Exemple: grup d'alumnes

```
public class Alumne {
   private long dni;
   private double nota;
   private String nom;
pubic class Grup {
     public static final int NUM_ALUMNES = 50;
     private Alumne[] grup;
private int numAl;
     public Grup() {
           grup = new Alumne[NUM_ALUMNES];
numAl = 0;
                                                             public class TestGrup {
   public static void main(String[] args) {
     public void afegir(Alumne a) {
   if (numAl < NUM_ALUMNES) {
      grup[numAl++] = a;
}</pre>
                                                                      Grup grupB = new Grup();
                                                                       /* Afegim alumnes ... */
     }
                                                                                                               grupB
     public boolean eliminar(Alumne a) {
           int i = cercar(a);
boolean trobat = i != -1;
                                                                                         numAl
                                                                                                grup
           4
                numAl--;
                                                       0
                                                                         3
                grup[numAl] = null;
                                                                               nu11
                                                                                           nu11
                                                                                     nu11
                                                                                                        nu11
           return trobat;
     }
                         12123123
                                                   72678256
                                                                           19456754
                                                                                                   18547645
}
                          "Vicent Juan"
                                                   "Josep Lluc"
                                                                            "Anna Flors"
                         8.5
                                                   10
                                                                             3
                                                                                                    10
```

Exemple: grup d'alumnes

- Es demana:
- Mètode recursiu que mostre per pantalla la informació de tots els alumnes.
- Mètode recursiu que torne la nota mínima.
- Mètode recursiu que compte el nombre de Matrícules d'Honor (MH).
- Mètode recursiu que torne la posició del primer alumne amb MH o -1 si no hi ha cap MH.
- Mètode recursiu que torne la posició del darrer alumne amb MH o -1 si no hi ha cap MH.
- Mètode recursiu que comprove si els alumnes estan ordenats ascendentment per nota (de menor a major).
- Mètode recursiu que torne la posició d'un alumne de nom donat, sabent que els alumnes estan ordenats per nom (lèxicogràficament), o -1 si no el troba.

28/01/2020



PRG

etsinf

Escuela Técnica
apperior de Ingeniaria
linformática

2

• Mètode recursiu que mostra per pantalla la informació de tots els alumnes.

• Mètode recursiu que torna la nota mínima.

```
public double notaMinima() { return notaMinima(0); }

/** 0 < numAl i 0 <= pos < numAl */
private double notaMinima(int pos) {

Recorregut recursiu ascendent

PRG

PRG

PRG

3</pre>
```

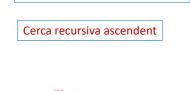
• Mètode recursiu que compta el nombre de Matrícules d'Honor (MH).

```
public int comptarMH() { return comptarMH(numAl - 1); }

/** -1 <= pos < numAl */
private int comptarMH(int pos) {

Recorregut recursiu descendent</pre>
```

 Mètode recursiu que torna la posició del primer alumne amb MH o -1 si no hi ha cap MH.
 Recorregut recursiu descendent



28/01/2020 UNI

PRG

 Mètode recursiu que torna la posició del primer alumne amb MH o -1 si no hi ha cap MH.

```
public int primeraMH() { return primeraMH(numAl - 1); }

/** -1 <= pos < numAl */
private int primeraMH(int pos) {

Recorregut recursiu descendent

Versió 1

public int primeraMH() { return primeraMH(numAl - 1, -1); }

/** -1 <= pos < numAl i -1 <= primMH < numAl */
private int primeraMH(int pos, int primMH) {

Recorregut recursiu descendent

Versió 2
```

 Mètode recursiu que torna la posició del primer alumne amb MH o -1 si no hi ha cap MH.

```
public int primeraMH() { return primeraMH(0); }

/** 0 <= pos <= numAl */
private int primeraMH(int pos) {

Cerca recursiva ascendent</pre>
```

 Mètode recursiu que torna la posició del darrer alumne amb MH o -1 si no hi ha cap MH.

```
public int darreraMH() { return darreraMH(numAl - 1); }
/** -1 <= pos < numAl */
private int darreraMH(int pos) {

Cerca recursiva descendent

28/01/2020

PRG

PRG

PRG

PRG

6</pre>
```

• Mètode recursiu que comprova si els alumnes estan ordenats ascendentment per nota (de menor a major).

```
public boolean ordreAsc() { return ordreAsc(0); }

/** 0 <= pos < numAl */
private boolean ordreAsc(int pos) {

Cerca recursiva ascendent</pre>
```

• Mètode recursiu que torna la posició d'un alumne de nom donat, sabent que els alumnes estan ordenats per nom (lèxicogràficament), o -1 si no el troba.