire du groupe " + nom() + " (" + nombreEtudiants()
75             + " étudiants)\n\n" + Lecon.horaire(lecons);
76     }
77 }
78
```