

```

1  # eleve.rb : Eleve est une représentation d'un élève de S1 à JDLM
2  class Eleve
3
4      attr_reader :numero, :nom, :prenom, :sexe, :programme, :natation, :anglais,
      :groupe, :cours_ang
5      attr_reader :musique, :cours_musique
6
7      def initialize(params) #params est un Hash qui contiendra les éléments requis pour
      construire un Eleve
8
9          @numero = params[:numero]
10         @nom = params[:nom]
11         @prenom = params[:prenom]
12         @sexe = params[:sexe]
13         @programme = params[:programme]
14         @natation = params[:natation]
15         @anglais = params[:anglais]
16
17         case @@regles
18         when S1
19             @musique = ""
20         when S2
21             @musique = params[:musique]
22         end
23
24         #configurer groupes_permis
25         @groupes_permis = @@regles[@programme] & @@regles[@anglais]
26
27         case @@regles
28         when S2
29             @groupes_permis = @groupes_permis & @@regles[@musique]
30         end
31     end
32
33     def to_s
34         if @groupes_permis.empty? == true
35             then groupes = "AUCUN!"
36         else groupes = @groupes_permis.join(", ")
37         end
38
39         if @groupe.nil?
40             then
41                 groupe = ""
42                 cours_ang = ""
43                 cours_musique = ""
44             else
45                 groupe = @groupe
46                 cours_ang = @cours_ang
47                 cours_musique = @cours_musique
48             end
49
50         "#{@numero}\t#{@nom.ljust(20, " ")}#{@prenom.ljust(20, " ")}" +
51         "#{@sexe}\t#{@programme}\t#{@natation}\t#{@anglais.ljust(4, " ")}\t" +

```

```

52         "#{@musique.ljust(4, " ")}\t<#{groupe}> <#{cours_ang}>\t
    <#{cours_musique}>\t" +
53         "[#{groupes}]"
54     end
55
56     def assigner_groupe(groupe)
57         if @groupes_permis.include?(groupe)
58             then
59                 @groupe = groupe
60                 assigner_ang
61                 assigner_musique
62             end
63         end
64
65         def assigner_groupe_au_hasard
66             @groupe = @groupes_permis[rand(@groupes_permis.size)]
67             assigner_ang
68             assigner_musique
69         end
70
71         def self.regles(niveau)
72             @@regles = niveau
73         end
74
75         def self.ang(ang)
76             @@ang = ang
77         end
78
79         def self.mus(musique)
80             @@mus = musique
81         end
82
83     private
84         def assigner_ang
85             @cours_ang = @@ang["#{@groupe}-#{@anglais}"]
86         end
87
88         def assigner_musique
89             case @@regles
90             when S2
91                 @cours_musique = @@mus["#{@groupe}-#{@musique}"]
92             end
93         end
94     end
95
96

```