

```

1  # eleve.rb
2  #
3  # Définissons un Class qui ne nomme Eleve.
4  # Eleve sera une représentation d'un élève de S1 à JDLM
5  #
6
7  class Eleve
8
9      def initialize(params) #params est un Hash qui contiendra les éléments requis pour construire un Eleve
10
11          @numero = params[:numero]
12          @nom = params[:nom]
13          @prenom = params[:prenom]
14          @sexe = params[:sexe]
15          @programme = params[:programme]
16          @natation = params[:natation]
17          @anglais = params[:anglais]
18      end
19
20      attr_reader :numero, :nom, :prenom, :sexe, :programme, :natation, :anglais
21
22      def to_s
23          "#{@numero}\t#{@nom.ljust(20, " ")}\t" +
24          "#{@prenom.ljust(20, " ")}\t#{@sexe}\t#{@programme}\t#{@natation}\t#{@anglais}"
25      end
26  end
27
28  #---- PROGRAMME PRINCIPAL
29
30  # Exemple 1 : Création d'un objet Eleve.
31  puts "Exemple 1"
32  params = { :numero => '007',
33            :nom => "Bond",
34            :prenom => "James",
35            :sexe => "M",
36            :programme => "enr",
37            :natation => "12",
38            :anglais => 'ela'
39  }

```

```

40  james = Eleve.new(params)
41  puts james.nom
42
43  # Exemple 2 : Créer plusieurs objets Eleve lus du fichier s1.csv,
44  #               mettre en mémoire les objets Eleve dans le Array eleves,
45  #               afficher tous les objets Eleve.
46  puts "Exemple 2"
47  eleves = Array.new #Création d'un objet Array vide
48
49  f = File.open("s1.csv", "r")
50  while ligne = f.gets
51    ligne = ligne.chomp
52    infos = ligne.split(";")
53    params = { :numero => infos[0],
54              :nom => infos[1],
55              :prenom => infos[2],
56              :sexe => infos[3],
57              :programme => infos[4],
58              :natation => infos[5],
59              :anglais => infos[6]
60            }
61    eleves << Eleve.new(params)
62
63  end
64  f.close
65
66  # Afficher tous les objets Eleve
67  eleves.each {|eleve| puts eleve}
68
69  # Exemple 3 : Afficher tous les objets Eleves ayant un nom comptant moins de 4 lettres
70  puts "Exemple 3"
71  eleves.each do |eleve|
72    if eleve.nom.size <= 4
73      then puts eleve
74    end
75  end
76
77  # Exemple 4 : Compter et afficher le nombre de gars et de filles en S1
78  puts "Exemple 4"

```

```

79
80  n_gars = 0
81  n_filles = 0
82  eleves.each do |eleve|
83      if eleve.sexe == "M"
84          then n_gars = n_gars + 1
85      else n_filles = n_filles + 1
86      end
87  end
88  puts "Total #{eleves.size} : #{n_gars} gars et #{n_filles} filles "
89
90  # Exemple 5 : Compter et afficher le nombre de gars et de filles en enrichi en S1
91  puts "Exemple 5"
92  n_gars = 0
93  n_filles = 0
94  eleves.each do |eleve|
95      if eleve.sexe == "M" and eleve.programme == "enr"
96          then n_gars = n_gars + 1
97      end
98      if eleve.sexe == "F" and eleve.programme == "enr"
99          then n_filles = n_filles + 1
100     end
101  end
102  total_enr = n_gars + n_filles
103  puts "Enrichi #{total_enr} : #{n_gars} gars et #{n_filles} filles "
104
105  #
106  # Ce type de recherche est tellement commun, que Ruby nous offre une méthode très utile find_all
107  #
108
109  #Exemple 6 : Compter et afficher le nombre de gars en enrichi qui sont en eesl en anglais
110  puts "Exemple 6"
111  gars = eleves.find_all{|eleve| eleve.sexe == "M" and eleve.programme == "enr" and eleve.anglais == "EESL"}
112  puts "Gars, enr, eesl : #{gars.size}"
113
114  #Exemple 7 : Compter et afficher le nombre d'élèves avec un niveau de natation supérieur ou égal à 10.
115  puts "Exemple 6"
116  natation = eleves.find_all{|eleve| eleve.natation.to_i >= 10}
117  puts "Natation >= 10 : #{natation.size}"

```

```
118
119 #Exercices
120 # a) Compter et afficher le nombre d'élèves dans le programme reg et anglais ELA
121 # b) Compter et afficher le nombre d'élèves dans le programme reg et anglais EESL
122 # c) Compter et afficher le nombre d'élèves dans le programme reg et anglais ESL
123 # d) Compter et afficher le nombre d'élèves dans le programme enr et anglais ELA
124 # e) Compter et afficher le nombre d'élèves dans le programme enr et anglais EESL
125 # f) Compter et afficher le nombre d'élèves dans le programme enr et anglais ESL
126
127 puts "Exercices"
128
129
```