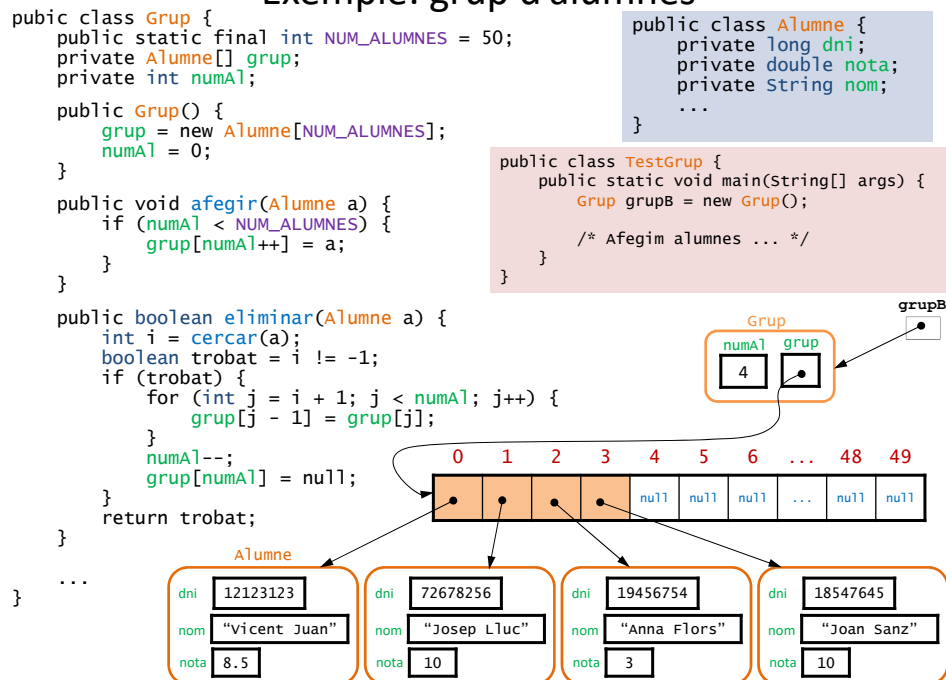


Exemple: grup d'alumnes



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 Matricules 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 comprovi 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.

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

```
public void mostrarAlumnes() { mostrarAlumnes(0); }

/** 0 <= pos <= numAl */
private void mostrarAlumnes(int pos) {
```

Recorregut recursiu ascendent

- 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



3

- Mètode recursiu que compta el nombre de Matricules d'Honor (MH).

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

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

Recorregut recursiu descendent

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

Recorregut recursiu descendent

Cerca recursiva ascendent



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

```
public int primeraMH() { return primeraMH(numA1 - 1); }

/** -1 <= pos < numA1 */
private int primeraMH(int pos) {
```

Recorregut recursiu descendent

Versió 1

```
public int primeraMH() { return primeraMH(numA1 - 1, -1); }

/** -1 <= pos < numA1 i -1 <= primMH < numA1 */
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 <= numA1 */
private int primeraMH(int pos) {
```

Cerca recursiva ascendent

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

```
public int darreraMH() { return darreraMH(numA1 - 1); }

/** -1 <= pos < numA1 */
private int darreraMH(int pos) {
```

Cerca recursiva descendent

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

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

```
/** 0 <= pos < numA1 */  
private boolean ordreAsc(int pos) {
```

Cerca recursiva ascendent

- 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.

```
public int encNom(String nom) { return encNom(nom, 0, numA1 - 1); }
```

```
/** 0 <= ini <= numA1 i -1 <= fi < numA1 */  
private int encNom(String nom, int ini, int fi) {
```

Cerca recursiva binària