



# Tema 7. Arrays: definició i aplicacions

Introducció a la Informàtica i a la Programació (IIP)

Curs 2019/20

Departament de Sistemes Informàtics i Computació



Duració: 9 sessions

# Continguts

1. Introducció

- La necessitat d'utilitzar arrays per a representar dades del mateix tipus
- Arrays en Java: declaració, creació, atribut length i operador []
- Accés a les components: directe i sequencial
- Ús d'un array com atribut, variable local, paràmetre o resultat d'un mètode
- 2. Representació i tractament de dades usant un array
  - Formes bàsiques de representació, en funció de l'accés directe o seqüencial a les dades, i operacions elementals associades
  - Recorregut d'un array: esquemes, exemples i exercicis
  - Cerca d'una dada donada en un array: esquemes, exemples i exercicis
- 3. Arrays multidimensionals: declaració, creació i accés a les components



Crea una carpeta Tema 7 dins de la teua carpeta W:\IIP\



- Descarrega (del Tema 7 de PoliformaT) els fitxers BlueJ exemples Tema 7.jar i exercicis Tema 7.zip. Descomprimeix aquest últim.
- Obre els projectes (des de l'opció Projecte de BlueJ, amb l'opció Obre Projecte... o fes clic en la icona de BlueJ) i prepara't per usar-los.

#### Introducció

- Sovint interessa emmagatzemar i referenciar variables que representen una col·lecció de valors, per poder tractar-los de manera uniforme.
- Exemples:
  - Obtenir estadístiques bàsiques sobre mesures diàries de la temperatura mitjana en una determinada zona geogràfica.
  - Gestionar una col·lecció d'objectes homogenis, per exemple, una classe Hospital que puga tenir associat un conjunt d'objectes de tipus Pacient.
  - **—** ...
- Java proporciona l'array com a mecanisme per agrupar dades de tipus homogeni (tant de tipus objecte com de tipus primitiu).

21/11/2019



IIP - Curs 2019/20



3

# Arrays unidimensionals

- Definició: Un array és una col·lecció d'elements homogenis (del mateix tipus de dades) agrupats de forma consecutiva en memòria.
- Característiques
  - Cada element d'un array té associat un índex, que és un nombre no negatiu que l'identifica inequívocament i permet accedir-hi.
  - L'amplària (nombre de components) de l'array s'ha d'establir en la seua declaració i és invariable al llarg de l'execució.
  - L'accés als elements d'un array és directe (gràcies al seu índex).
- Aquestes estructures de dades són adequades per a situacions en les que l'accés a les dades es realitze de forma aleatòria (és a dir, conjunts de dades que poden ser indexats) o seqüencial (posicionalment, una dada darrere de l'altra).



# Arrays unidimensionals

#### Declaració i ús

Valors: Es representen com una successió d'elements entre claus i separats per comes.

$$\{e_0, e_1, ..., e_{n-1}\}$$

Operador d'arrays: Operador d'accés a les components []

index és una expressió que s'avalua a un valor vàlid per l'array nom vble.

Declaració i creació:

Vàlid però no aconsellat

- Declaració: tipus[] nomVble; ó tipus nomVble[];
- Inicialització (amb components): nomVble = new tipus[amplària];
- tipus[] nomVble = new tipus[amplaria]; Conjuntament:

tipus és qualsevol tipus primitiu o referència, nomvble és qualsevol identificador vàlid i amplària és una expressió que s'avalua a un enter no negatiu i determina la quantitat de components.

21/11/2019



IIP - Curs 2019/20



5

# Arrays unidimensionals Declaració i ús

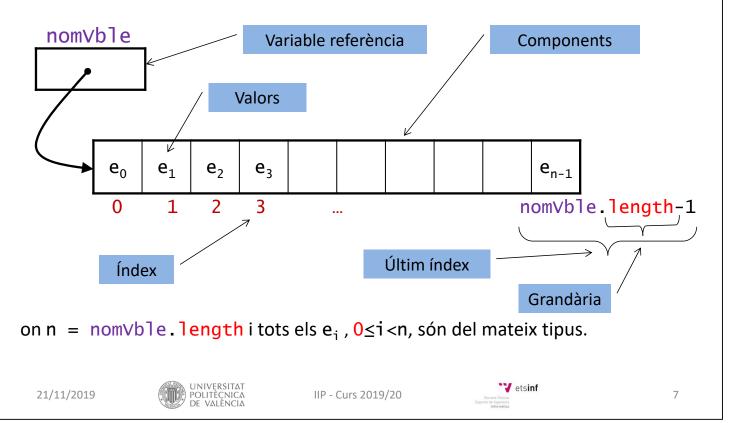
Tots els arrays disposen d'un atribut constant (de només consulta) length que indica la quantitat de components (amplària o grandària) d'un array:

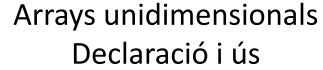
nomVble.length

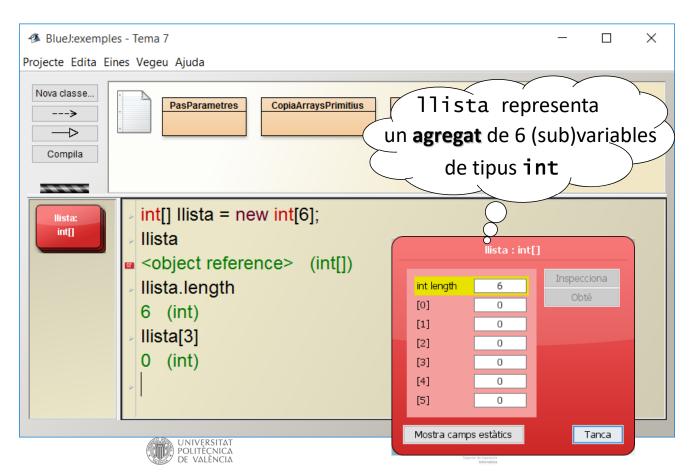
- Els índexs vàlids d'un array van des de 0 fins length-1.
- Una variable array es pot veure com una referència a la posició de memòria del primer element de l'array.
- Hi ha una referència especial que es pot utilitzar amb qualsevol array (o amb altres tipus referència): null.
- Tots els components d'arrays de tipus numèrics s'inicialitzen per defecte a 0.
- Tots els components d'arrays de tipus referència (p.e., un array de String) s'inicialitzen per defecte a null.

# Arrays unidimensionals Declaració i ús

• Gràficament podem veure un array de la manera següent :







#### **Arrays - Excepcions**

- En un array d'amplària N, els índexs vàlids per accedir als seus components pertanyen a l'interval [0, N-1].
- Un accés fora d'aquest interval produeix l'excepció en execució del tipus :

```
java.lang.ArrayIndexOutOfBoundsException
```

L'accés als components d'un array s'ha de controlar utilitzant les fites inferior
 (0) i superior (nom\ble.length-1) dels índexs.

```
int[] Ilista = new int[6];
Ilista
<object reference> (int[])
Ilista.length
6 (int)
Ilista[3]
0 (int)
Ilista[6]
Exception: java.lang.ArrayIndexOutOfBoundsException (6)
Ilista[-1]
Exception: java.lang.ArrayIndexOutOfBoundsException (-1)
```

# Arrays unidimensionals Declaració i ús - Exemples

```
    Array de 5 caràcters:
```

```
- Variable referència: char[] lletres;
```

- Objecte array (amb components): lletres = new char[5];
- Conjuntament: char[] lletres = new char[5];

```
1letres 'V' 'g' 'A' 'i' 'J' 0 1 2 3 4 1letres.length-1
```

```
lletres[0] = 'V';
lletres[1] = 'g';
lletres[2] = 'A';
lletres[3] = 'i';
lletres[4] = 'J';
```

Dos arrays d'enters:

```
int[] nums1 = new int[5000], nums2 = new int[50];
```

Array de 20 cadenes de caràcters: final int NUM = 10;

```
String[] noms = new String[NUM * 2];
```

• Array de reals tal que la quantitat de components es decideix per teclat:

```
double[] preus = new double[teclat.nextInt()];
```

Creació i inicialització d'un array de 4 enters: int[] v = {-5, 6, 10, 3};

DE VALLITOIAL

# Arrays - Memòria

- Una variable referència array (nomvble) utilitza la memòria on haja estat definida (p.e. al registre de activació del mètode on s'ha definit).
- Els components de l'array es creen en el monticle o heap que és la zona de memòria reservada per guardar variables dinàmiques.
- Les variables dinàmiques són variables que es creen en temps d'execució mitjançant l'operació de creació new.
- Els components només són accessibles si hi ha una referència a ells.
- És possible suggerir la destrucció d'un array v al desreferenciar-lo (v = null)
  - Recorda que hi ha un mecanisme de recollida de fem (garbage collector) que s'activa automàticament en temps d'execució i allibera la memòria que està ocupada però no és accessible al no estar referenciada.
  - També es pot suggerir la seua execució mitjançant la instrucció System.gc().

21/11/2019



IIP - Curs 2019/20



11

# Arrays unidimensionals - Assignació

Les components d'un array es poden vore com a variables del tipus dels elements de l'array.

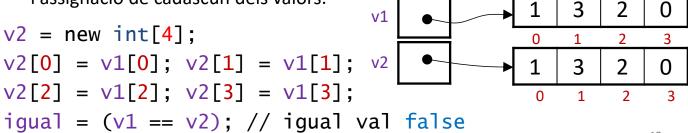
```
nomVble[index] = expressió;
```

expressió ha de ser del tipus dels components de l'array nomvble.

L'assignació entre arrays tan sols afecta a les referències.

```
int[] v1 = \{1, 3, 2, 0\};
int[] v2;
                                            1
                                                3
                                                    2
                                                        0
v2 = v1:
boolean igual = (v1 == v2):
   // iqual val true
```

Si es desitja una còpia d'un array s'han de crear nous components i realitzar l'assignació de cadascun dels valors.



#### Arrays unidimensionals – Mètodes

• Els arrays es defineixen com qualsevol paràmetre formal, indicant el tipus i el nom de la variable:

```
public static int metode1(int[] v1, int[] v2) { ... }
public static void main(String[] args) { ... }
```

En la crida, només s'utilitza el nom de la variable :

```
int[] a1 = new int[10], a2 = new int[5];
...
int i = mètode1(a1, a2);
```

- Quan s'invoca a un mètode amb algun argument de tipus referència només es còpia la referència del paràmetre real en el formal.
- Els mètodes poden retornar com a resultat un array (referència als components). Per exemple:

```
public static char[] mètode2(int[] v1) {
    char[] nou = new char[v1.length + 10];
    ...
    return nou;
}

UNIVERSITAT
POLITECNICA
DE VALENCIA
Invocació:
char[] res = mètode2(a1);
```

21/11/2019

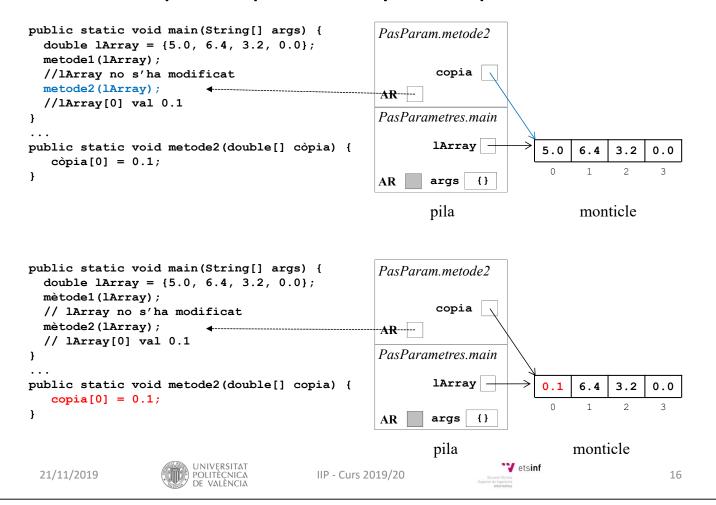
# Exemple de pas d'arrays com paràmetres

```
PasParametres - exemples - Tema 7
Classe Edita Eines Opcions
Compila Desfés Retalla Copia Enganxa Cerca... Tanca
                                                                   Implementació
  public class PasParametres {
       public static void main(String[] args) {
           double[] lArray = {5.0, 6.4, 3.2, 0.0};
           metodel(lArray);
           // l'array no ha sigut modificat en absolut
           metode2(lArray);
           // el primer component de l'array val ara 0.1
       public static void metodel(double[] copia) {
           copia = new double[4]; // Aquest array desapareix en acabar el metode
       public static void metode2(double[] copia) {
           copia[0] = 0.1;
                                                                                       canviat
```

# Exemple de pas d'arrays com paràmetres

```
public static void main(String[] args) {
                                                   PasParametres.main
  double lArray = \{5.0, 6.4, 3.2, 0.0\};
  metode1(lArray);
                                                          lArray -
  // lArray no s'ha modificat
                                                                                    3.2
                                                                                          0.0
  metode2(lArray);
  // lArray[0] val 0.1
                                                   AR
                                                          args
                                                                               monticle
                                                           pila
public static void main(String[] args) {
                                                   PasParam.metode1
  double lArray = \{5.0, 6.4, 3.2, 0.0\};
  metode1(lArray);
  // lArray no s'ha modificat
                                                           copia
                                                                               0.0
                                                                                    0.0
                                                                                          0.0
  metode2(lArray);
                                                   AR
  // lArray[0] val 0.1
                                                   PasParametres.main
public static void metodel(double[] copia) {
                                                          lArray -
   copia = new double[4];
                                                                          5.0
                                                                                     3.2
                                                                                          0.0
                                                   AR
                                                          args
                                                           pila
                                                                               monticle
                      UNIVERSITAT
                                                                   **/ etsinf
21/11/2019
                                        IIP - Curs 2019/20
                                                                                            15
```

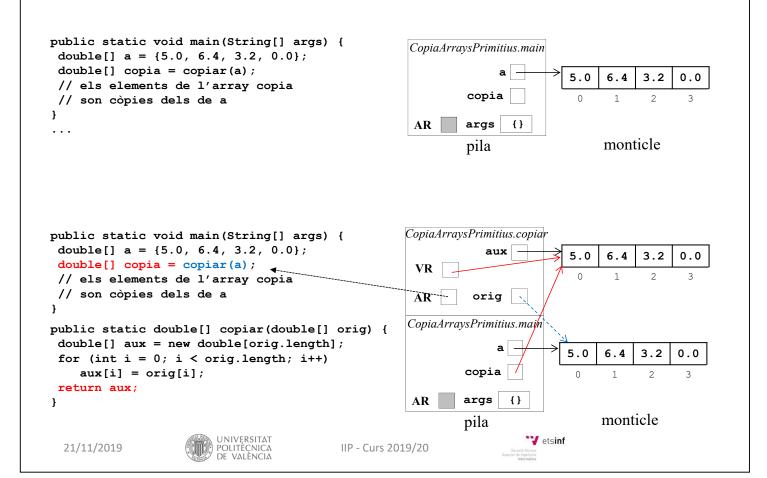
# Exemple de pas d'arrays com paràmetres

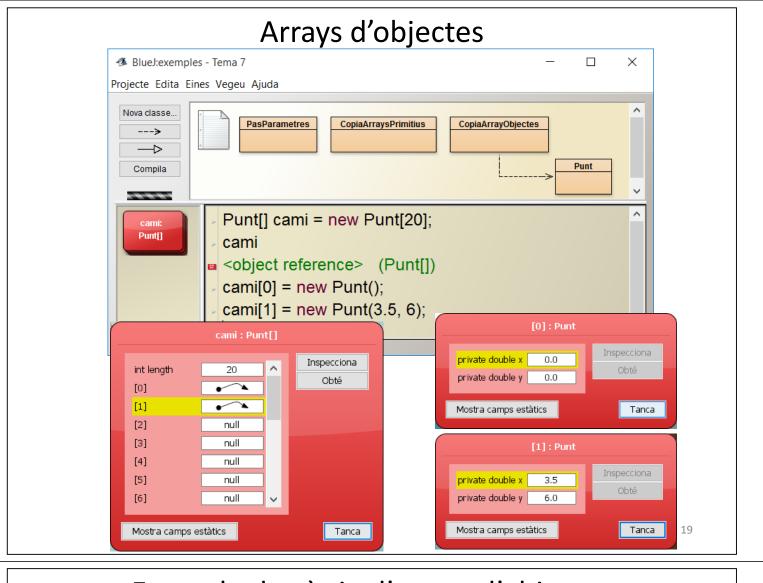


# Exemple de còpia d'arrays de tipus primitius

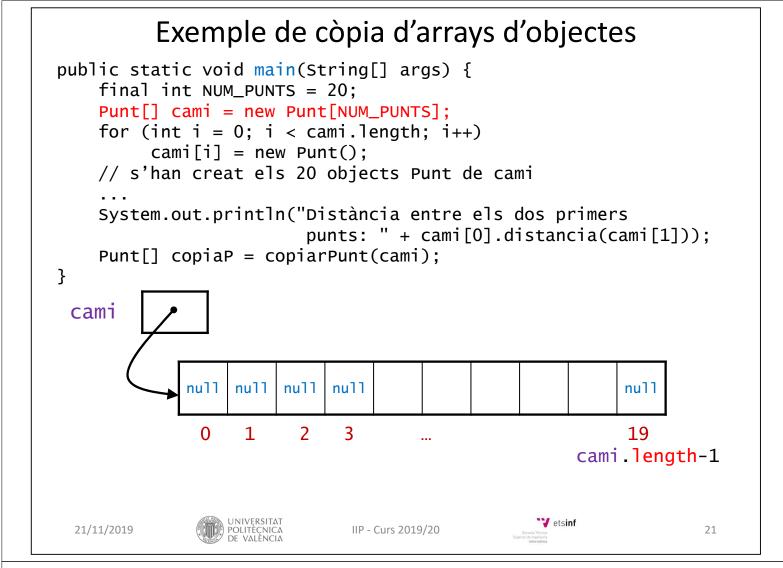
```
X
CopiaArraysPrimitius - exemples - Tema 7
Classe Edita Eines Opcions
Compila Desfés Retalla Copia Enganxa Cerca... Tanca
                                                           Implementació
  public class CopiaArraysPrimitius {
       public static void main(String[] args) {
            double[] a = {5.0, 6.4, 3.2, 0.0};
            double[] copia = copiar(a);
            // els elements de l'array copia son copies dels de a
       public static double[] copiar(double[] orig) {
            double[] aux = new double[orig.length];
            for (int i = 0; i < orig.length; i++) {
                 aux[i] = orig[i];
            return aux;
Classe compilada. No hi ha errors de sintaxi
                                                                           desat
```

# Exemple de còpia d'arrays de tipus primitius

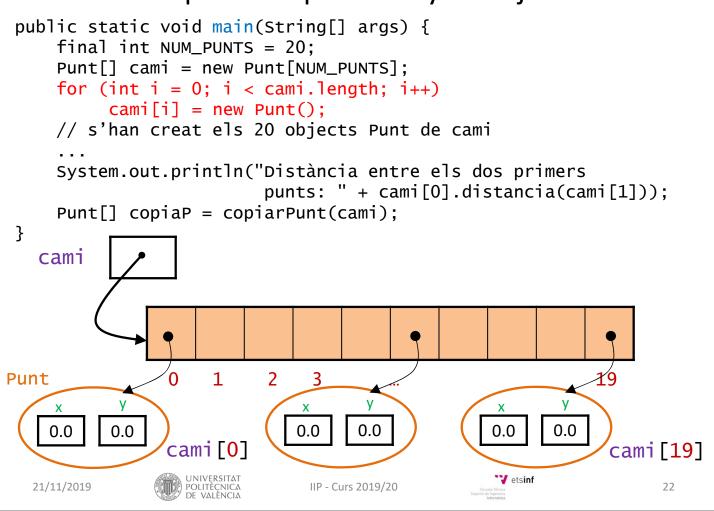


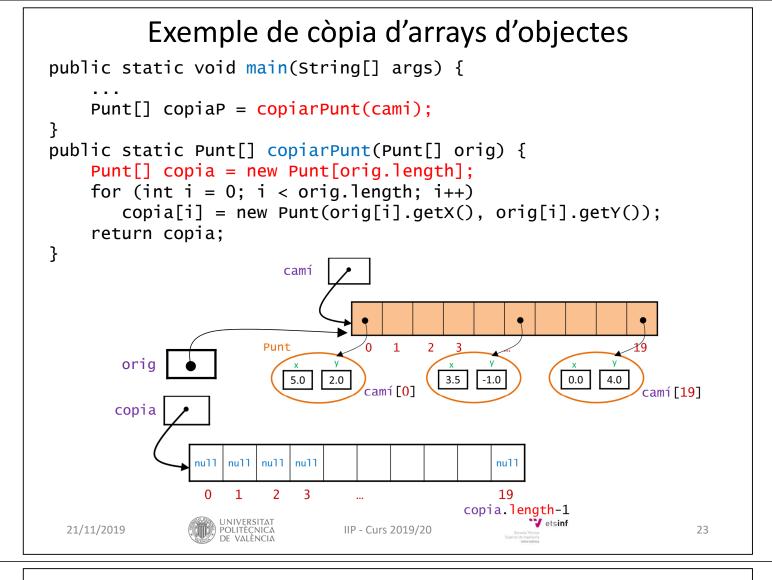


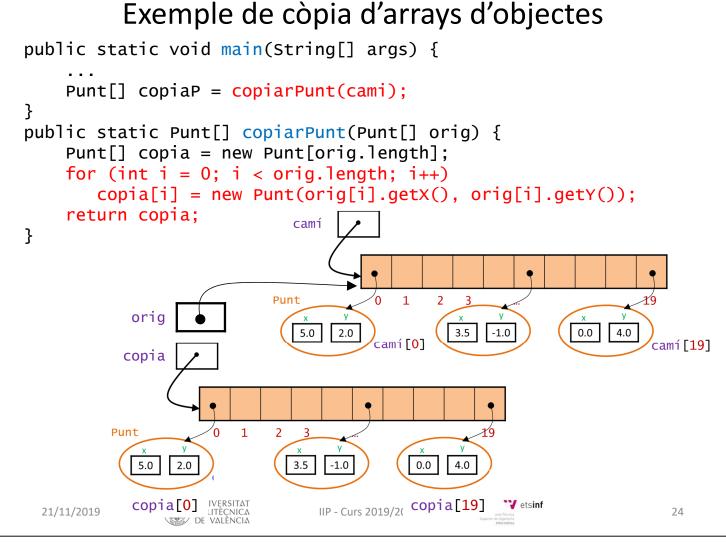
```
Exemple de còpia d'arrays d'objectes
CopiaArrayObjectes
Classe Edita Eines Opcions
Compila Desfés Retalla Copia Enganxa Cerca... Tanca
                                                         Implementació
  public class CopiaArrayObjectes {
      public static void main(String[] args) {
           final int NUM PUNTS = 20;
           Punt[] cami = new Punt[NUM PUNTS];
           for (int i = 0; i < cami.length; i++)
               cami[i] = new Punt();
           // s'han creat els 20 objects Punt de cami
           System.out.println("Distància entre els dos primers punts:
                               + cami[0].distancia(cami[1]));
           Punt[] copiaP = copiarPunt(cami);
      public static Punt[] copiarPunt(Punt[] orig) {
           Punt[] copia = new Punt[orig.length];
           for (int i = 0; i < orig.length; i++) {</pre>
               copia[i] = new Punt(orig[i].getX(), orig[i].getY());
           return copia;
```



# Exemple de còpia d'arrays d'objectes







# Exemple de còpia d'arrays d'objectes

```
public static void main(String[] args) {
     Punt[] copiaP = copiarPunt(cami);
}
                                       camí
                                    Punt
                                             2.0
                                                              3.5
                                                                                0.0
                                                                   -1.0
                                                                                      4.0
                                                   camí[0]
                                                                                           camí [19]
             copiaP
            Punt
                                       3.5
                                            -1.0
                5.0
                      2.0
                                                        0.0
                                                              4.0
               copiaP[0]
                                                        copiaP[19]
                        UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA
```

#### Exercicis de CAP

IIP - Curs 2019/20



25

- La clase Reloj: clau CCDJG4ai
- La clase TestReloj: clau CCDJH4ai



📆 etsinf

21/11/2019

# Representació de dades amb arrays

- Amb un array es pot representar una col·lecció de dades del mateix tipus. Per exemple, suposem que volem comptar la freqüència d'aparició de cadascuna de les cares d'un dau de 10 cares (numerades del 0 al 9).
  - Un conjunt de NUM\_COMPT comptadors:

int[] compt = new int[NUM\_COMPT];

Índex i, 0 <= i < compt.length	Component compt[i]
Valor o codi a comptar	Freqüència d'aparició d'i

- Representació de la freqüència d'aparició de cadascuna de les cares d'un dau
- Índex i : valor del dau
- Cada component compt[i]: freqüència d'aparició del valor i



# Representació de dades amb arrays

- Amb un array es pot representar una col·lecció de dades del mateix tipus. Per exemple:
  - Un conjunt de nombres naturals de qualsevol talla a l'interval [0, N]:

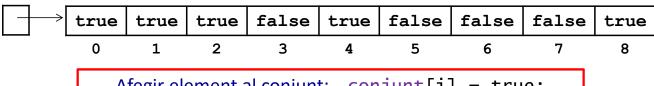
boolean[] conjunt = new boolean[N + 1];

Índex i, 0 <= i < conjunt.length	Component conjunt[i]
Valor natural	Pertany (true) o no (false) al conjunt

- Índex i: número natural
- Cada component conjunt[i]: true si el valor i pertany al conjunt i
  false en cas contrari.

0 1 2





Afegir element al conjunt: conjunt[i] = true; Eliminar element del conjunt: conjunt[i] = false;

• Aquesta representació permet conèixer de forma molt ràpida (temps independent del nombre d'elements) si un natural pertany o no al conjunt.

# Representació de dades amb arrays

```
BlueJ:exemples - Tema 7
 public class Conjunt {
     private boolean[] conjunt;
     private int ultim;
     /** Crea un conjunt buit que contindra
        naturals en el rang [0..ult], ult > 0 */
     public Conjunt(int ult) {
         conjunt = new boolean[ult + 1];
         ultim = ult;
     /** Comprova si un x donat pertany o no al conjunt,
      * 0 <= x <= ultim */
     public boolean pertany(int x) { return conjunt[x]; }
     /** Afegeix al conjunt un x donat,
      * 0 <= x <= ultim */
     public void afegeix(int x) { conjunt[x] = true; }
     /** Elimina del conjunt un x donat,
      * 0 <= x <= ultim */
     public void elimina(int x) { conjunt[x] = false; }
```

#### Representació de dades amb arrays

Exemple ls alumnes d'un grup de grandària ALUM\_A:

Estudiant[] grupA = new Estudiant[ALUM\_A];

Índex i, 0 <= i < grupB.length	Component grupB[i]
Ordre d'inserció	Informacio relativa a un alumne

- Índex i: l'ordre en la llista de l'estudiant, pot ser irrellevant.
- Cada component grupA[i]: dades de l'estudiant i-ésim.



```
grupA 0 1 2 3 ... ... ... ... ... ... ... ... ALUM_A - 1
```

```
public class Estudiant {
   private long dni;
   private double nota;
   private String nom;
   private boolean assistencia;
   ...
```

Afegir informació de l'objecte estudiant al grup:

```
grupA[i] = new Estudiant(...);
```



#### Representació de dades amb arrays public class Estudiant { private long dni; **? ! ! ! ! ! ! !** private double nota; private String nom; 9 9 8 7 9 9 9 9 private boolean assistencia; private char grup; 3 6 2 2 3 3 2 /\*\* Crea un objecte Estudiant, el nom i dni del \* qual s'introdueixen per teclat. \*/ public Estudiant(Scanner tec) { 9 9 System.out.println("Introdueix les dades"); System.out.print("Nom: "); nom = tec.nextLine(); System.out.print("Dni: "); dni = tec.nextLong(); tec.nextLine(); assistencia = true; BlueJ:exemples - Tema 7 nota = 0; public class GrupA { grup = ' '; public static final int ALUM\_A = 43; private Alumne[] grupA; /\*\* Crea un GrupB amb els alumnes del grup A, \* les dades dels quals es lligen des de teclat. \*/ public GrupA(Scanner tec) { grupA = new Alumne[ALUM\_A]; for (int i = 0; i < grupA.length; i++) { grupA[i] = new Alumne(tec); grupA[i].setGrup('A'); UNIVERSITAT 21/11/2019

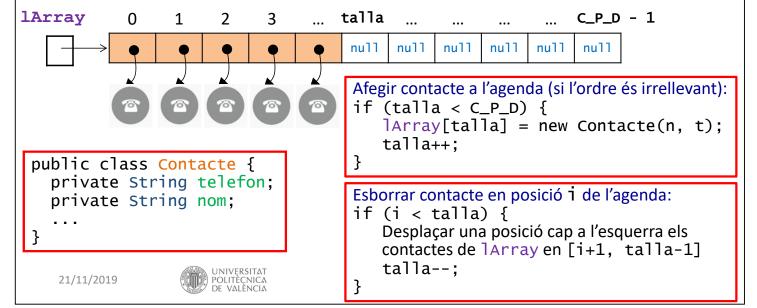
#### Representació de dades amb arrays

- Amb un array es pot representar una col·lecció de dades del mateix tipus. P.e.:
  - Agenda de contactes de grandària variable (màxim C\_P\_D):
     Contacte[] larray = new Contacte[C\_P\_D];

Índex i, 0 <= i < agenda.length</th>Component agenda[i]Ordre d'insercióInformacio relativa a un contacte



- Índex i: ordre en la llista del contacte, pot ser irrellevant.
- Cada component larray[i]: dades del contacte i-ésim.



# Representació de dades amb arrays

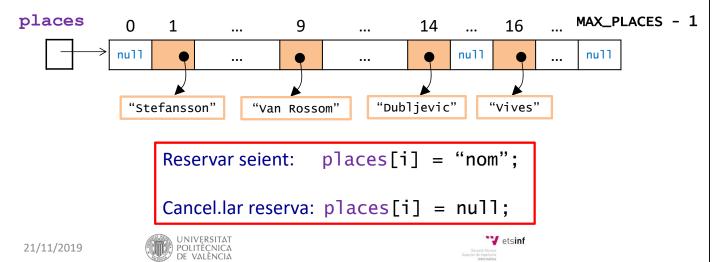
```
public class Contacte {
                                         private String telefon, nom;
                                          * Crea un Contacte a partir d'un telefon i un nom.
                                          * @param telefon String que representa un telefon.
                                          * @param nom String que representa un nom.
                                         public Contacte(String n, String t) {
BlueJ:AgendaSenseOrdre
                                            nom = n; telefon = t;
 public class Agenda {
      private static final int C P D = 250;
      private Contacte[] lArray;
      private int talla;
      /** Crea una Agenda buida, amb 0 contactes. */
      public Agenda() {
          lArray = new Contacte[C P D];
          talla = 0;
       * Torna la talla de l'agenda, es a dir, del numero de contactes.
       * @return int numero de contactes de l'agenda actual.
      public int getTalla() { return talla;
```

# Representació de dades amb arrays

- Amb un array es pot representar una col·lecció de dades del mateix tipus. Per exemple:
  - Els noms dels ocupants de les places d'un autobús de grandària MAX\_PLACES:
     String[] places = new String[MAX\_PLACES];

Índex i, 0 <= i < places.length	Component places[i]	A COUNTY OF THE PARTY OF THE PA
Número de seient	Nom del passatger	Briter

- Índex i: número de seient.
- Cada component places[i]: nom del passatger que ocupa el seient i.



# Representació de dades amb arrays

```
BlueJ:Autobus
public class Autobus {
     public static final int MAX PLACES = 50;
     private String[] places;
     /** Crea un Autobus buit */
     public Autobus() {
         places = new String[MAX PLACES];
     /** Reserva el seient i al passatger nom
         Precondició: 0 <= i < MAX PLACES i places[i] lliure
      */
     public void reservar(int i, String nom) { places[i] = nom;
     /** Cancel.la la reserva del seient i
         Precondició: 0 <= i < MAX PLACES i places[i] ocupat
      */
     public void cancelar(int i) { places[i] = null; }
                                               **/ etsinf
21/11/2019
```

# Tractament sequencial i directe d'un array

- En general es consideren dos tipus de tractament dels elements d'una estructura lineal, els anomenats tractament directe i següencial:
  - Directe: S'accedeix als elements per la localització, sense cap patró d'accés específic.
    - Exemples d'accés per posició: Problemes que usen l'estructura com un conjunt de comptadors o com a referències posicionals.
    - Exemples més complexos: Problemes que aprofiten l'accés directe dels arrays i les propietats d'ordenació dels seus elements com la cerca binària o els algorismes d'ordenació, etc. (Es veuran més endavant i en PRG)
  - Seqüencial: S'accedeix als elements de l'estructura (o d'una part d'ella) posicionalment, un darrere l'altre.
    - Exemples: Problemes de recorregut i cerca sequencial.
- Com els arrays són estructures d'accés directe, qualsevol dels dos tipus de tractament és possible.



#### Accés sequencial: recorregut i cerca

- Un recorregut es caracteritza per haver de visitar tots els elements de l'array per poder trobar la solució del problema.
  - Per contra, una cerca persegueix trobar el primer element que compleix una característica donada.
- El recorregut d'arrays s'usa per resoldre problemes que necessiten processar totes les dades per a poder determinar la solució.
- Exemples: Obtindre el màxim o el mínim d'un conjunt de nombres, obtenir la suma o el producte de tots els números d'un conjunt donat, obtenir la mitjana, etc.
- Es duen a terme mitjançant variables enteres que s'usen com a índexs per accedir a les seues diferents posicions.
- S'ha de portar el control de quines posicions ja s'han visitat i quantes s'han de visitar per poder resoldre el problema.



37

# Recorregut ascendent d'un array

des d'una posició inici fins a una posició fi 0≤inici≤fi<a.length

Recorregut iteratiu ascendent amb un bucle while:

```
int i = inici;
while (i <= fi) {
    tractar(a[i]); // Operacions amb l'element i-èsim
    avançar(i);
}</pre>
```

Recorregut iteratiu ascendent amb un bucle for:

```
for (int i = inici; i <= fi; avançar(i)) {
   tractar(a[i]); // Operacions amb l'element i-èsim
}</pre>
```

El mètode avançar representa l'increment de l'índex.

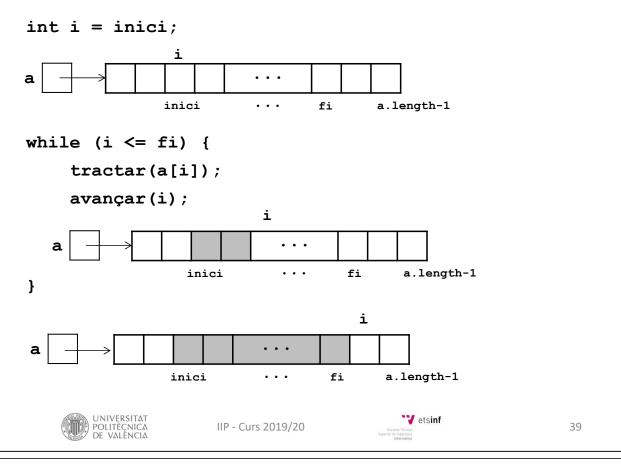




21/11/2019

#### Recorregut ascendent d'un array

des d'una posició inici fins a una posició fi 0≤inici≤fi<a.length



# Recorregut descendent d'un array

des d'una posició fi fins a una posició inici 0≤inici≤fi<a.length

Recorregut iteratiu descendent amb un bucle while:

```
int i = fi;
while (i >= inici) {
    tractar(a[i]); // Operacions amb l'element i-èsim
    retrocedir(i);
}
```

• Recorregut iteratiu descendent amb un bucle for:

```
for (int i = fi; i >= inici; retrocedir(i)) {
   tractar(a[i]); //Operacions amb l'element i-èsim
}
```

• El mètode retrocedir representa el decrement de l'índex.

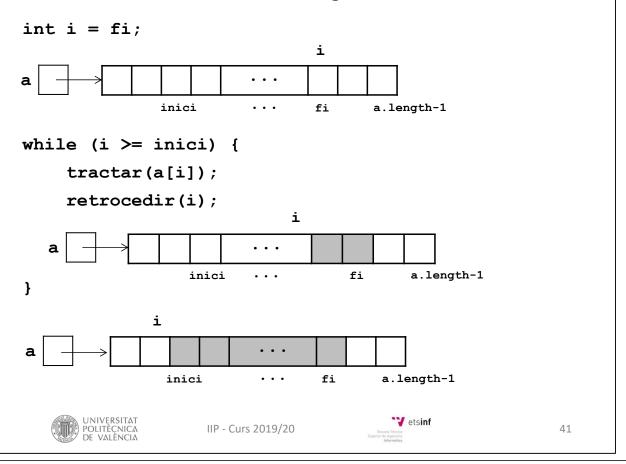




21/11/2019

# Recorregut descendent d'un array

des d'una posició fi fins a una posició inici 0≤inici≤fi<a.length



# Problemes de recorregut en arrays numèrics

21/11/2019

• Exemple de recorregut iteratiu ascendent: mètode que suma tots els elements d'un array d'enters (v.length>0).

```
public static int sumaIteAsc(int[] v) {
   int suma = 0;
   for (int i = 0; i < v.length; i++) {
      suma = suma + v[i];
   }
   return suma;
}

Isi volem calcular la mitjana dels elements de l'array?</pre>
```

```
public static double mitjanaIteAsc(int[] v) {
    double suma = 0;
    for (int i = 0; i < v.length; i++) {
        suma = suma + v[i];
    }
    return suma / v.length;</pre>
```

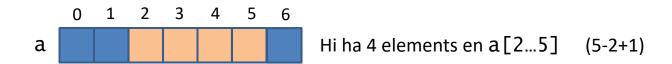
DE VALÈNCIA

#### Problemes de recorregut en arrays numèrics

• Exemple de processament de part d'un array: mètode que calcula la mitjana aritmètica del subarray d'enters a[esq...dre], 0<=esq<=dre<a.length

```
public static double mitjana(int[] a, int esq, int dre) {
    double suma = 0;
    for (int i = esq; i <= dre; i++) { suma += a[i]; }
    return suma / (dre - esq + 1);
}</pre>
```

• El nombre d'elements entre esq i dre (esq<=dre) és: dre - esq + 1



21/11/2019



IIP - Curs 2019/20



43

#### Problemes de recorregut en arrays numèrics

• Exemple de recorregut iteratiu descendent: mètode que suma tots els elements d'un array d'enters (v.length>0).

```
public static int sumaIteDesc(int[] v) {
   int suma = 0;
   for (int i = v.length - 1; i >= 0; i--) {
      suma = suma + v[i];
   }
   return suma;
}
```

```
0 1 2 3 4 5
V v.length = 6
```

• El bucle es deté quan i<0, és a dir, després de processar l'element que ocupa la posició 0 de l'array, quan i val -1.



#### Problemes de recorregut en arrays d'String

• Exemple de recorregut iteratiu ascendent: Calcula la posició del màxim a l'array a (a.length>0).

```
public static int maxim(String[] a) {
    int posMax = 0;
    for (int i = 1; i < a.length; i++) {
        if (a[i].compareTo(a[posMax]) > 0) {
            posMax = i;
        }
    }
    return posMax;
}
```

- Fixa't que el recorregut s'inicia en 1, ja que la posició 0 ja ha sigut tractada.
- Recorda que els tipus primitius es comparen amb ==,>,>=,<,<= mentre que els tipus objecte es comparen amb els mètodes equals() i/o compareTo().

21/11/2019



IIP - Curs 2019/20



45

#### Problemes de recorregut en arrays numèrics

• Exemple de recorregut iteratiu descendent: Calcula la posició del màxim a l'array a (a.length>0).

```
public static int maxim(double[] a) {
   int posMax = a.length-1;
   for (int i = a.length - 2; i >= 0; i--) {
      if (a[i] > a[posMax]) { posMax = i; }
   }
   return posMax;
}
```

 El recorregut s'inicia en a.length-2, ja que la posició a.length-1 ja ha sigut tractada.

# Problemes de recorregut en arrays numèrics

BlueJ:exemples - Tema 7

El mètode main de la classe Comptadors del projecte BlueJ exemples – Tema 7, simula el llançament d'un dau de 10 cares (numerades del 0 al 9) i obté la frequencia d'aparició de cadascun dels valors.



47

```
Classe Edita Eines Opcions

Compila Desfés Retalla Copia Enganxa Cerca... Tanca

public static void main(String[] args) {

int[] compt = new int[10];

int val;

for (int i = 0; i < 1000; i++) {

val = (int) (Math.random() * 10);

compt[val]++;

}

llistarFreq(compt);

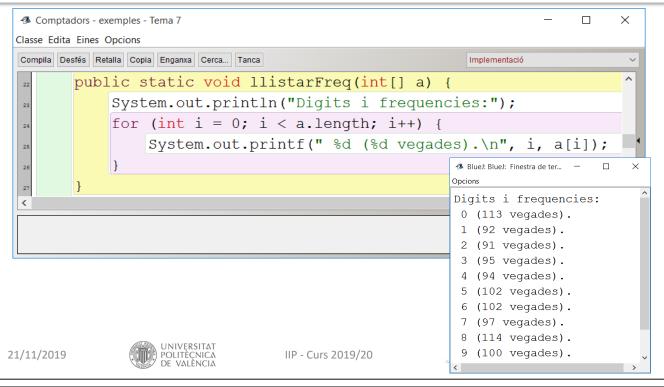
Classe compilada. No hi ha errors de sintaxi

desat
```

# Problemes de recorregut en arrays numèrics

BlueJ:exemples - Tema 7

El mètode llistarFreq de la classe Comptadors del projecte BlueJ exemples – Tema 7, té com a paràmetre un array d'enters a que ha de contenir les freqüències d'aparició dels índex i fa un recorregut ascendent d'aquest array mostrant per pantalla cada índex (i) i la seua freqüència d'aparició (a[i]).

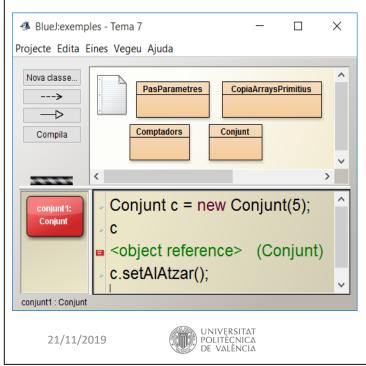


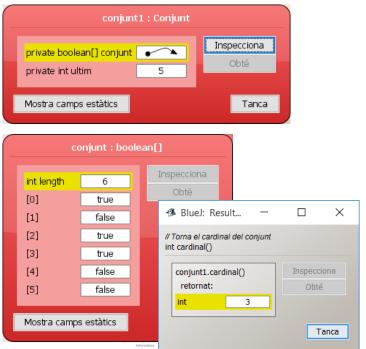
# Problemes de recorregut en arrays numèrics

BlueJ:exemples - Tema 7

El mètodes cardinal i setAlAtzar de la classe Conjunt del projecte BlueJ *exemples – Tema 7* són exemples de mètodes de recorregut. Fixa't en la seua implementacio i, després, prova'ls des del *Code Pad* i el *Object Bench* de BlueJ.

0 1 2





#### Problemes de recorregut en arrays numèrics

BlueJ:UtilsArrays

Completa en la classe Recorregut del projecte BlueJ *UtilsArrays*, el mètode minimplutja per tal que, a partir de les mesures diaries de pluviositat d'una setmana, mostre per pantalla quina és la mesura mínima i el dia de la setmana que es va produir.



recorregut iteratiu ascendent

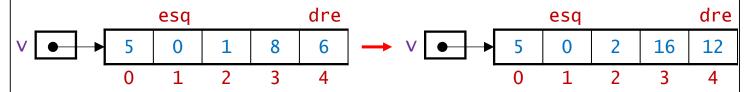
public static void minimPlutja(double[] plutja) {

}

# \_Problemes de recorregut en arrays numèrics

BlueJ:UtilsArrays

Completa en la classe Recorregut del projecte BlueJ *UtilsArrays*, el mètode duplicar per tal que donat un array d'enters v i dues posicions donades, esq i dre, de l'array, 0≤esq≤dre≤v.length-1, duplique el valor dels elements de l'array situats entre aquestes posicions.



```
public static void duplicar(int[] v, int esq, int dre) {
}
```

21/11/2019



IIP - Curs 2019/20

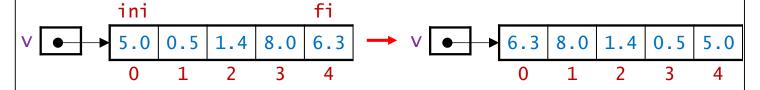


51

# Problemes de recorregut en arrays numèrics

BlueJ:UtilsArrays

Completa en la classe Recorregut del projecte BlueJ *UtilsArrays*, el mètode invertir per tal que, donat un array de reals v i dues posicions donades, ini i fi, de l'array, 0≤ini≤fi≤v.length-1, invertisca tots els elements de l'array situats entre aquestes posicions.

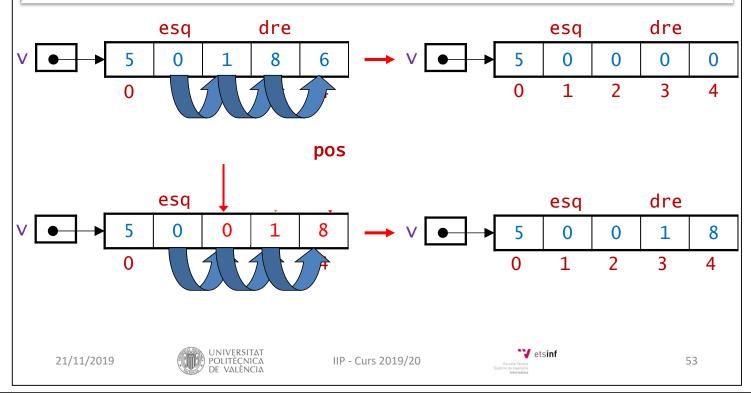


```
public static void invertir(double[] v, int ini, int fi) {
```

#### Problemes de recorregut en arrays numèrics

BlueJ:UtilsArrays

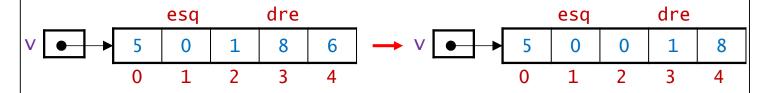
Completa en la classe Recorregut del projecte BlueJ *UtilsArrays*, el mètode desplasarDreta per tal que, donat un array d'enters v i dues posicions donades, esq i dre, de l'array, 0≤esq≤dre<v.length-1, desplace una posició cap a la dreta tots els elements de v compresos entre les posicions esq i dre, ambues inclusive.



# Problemes de recorregut en arrays numèrics

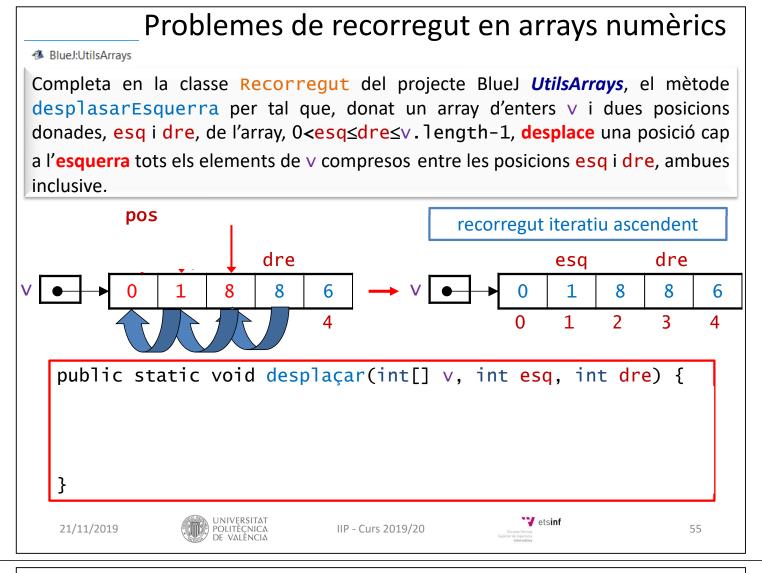
BlueJ:UtilsArrays

Completa en la classe Recorregut del projecte BlueJ *UtilsArrays*, el mètode desplasarDreta per tal que, donat un array d'enters v i dues posicions donades, esq i dre, de l'array, 0≤esq≤dre<v.length-1, desplace una posició cap a la dreta tots els elements de v compresos entre les posicions esq i dre, ambues inclusive.



recorregut iteratiu descendent

public static void desplacar(int[]\_v.\_int\_esq.\_int\_dre)\_{\_\_



#### Problemes de recorregut en arrays d'objectes

BlueJ:Autobus

 El mètode numeroReserves de la classe Autobus del projecte BlueJ Autobus torna el número de places ocupades en l'autobús comptant quantes posicions del seu array places són diferents de null.



```
public int numeroReserves() {
             int num = 0;
22
             for (int i = 0; i < places.length; <math>i++) {
23
                   if (places[i] != null) { num++; }
25
             return num;
26
                                                              Autobus bus = new Autobus();
                                                              bus.reservar(14, "Dubljevic");
27
<
                                                              bus.reservar(9, "Van Rossom");
                                                              bus.reservar(17, "Martinez");
                                                              bus.reservar(1, "Stefansson");
                                                              bus.reservar(16, "Vives");
                                                              int n = bus.numeroReserves();
                                                              5 (int)
                                                            <object reference> (Autobus)
                                          IIP - Curs 20
   21/11/2019
                                                  bus : Autobus
```

#### Problemes de recorregut en arrays d'objectes

BlueJ:Autobus

- El mètode nomMajor de la classe Autobus del projecte BlueJ Autobus torna el nom del passatger de l'autobús que tinga el "nom major" segons l'ordre lèxicografic.
- Coloca un punt de ruptura en la línia de l'if i observa l'estat de les variables i i res al llarg de l'execució del codi.



```
public String nomMajor() {
    String res = "";
    for (int i = 0; i < places.length; i++) {
        if (places[i] != null && places[i].compareTo(res) >= 0) {
            res = places[i];
        }
    }
    return res;
}
```

21/11/2019



IIP - Curs 2019/20



57

#### Problemes de recorregut en arrays d'objectes

BlueJ:Autobus

- Completa en la classe Autobus del projecte BlueJ Autobus els següents mètodes:
  - 1listaPassatgers que torna un String amb els noms dels passatgers i les places que ocupen.
  - llistaSeientsLliures que torna un String amb els números de seients que no estan ocupats.



- BlueJ:AgendaSenseOrdre
- Completa en la classe Agenda del projecte BlueJ AgendaSenseOrdre el mètode toString que torna un String amb la informació de tots els contactes de l'agenda o, en cas que no hi hagen contactes a l'agenda, torna "Agenda buida".

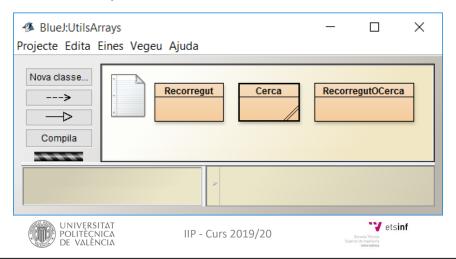


UNIVERSITAT POLITÈCNICA DE VALÈNCIA



#### Accés sequencial: recorregut i cerca

- Una cerca requereix determinar si hi ha algun element de l'array que compleix una certa propietat.
- Aquest tipus de problemes impliquen operacions que poden obtenir la solució sense necessitat de conèixer totes les dades: trobar el primer que compleix cert requisit, tractar totes les dades fins que es compleixca certa condició, etc.
  - Les cerques a un array requereixen l'ús de variables de tipus int per accedir a les seues diferents posicions, de manera semblant als recorreguts, i de variables de tipus boolean per a comprovar la condició de terminació.
  - S'ha de portar el control de quines posicions ja s'han visitat i quines s'han de visitar per poder resoldre el problema.



Problemes de cerca en arrays

- S'estableixen dues estratègies principals a l'hora de buscar:
  - Cerca lineal: es va reduint l'espai de cerca (quantitat d'informació sobre la que buscar) element a element. Es pot aplicar sempre independentment de si les dades estan ordenades o no a l'array.
  - Cerca binària o dicotòmica: es va reduint l'espai de cerca, eliminant cada vegada la meitat d'elements. Aquesta estratègia necessita que les dades dintre de l'array estiguen ordenades d'una forma coneguda (per exemple, de menor a major), però és més eficient que les cerques lineals.





21/11/2019

#### Cerca ascendent en un array

• Estructura de cerca ascendent usant una variable lògica: Existeix algun element a [i] que compleix la propietat?

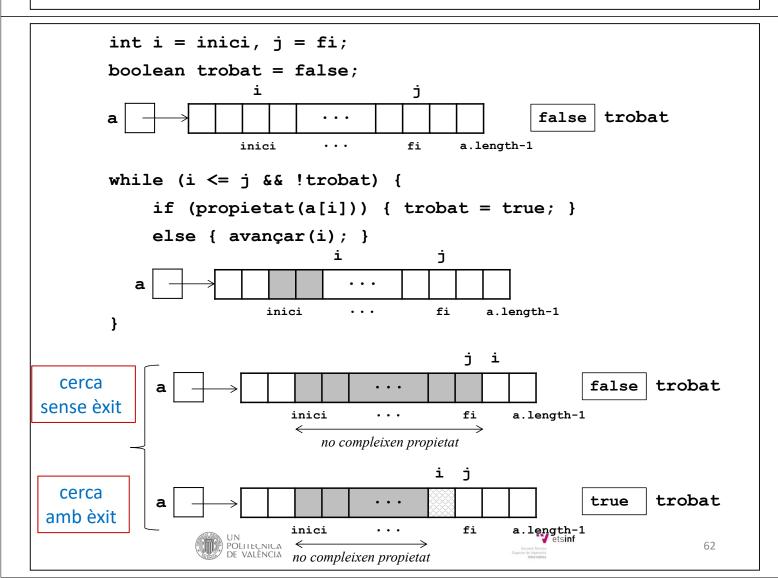
```
int i = inici, j = fi;
boolean trobat = false;
while (i <= j && !trobat) {
    if (propietat(a[i])) { trobat = true; }
    else { avançar(i); }
}
// Resoldre la cerca
if (trobat) ... // a[i] compleix la propietat
else ... // cap element compleix la propietat</pre>
```

21/11/2019



IIP - Curs 2019/20





# Cerca descendent en un array

• Estructura de cerca descendent usant una variable lògica: Existeix algun element a [i] que compleixca la propietat?

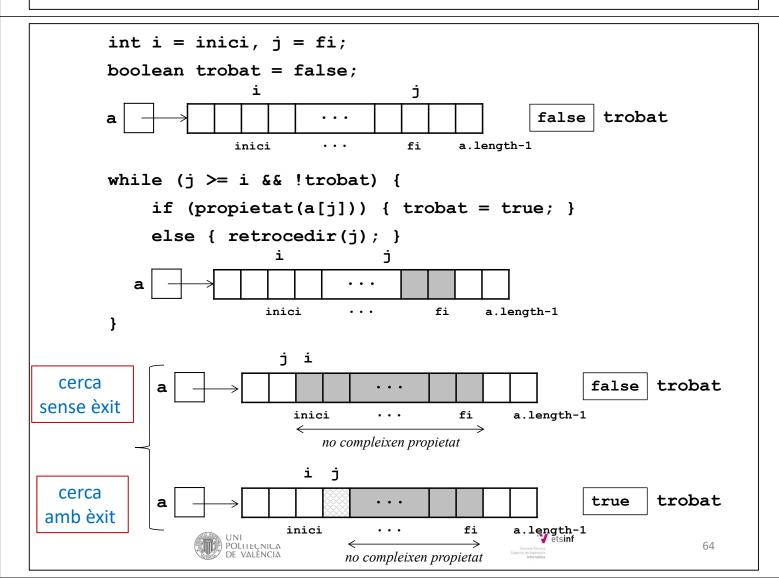
```
int i = inici, j = fi;
boolean trobat = false;
while (j >= i && !trobat) {
   if (propietat(a[j])) { trobat = true; }
   else { retrocedir(j); }
}
// Resoldre la cerca
if (trobat) ... // a[j] compleix la propietat
else ... // cap element compleix la propietat
```

21/11/2019



IIP - Curs 2019/20





# Estratègies de cerca en arrays

• Estructura de cerca lineal iterativa (sense garantia d'èxit) amb guarda que avalua la propietat. Algun element a[i] compleix la propietat?

```
int i = inici, j = fi;
while (i <= j && !propietat(a[i])) {
    avançar(i);
}
// El bucle acaba perquè:
// i <= fi → a[i] compleix la propietat o
// i val fi + 1

// Resoldre la cerca
if (i <= fi) ... // Es compleix propietat(a[i])
else ... // Cap element compleix propietat(a[i])</pre>
```

21/11/2019



IIP - Curs 2019/20



65

# Estratègies de cerca en arrays

• Estructura de cerca ascendent amb garantia d'èxit: es coneix que hi ha algun element a[i] que compleix la propietat.

```
int i = inici;
while (!propietat(a[i])) {
    avançar(i);
}
// Resoldre la cerca: En la posició i hi ha
// un element que compleix la propietat
```

• És possible incloure deliberadament un element que compleixi la propietat. Aquest element s'anomena sentinella. En aquest cas, la cerca es diu cerca amb sentinella.





#### Problemes de cerca en arrays

• Cerca ascendent d'un element en un array de String (a.length>0) sense garantia d'èxit. Si no el troba, torna -1.

```
public static int cercarPos(String[] a, String dada) {
   int i = 0;
   while (i < a.length && !a[i].equals(dada)) { i++; }
   if (i < a.length) { return i; }
   else { return -1; }
}</pre>
```

• Cerca descendent d'un element en un array de double (a.length>0) sense garantia d'èxit. Si no el troba, torna -1.

```
public static int cercarPos(double[] a, double dada) {
   int i = a.length - 1;
   while (i >= 0 && a[i] != dada) { i--; }
   return i;
}
```

21/11/2019



IIP - Curs 2019/20



67

#### Problemes de cerca en arrays

• Cerca ascendent d'un element en un array de String (a.length>0) amb garantia d'èxit.

```
public static int cercarPosEsta(String[] a, String dada) {
   int i = 0;
   while (!a[i].equals(dada)) { i++; }
   return i;
}
```

 Cerca descendent d'un element en un array de double (a.length>0) amb garantia d'èxit.

```
public static int cercarPosEsta(double[] a, double dada) {
   int i = a.length - 1;
   while (a[i] != dada) { i--; }
   return i;
}
```

#### Problemes de cerca en arrays

• Comprova si hi ha algun element de l'array (a.length>0) més gran que dada.

```
public static boolean hiHaMajor(double[] a, double dada) {
   int i = 0;
   while (i < a.length && a[i] <= dada) { i++; }
   return (i < a.length);
}</pre>
```

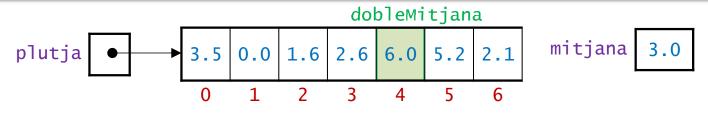
 Retorna, si existeix, la posició del primer element de l'array (a.length>0) més gran que la suma dels anteriors. Si no, torna -1.

```
public static int cercaPosMajorQueSuma(int[] a) {
   int i = 1, suma = a[0];
   while (i < a.length && a[i] <= suma) { suma += a[i]; i++; }
   if (i < a.length) { return i; }
   else { return -1; }
}</pre>
```

BlueJ:UtilsArrays

#### Problemes de cerca en arrays numèrics

Completa en la classe Cerca del projecte BlueJ *UtilsArrays*, el mètode dobleMitjana per tal que, a partir de les mesures diàries de pluviositat d'una setmana i de la mitjana d'aquestes mesures, indique si algun dia ha plogut el doble de la mitjana.



cerca linial iterativa ascendent amb variable boolean

```
public static boolean dobleMitjana(double[] plutja, double mitjana) {
```



# Problemes de cerca en arrays numèrics

Completa en la classe Cerca del projecte BlueJ *UtilsArrays*, el mètode dobleMitjana per tal que, a partir de les mesures diàries de pluviositat d'una setmana i de la mitjana d'aquestes mesures, indique si algun dia ha plogut el doble de la mitjana.



public static boolean dobleMitjana(double[] plutja, double mitjana) {

}

21/11/2019



IIP - Curs 2019/20



71

#### BlueJ:UtilsArrays

# Problemes de cerca en arrays numèrics

Completa en la classe Cerca del projecte BlueJ *UtilsArrays*, el mètode ordenAsc per tal que, a partir de les mesures diàries de pluviositat d'una setmana a l'array plutja, indique si l'array està ordenat ascendentment.

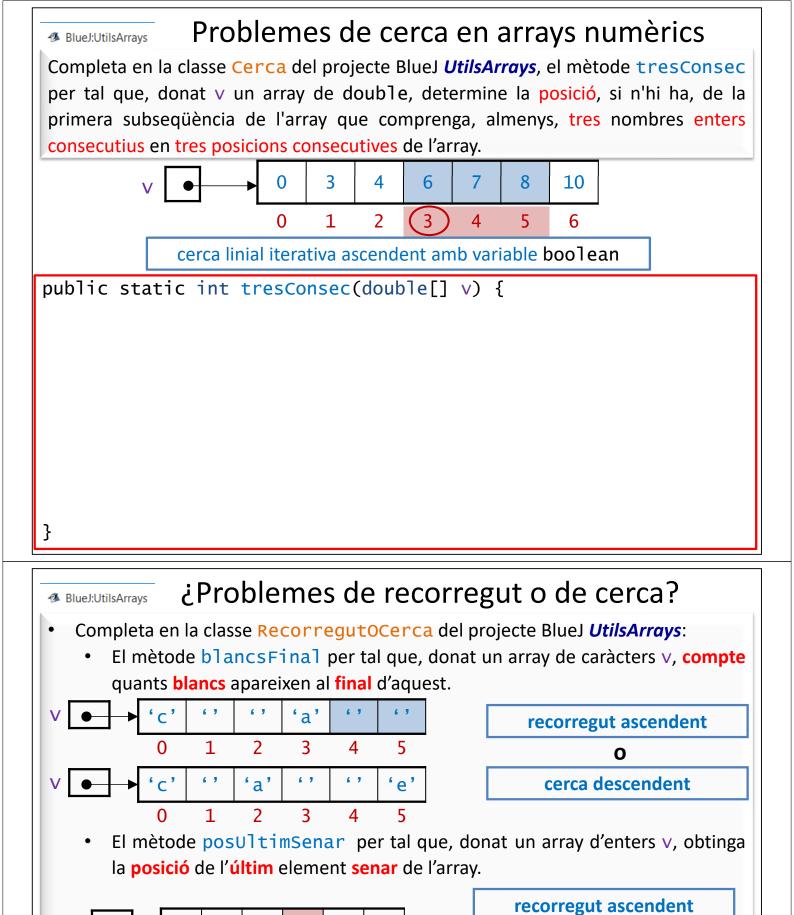


public static boolean ordenAsc(double[] plutja) {

}

```
Problemes de cerca en arrays
public static boolean ordenAsc(double[] plutja) {
  int i = 1;
  while (plutja[i-1]<=plutja[i] && i<plutja.length) { i++; }</pre>
  return i == plutja.length;
}
                                                      És correcta?
                                                                   NO
public static boolean ordenAsc(double[] plutja) {
  int i = 1;
  while (i<plutja.length & plutja[i-1]<=plutja[i]) { i++; }</pre>
  return i == plutja.length;
}
                                                      És correcta?
                                                                   NO
public static boolean ordenAsc(double[] plutja) {
  int i = 1;
  while (i<plutja.length && plutja[i-1]<=plutja[i]) { i++; }</pre>
  if (plutja[i-1] > plutja[i]) { return false; }
  else { return true; }
}
                                                        És correcta? NO
  Si l'array està ordenat ascendentment, en els tres casos, es produeix l'excepció
        en execució: java.lang.ArrayIndexOutOfBoundsException
```

### Problemes de cerca en arrays numèrics BlueJ:UtilsArrays Completa en la classe Cerca del projecte BlueJ UtilsArrays, el mètode esCapicua per tal que, a partir de les mesures diàries de pluviositat d'una setmana a l'array plutja, indique si l'array és capicua. plutja 1.6 2.1 5.5 0.0 2.1 1.6 0.0 true 3 0 1 2 4 6 plutja 0.0 1.6 false 3.4 5.5 2.1 1.6 0.0 5 0 1 2 3 6 cerca linial iterativa ascendent amb guarda que avalua la propietat public static boolean esCapicua(double[] plutja) {



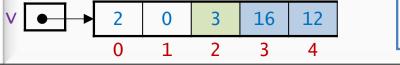
2 5 0 1 16 12

0 1 2 3 4 5

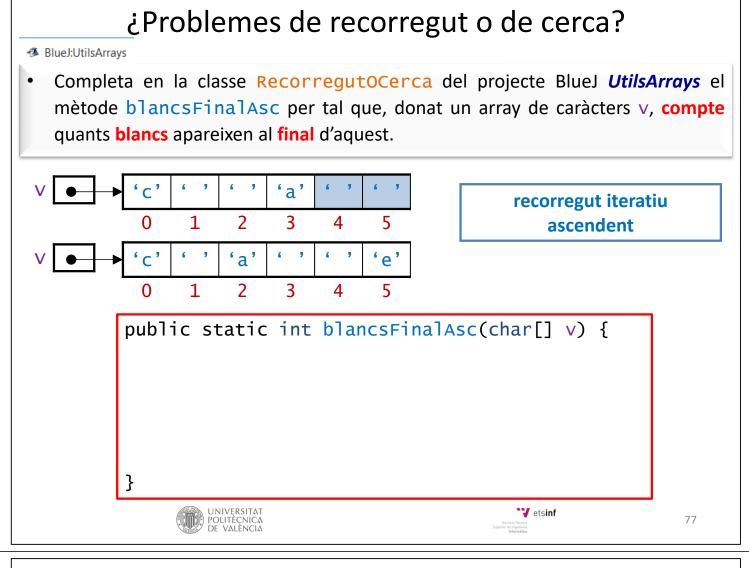
cerca descendent

• El mètode sumaDespresPrimerSenar per tal que, donat un array d'enters

positius v, sume els elements que apareixen després del primer senar.

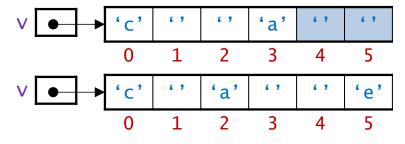


cerca + recorregut ascendent



## ¿Problemes de recorregut o de cerca?

- BlueJ:UtilsArrays
- Completa en la classe RecorregutOCerca del projecte BlueJ UtilsArrays el mètode blancsFinalDesc per tal que, donat un array de caràcters v, compte quants blancs apareixen al final d'aquest.



cerca linial iterativa descendent

public static int blancsFinalDesc(char[] v) {

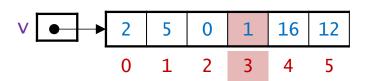
}

| Polity Rallari
| De Vive Rallari

78



- BlueJ:UtilsArrays
- Completa en la classe RecorregutOCerca del projecte BlueJ UtilsArrays el mètode posUltimSenarAsc per tal que, donat un array d'enters v, obtinga la posició de l'últim element senar de l'array.



recorregut iteratiu ascendent

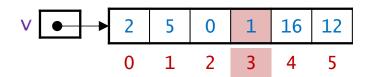
• Si posultSenar >= 0 la posició de l'últim senar és posultSenar sino no hi ha senars a l'array i posultSenar és -1

DE VALÈNCIA

79

## ¿Problemes de recorregut o de cerca?

- BlueJ:UtilsArrays
- Completa en la classe RecorregutOCerca del projecte BlueJ *UtilsArrays* el mètode posUltimSenarDesc per tal que, donat un array d'enters v, obtinga la posició de l'últim element senar de l'array.



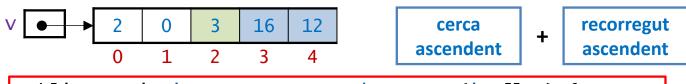
cerca linial iterativa descendent

```
public static int posUltimSenarDesc(int[] v) {
}
```

• Si i >= 0 la posició de l'últim senar és i sino no hi ha senars a l'array i i és -1

## ¿Problemes de recorregut o de cerca?

 Completa en la classe RecorregutOCerca del projecte BlueJ UtilsArrays el mètode sumaDespresPrimerSenar per tal que, donat un array d'enters positius v, sume els elements que apareixen després del primer senar.



```
public static int sumaDespresPrimerSenar(int[] v) {
```

# Problemes de cerca en arrays d'objectes

BlueJ:Autobus

}

 El mètode estaComplet de la classe Autobus del projecte BlueJ Autobus torna true si l'autobus està complet (no queden places Iliures) i, en cas contrari, torna false.



```
public boolean estaComplet() {
    int i = 0;
    while (i < places.length && places[i] != null) { i++; }
    return i == places.length;
}</pre>
```



## Problemes de cerca en arrays d'objectes

- BlueJ:AgendaSenseOrdre
- Completa en la classe Agenda del projecte BlueJ AgendaSenseOrdre el mètodes:
  - cercar que cerca un contacte a l'agenda donat un nom. Torna la posició de l'element si el troba o -1 si no el troba.
  - inserir que afegeix un nou contacte vàlid a l'agenda o l'actualitza si ja existeix. Torna true si s'ha afegit amb èxit o false en cas que l'agenda estiga plena.
  - eliminar que donat un nom vàlid, elimina el Contacte de l'agenda amb eixe nom. Torna true si s'ha pogut eliminar i false si no s'ha pogut eliminar per que no existeix un contacte amb eixe nom a l'agenda.



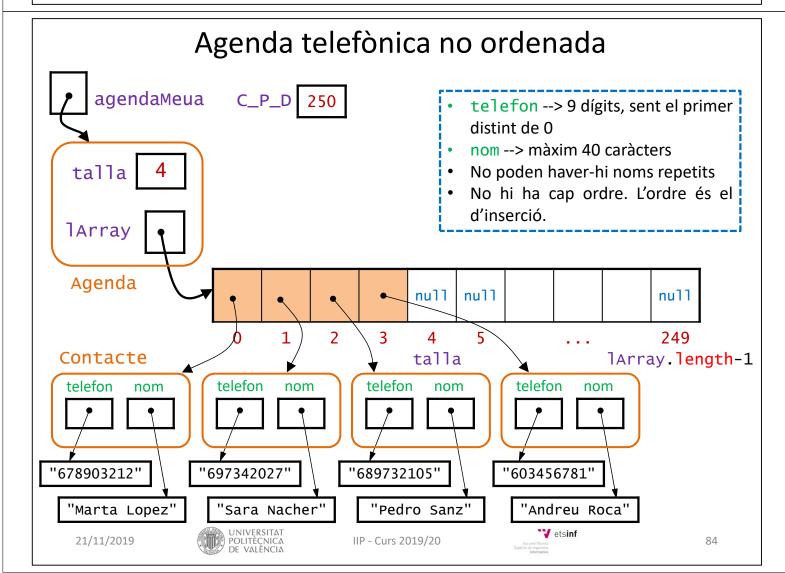
21/11/2019

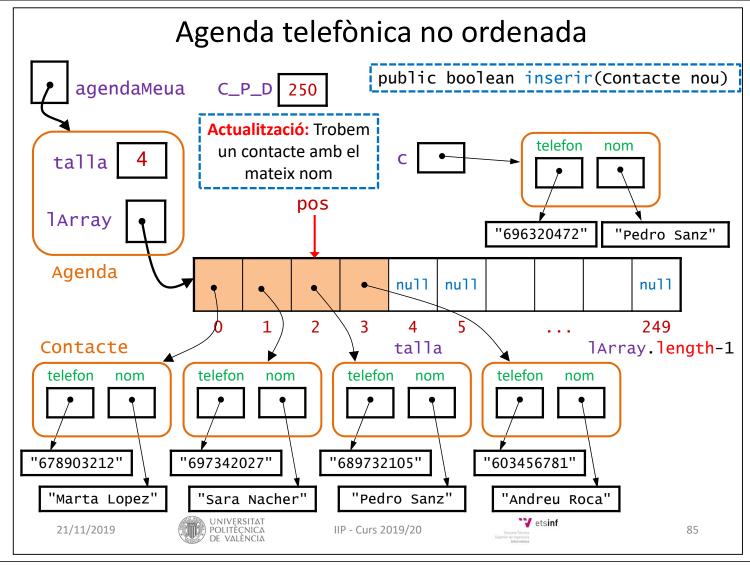


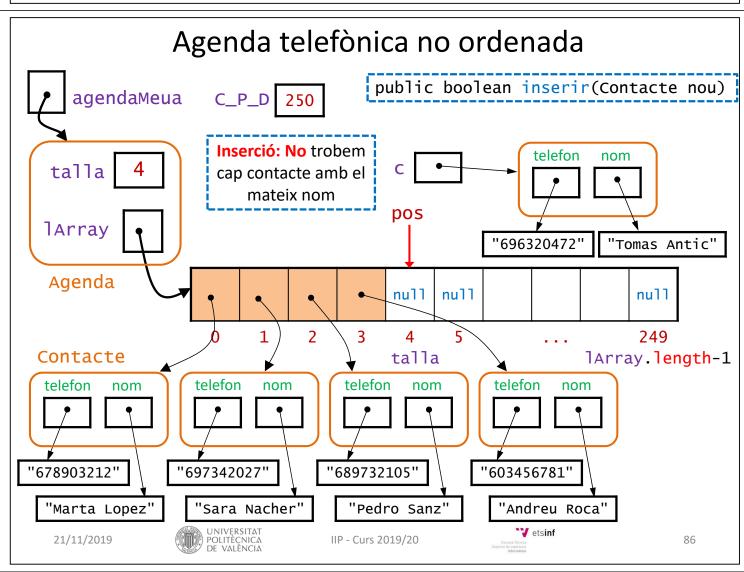
IIP - Curs 2019/20



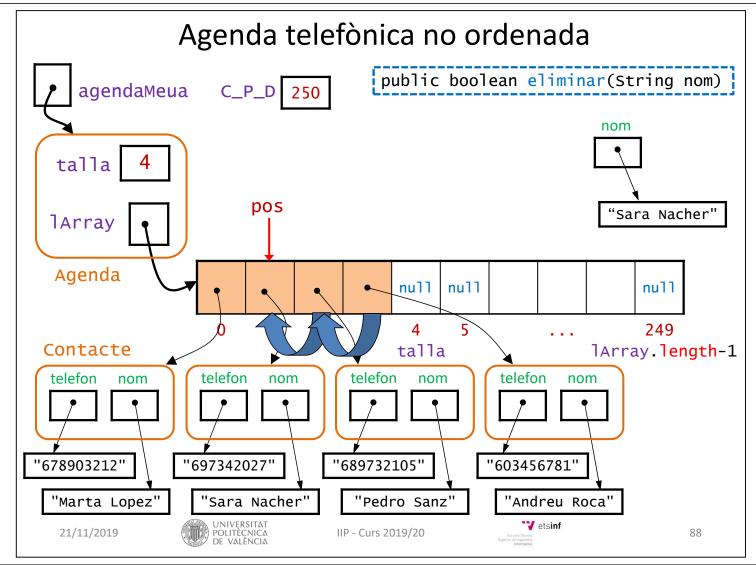
83



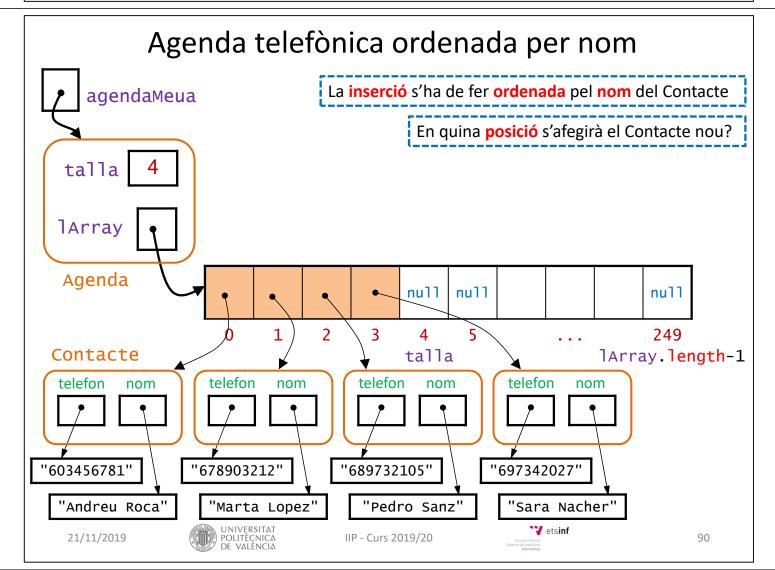


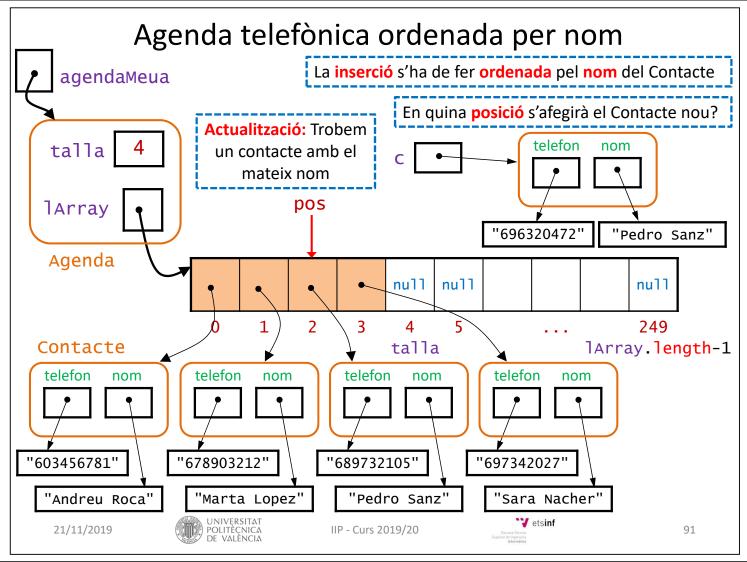


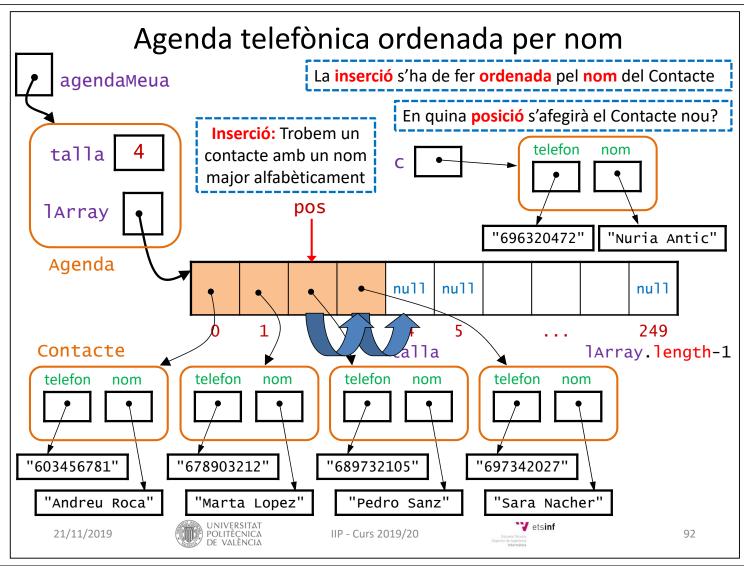
```
/** Cerca un contacte a l'agenda donat un nom. Torna la posició
    de l'element si el troba o -1 si no el troba. */
private int cercar(String nom) {
    int i = 0;
    while (i < talla && !larray[i].getNom().equals(nom)) { i++; }</pre>
    if (i < talla) { return i; } else { return -1; }
}
/** Afegeix un nou contacte vàlid a l'agenda o l'actualitza si ja
    existeix. Torna true si s'ha afegit amb èxit o false en cas de
    que l'agenda estiga plena. */
public boolean inserir(Contacte nou) {
    boolean cap = true;
    // cerca del nou contacte: si existeix, el reemplaça pel nou, l'actualitza;
// sino, si i només si cap, insereix el nou contacte darrere de l'últim
    // existent, és a dir, en posició talla.
    int pos = cercar(nou.getNom());
    if (pos != -1) {
         larray[pos] = nou;
    }
    else if (talla < C_P_D) {
         larray[talla++] = nou;
    else { cap = false; }
    return cap;
}
```

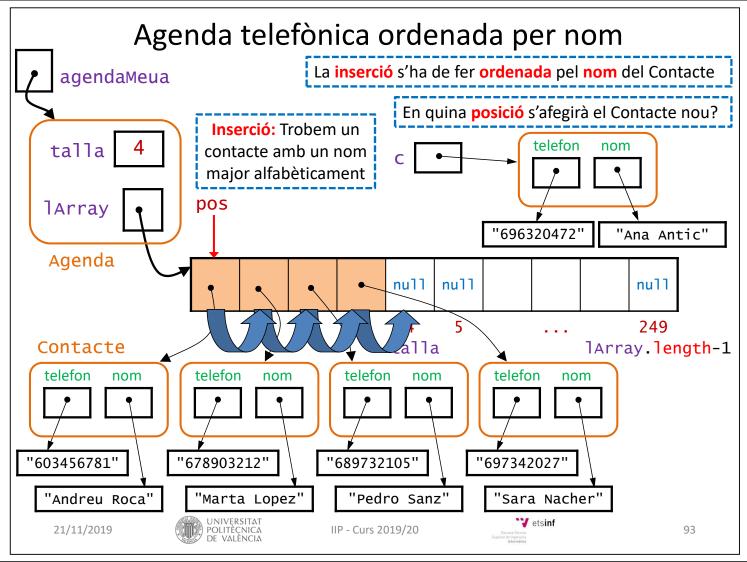


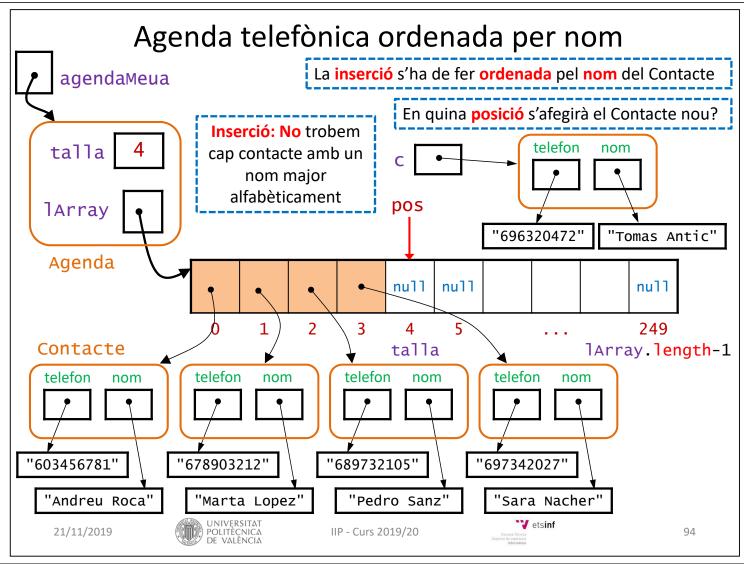
```
Desplaça una posició cap a l'esquerra tots els elements de l'Array
    compresos entre les posicions ini i fi, ambdues incloses,
    0<ini<=fi<=lArray.length-1. */</pre>
private void desplacarEsq(int ini, int fi) {
 for (int pos = ini-1; pos < fi; pos++) { |Array[pos] = |Array[pos+1]; }
}
    Donat un nom vàlid, elimina el Contacte de l'agenda amb eixe nom.
    Torna true si s'ha pogut eliminar i false si no s'ha pogut eliminar
    per que no existeix un contacte amb eixe nom a l'agenda.*/
public boolean eliminar(String nom) {
  boolean esta = true:
  // cerca, per nom, el contacte que es vol eliminar: si existeix, l'elimina:
  // desplaça una posició cap a l'esquerra tots els contactes posteriors al
  // que es va a eliminar, actualitza la talla, i torna true com resultat.
  // Si no troba el contacte a eliminar, torna false per a indicar-ho.
  int pos = cercar(nom);
  if (pos != -1) {
      desplacarEsq(pos+1, talla-1);
      talla--;
  else { esta = false; }
  return esta;
}
```







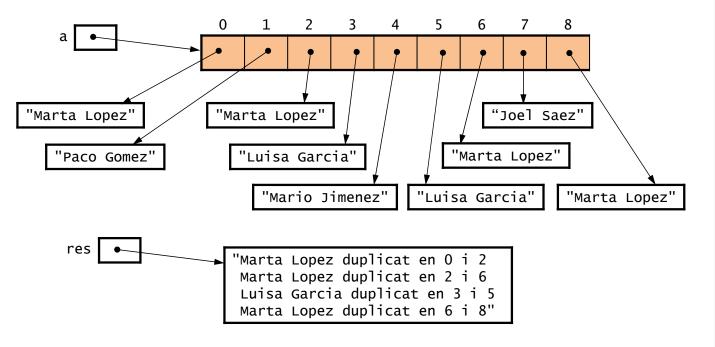




```
.a inserció s'ha de fer ordenada pel nom del Contacte
Com hauriem de modificar cercar i inserir (de l'Agenda no ordenada)?
private int cercar(String nom) {
    int i = 0:
    while (i < talla && !larray[i].getNom().equals(nom))</pre>
    if (i < talla) { return i; }
                                                       larray[i].getNom()
    else { return -1; }
                                                   menor alfabèticament que nom
                                    return i;
public boolean inserir(Contacte nou) {
  boolean cap = true;
  int pos = cercar(nou.getNom());
 // cerca del nou contacte: si existeix, el reemplaça pel nou, l'actualitza;
 // sino, si i només si cap, insereix el nou contacte darrere de l'últim
                                                      en posició pos.
 // existent, és a dir, en posició talla.
  if (pos!=-1) { (pos < talla && lArray[pos].getNom().equals(nou.getNom()))
      larray[pos] = nou;
                                      desplaçarDre(pos,talla-1);
                                      larray[pos] = nou;
  else if (talla < C_P_D) {
                                      talla++;
      larray[talla++] = nou;
  } else { cap = false; }
                              private void desplaçarDre(int ini, int fi) {
                                 for (int pos = fi+1; pos > ini; pos--) {
  return cap;
                                       |Array[pos]| = |Array[pos-1]|;
}
                 DE VALENCIA
```

## Esquemes combinats de cerca i recorregut

- Hi ha problemes que requereixen combinar ambdues estratègies:
  - Donat un array de String, determinar per a cada String de l'array la primera repetició. El resultat ha de ser un String en el qual en cada línia estiga la dada i les dues posicions en les quals apareix.



## Esquemes combinats de cerca i recorregut

- BlueJ:UtilsArrays
- Fixa't en el mètode duplicatsLlista de la classe RecorregutOCerca del projecte BlueJ *UtilsArrays* que, donat un array de String, determina per a cada String de l'array la primera repetició.
- Executa el main que prova el mètode duplicatsLlista.
  - Recorregut de l'array en què per a cada element es realitza, a la vegada, la cerca d'una component igual a ell des de la seua posició en endavant.

```
public static String duplicatsLlista(String[] a) {
    String res = "";
    for (int i = 0; i < a.length - 1; i++) {
        int j = i + 1;
        while (j < a.length && !a[i].equals(a[j])) { j++; }
        if (j < a.length) {
            res += a[i] + " duplicat en: " + i + " i " + j + "\n";
        }
    }
    return res;
}

21/11/2019
</pre>

UNIVERSITAT DOLITECNICA DE VALENCIA
DE VALENCIA

IIP-Curs 2019/20
```

# Més problemes d'accés directe

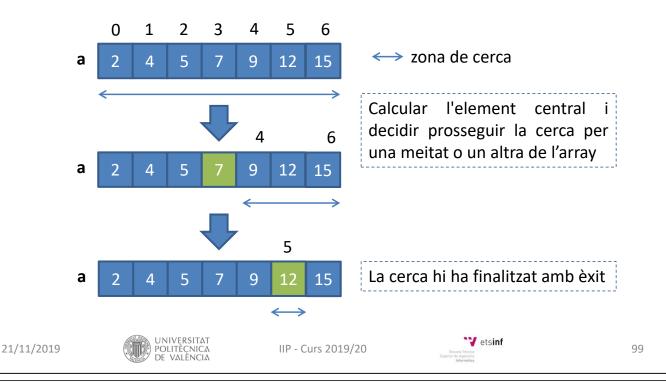
- L'array permet l'accés en temps constant a qualsevol component, coneguda la seua posició.
  - En l'array a [0..999], a [0] té el mateix cost temporal que a [999].
- Això permet implementar alguns algorismes de forma molt eficient. Exemples:
  - Cerca binària o dicotòmica
    - Permet la cerca eficient sobre un array ordenat.
  - Càlcul de la moda d'un multiconjunt de naturals
    - El membre del multiconjunt que més vegades es repeteix.





# Cerca binària: estratègia

- Només aplicable a arrays ordenats (ascendentment en este cas).
- Exemple estratègia: cercar l'element amb valor 12



#### BlueJ:UtilsArrays

## Cerca binària: implementació

- El mètode cercaBin de la classe Cerca del projecte BlueJ UtilsArrays, aplica l'estratègia de la cerca binària per cercar un enter x en un array d'enters a ordenat ascendentment entre dues posicions inici i fi de l'array (0<=inici<=fi<a.length).</li>
- Fica un punt de ruptura en la línia del while i observa l'estat de les variables al llarg de l'execució del codi per a l'array a = {0, 3, 4, 6, 7, 8, 10, 17}, inici = 0, fi = 7 i x = 20. Quantes iteracions es fan?

```
public static int cercaBin(int[] a, int inici, int fi, int x) {
       int esq = inici, dre = fi, meitat = 0;
       boolean trobat = false;
       while (esq <= dre && !trobat) {
           meitat = (esq + dre) / 2;
           if (x == a[meitat]) { trobat = true; }
           else if (x < a[meitat]) { dre = meitat - 1; }</pre>
           else { esq = meitat + 1; }
       if (trobat) { return meitat; }
                                                  ordenat creixentment
       else { return -1; }
                                                      meitat
                                                                  dre
                                                                        a.length-1
                                                            majors que
                                              menors que
21/11/2019
                                              a[meitat]
                                                            a[meitat]
```

```
n = a.length
                     3
                                             8
                                                        17
                                                                                            20
                           4
                                 6
                                       7
                                                  10
a
                                                                       int esq = 0,
                                                        dre 1ª iter.
                           2
                                 3
                                             5
                     1
                                       4
                                                   6
             esq
                                                                            dre = a.length-1,
          meitat
                                                                            meitat = 0;
                                                                       boolean trobat = false:
                     3
                           4
                                       7
                                             8
                                                  10
                                                                 n/2
                                 6
                                                        17
a
                                                                       while(esq<=dre && !trobat)</pre>
                                                                        { meitat = (esq+dre)/2;
               0
                     1
                           2
                                 3
                                             5
                                                   6
                                                        dre
                                     esa
                                                                          if (x==a[meitat])
                                                             2ª iter.
                             meitat
                                                                               trobat = true:
                                                                          else if (x<a[meitat])</pre>
                                                                                 dre = meitat-1;
                                                                 n/2<sup>2</sup>
                                 6
                                             8
               0
                     3
                           4
                                       7
                                                  10
                                                        17
a
                                                                          else esq = meitat+1;
                                                                       }
                     1
                           2
                                 3
                                       4
                                             5
               0
                                                 esa dre
                                                             3ª iter.
                                         meitat
                                                                       if (trobat)
                                                                              return meitat;
                                                                 n/2<sup>3</sup> else return -1;
                     3
                           4
                                       7
                                             8
                                                  10
               0
                                 6
a
                     1
                           2
                                 3
                                             5
               0
                                       4
                                                   6
                                                         esa
                                                                               n/2^{1imit} = 1
                                              meitat<sub>dre</sub>4ª iter
                     3
                           4
                                       7
                                             8
                                                  10
                                                        17
a
                                 6
                                                                              límit = log<mark>,</mark>n
                                                        dre esq
                           2
                                 3
                                             5
               0
                     1
                                       4
                                                   6
                                                                        log<sub>2</sub>n+1 iteracions
                                                      meitat
                           UNIVERSITAT
                                                                       etsinf
     21/11/2019
                                            IIP - Curs 2019/20
                                                                                              101
```

```
La inserció s'ha de fer ordenada pel nom del Contacte
         Hauriem de modificar recuperar i eliminar (de l'Agenda no ordenada)?
public String recuperar(String nom) {
    String telefon = null:
    // cerca, per nom, el contacte del qual es vol recuperar el telèfon:
    // si existeix, el torna com resultat i si no existeix torna null
    int pos = cercar(nom);
                                    cercaBin(0, talla - 1, nom);
    if (pos != -1) { telefon = lArray[pos].getTelefon(); }
    return telefon;
}
public boolean eliminar(String nom) {
   boolean esta = true:
   // cerca, per nom, el contacte que es vol eliminar: si existeix, l'elimina:
   // desplaça una posició cap a l'esquerra tots els contactes posteriors al
   // que es va a eliminar, actualitza la talla, i torna true com resultat.
   // Si no troba el contacte a eliminar, torna false per a indicar-ho.
   int pos = cercar(nom);
                                    cercaBin(0, talla - 1, nom);
   if (pos != -1) {
       desplacarEsq(pos + 1, talla - 1);
        talla--;
   else { esta = false; }
   return esta;
}
```



## Exercicis de CAP



Agenda con contactos en orden alfabético: clau CCDJI4ai

21/11/2019



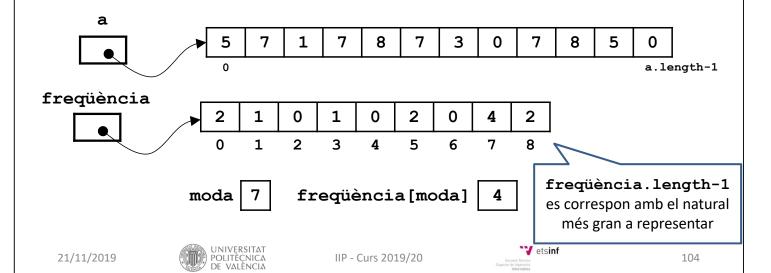
IIP - Curs 2019/20



103

# Moda d'un multiconjunt de naturals

- La moda d'un multiconjunt és l'element d'aquest que més vegades apareix.
- Estratègia en dues fases:
  - Fase 1: Recorregut sequencial ascendent i array auxiliar de comptadors (frequència) on frequència[i]: nº de vegades que apareix i a l'array.
  - Fase 2: La moda s'obté calculant el màxim de l'array frequència.
- Exemple per a C = {5, 7, 1, 7, 8, 7, 3, 0, 7, 8, 5, 0} amb n = 8 (nº més gran).

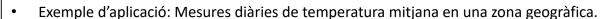


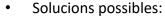
## Moda d'un multiconjunt de naturals

- BlueJ:UtilsArrays
- Fixa't en el mètode modaDeOaN de la classe Recorregut del projecte BlueJ UtilsArrays que, donat un array que conté elements enters en el rang [0..n], calcula la moda (el valor que més es repeteix).
- Prova el mètode en el Code Pad de BlueJ.

```
public static int modaDeOaN(int[] a, int n) {
       // es construeix un array entre 0 i n
       int[] frequencia = new int[n + 1];
       // recorregut d'a i obtencio de frequencies
       for (int i = 0; i < a.length; i++) { frequencia[a[i]]++; }</pre>
       // la moda es el maxim de l'array frequencia
       int moda = 0;
       for (int i = 1; i < frequencia.length; i++) {</pre>
            if (frequencia[i] > frequencia[moda]) { moda = i; }
       return moda;
                             int[] a = {5, 7, 1, 7, 8, 7, 3, 0, 7, 8, 5, 0};
                             int moda = Recorregut.modaDe0aN(a, 8);
                             moda
                             7 (int)
21/11/2019
```

## Arrays multidimensionals: Justificació





 Array temp de 366 elements (per si l'any és bixest) que emmagatzema de forma correlativa les dades de temperatura per a cada dia.

• 1 de gener → temp[0] 3 de febrer → temp[33] 2 de juliol → temp[183]

0

Representació matricial de dimensions 12x31

• Simplifica els càlculs de la posició real de cada dia en l'estructura de dades.

 Representació mitjançant un array multidimensional amb 12 files i grandària variable de columnes:

Aquesta representació redueix

el consum de memòria

31 columnes Aquesta representació 30 desaprofita memoria 0 n. variable de columnes 29 30 Gener té 31 dies Febrer pot tindre 28 o 29 dies Març té 31 dies

Desembre té 31 dies

- Sempre és convenient ajustar la mida de l'array a les dades que es vagen a emmagatzemar per reduir el consum de memòria.
  - Es tracta d'un array de 12 components on cada component és al seu torn un array amb un nombre variable de components. Això és un exemple d'array multidimensional (array d'arrays).

## Arrays multidimensionals: Definició, declaració i ús

- Definició: Un array multidimensional és un array els components del qual són també arrays.
- La dimensió d'un array és el nombre de definicions niuades que inclou. La dimensió defineix el nombre d'índexs que s'utilitzaran per accedir als elements primitius de l'array.
- Cada niuament és un array d'una dimensió menor.
- Valors: Es representen com una successió d'elements entre claus i separats per comes. Per exemple: {{1, 2}, {1, 2, 3}, {1}} és un array bidimensional d'enters amb tres arrays unidimensionals.
- Operador: L'operador d'accés [] ha de repetir-se per a cada dimensió.

```
nomVble[index1][index2]...[indexN]
```

indexk és una expressió que s'avalua a un índex vàlid per l'array nomvble en la dimensió k

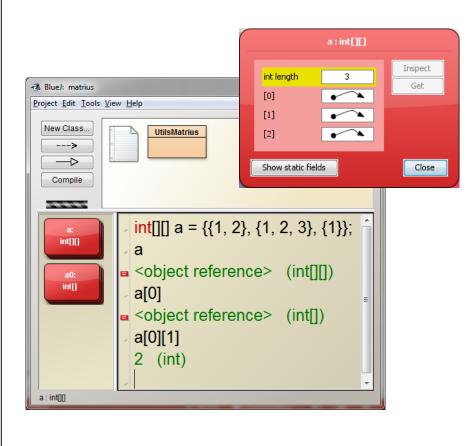
- Declaració i creació:
- Declaració: han d'aparèixer tants parells de claudàtors com la dimensió de l'array.
   tipus[]...[] nom/ble;
- Inicialització: han d'aparèixer tants parells de claudàtors com la dimensió de l'array i, almenys, la primera dimensió ha de tenir una mida específica. Les altres dimensions poden estar buides.

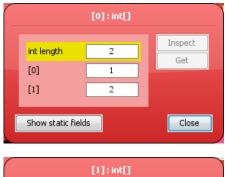
```
nomVble = new tipus[expressió1]...[expressióN];
```

Conjuntament:

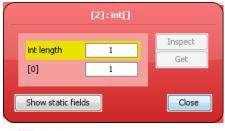
tipus[]...[] nomVble = new tipus[expressió1]...[expressióN];

# Arrays multidimensionals: Definició, declaració i ús

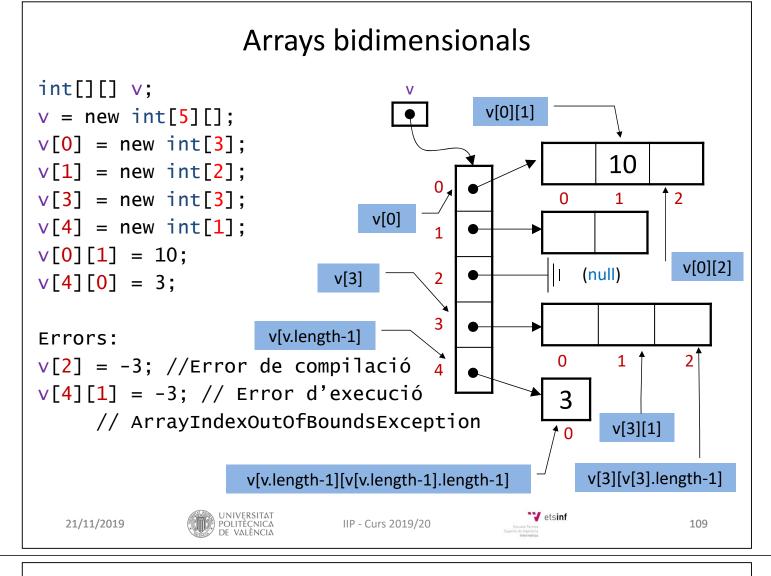






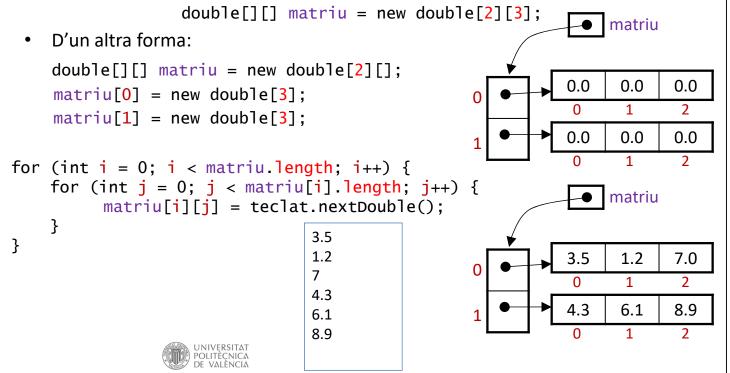






## Arrays bidimensionals: Matrius

- Les matrius són un cas particular d'arrays bidimensionals on tots els arrays de la segona dimensió tenen la mateixa mida.
- Es poden definir amb una única instrucció new.
- Exemple: matriu amb dues files i tres columnes:



## Arrays de dimensió N

- Per a dimensions majors que 2 es segueixen les mateixes regles. Per exemple :
- Tres dimensions:

```
int[][][] tres = {{{1}}}, {{1, 2}, {3, 4}}, {{1}, {2}}};
int[][] dos = {{5, 4}, {6, 7}};
tres[1] = dos;
tres[1][0][1] = 6;
```

Quatre dimensions:

```
int[][][][] quatre = new int[10][2][][];
for (int i = 0; i < quatre.length; i++) {
    for (int j = 0; j < quatre[i].length; j++) {
        quatre[i][j] = new int[3][3];
    }
}</pre>
```

De forma equivalent:

```
int[][][][] quatre = new int[10][2][3][3];
```

21/11/2019



IIP - Curs 2019/20



111

# Arrays bidimensionals Exemple: Temperatura



• Declaració d'una matriu de 12x31 per emmagatzemar les mesures diàries de temperatura per a cada mes.

```
double[][] temp = new double[12][31];
temp[3][6] = 50.1; // Temp. del dia 7 d'abril
```

• Declaració d'un array multidimensional de 12 files i nombre variable de columnes.

```
double[][] temp = new double[12][];
// temp.length és igual a 12
temp[0] = new double[31]; // índexs del 0 al 30
temp[1] = new double[29]; // índexs del 0 al 28
// ...
temp[2][4] = 78.0; // Temp. del dia 5 de març
```



# Arrays bidimensionals Exemple: Temperatura



Alternativa d'implementació amb definició prèvia del nombre de dies.

```
final int[] NUM_DIES = {31,28,31,30,31,30,31,30,31,30,31}
// NUM_DIES[i] = no dies del mes, 0 <= i <= 11

double[][] temp = new double[12][];
// temp[i] representa el mes, 0 <= i <= 11

for (int i = 0; i < temp.length; i++) {
    temp[i] = new double[NUM_DIES[i]];
    // el no elements de temp[i] = NUM_DIES[i], 0 <= i <= 11
}</pre>
```

- temp[i][j] representa la temperatura mitjana del dia (j+1) del mes (i+1).
  - temp[3][14] és la temperatura mitjana del 15 d'abril.

21/11/2019



IIP - Curs 2019/20



113

# Arrays bidimensionals Exemple: Temperatura



• Alternativa d'implementació usant numeració des d'1.

```
final int[] NUM_DIES = {0,31,28,31,30,31,30,31,30,31,30,31};
// NUM_DIES[0] = 0 i NUM_DIES[i] = n° dies del mes i, 1<=i<=12

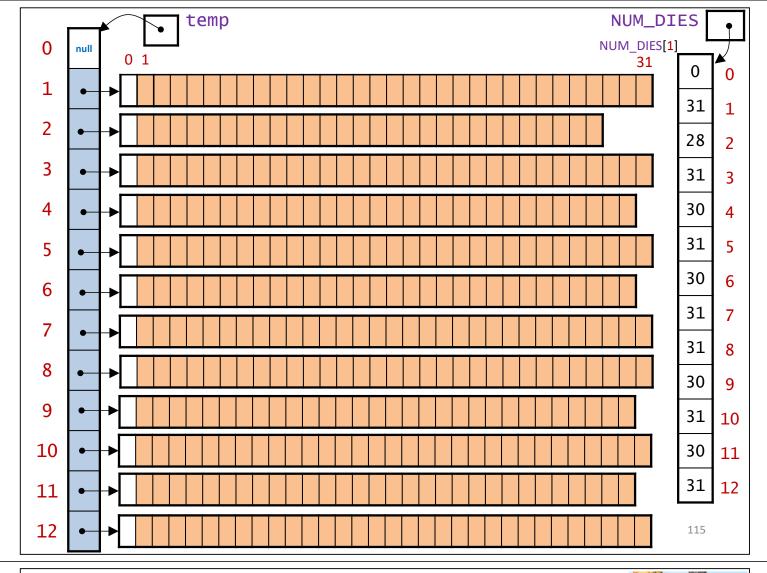
double[][] temp = new double[13][];
// temp[0] = null i temp[i] representa el mes i, 1<=i<=12

for (int i = 1; i < temp.length; i++) {
    temp[i] = new double[NUM_DIES[i] + 1];
    // el n° d'elements de temp[i] = NUM_DIES[i] + 1, 1<=i<=12
}</pre>
```

- Ara, temp[i][j] representa la temperatura mitjana del dia j del mes i.
  - temp[4][15] és la temperatura mitjana del 15 d'abril.







# Recorregut en arrays bidimensionals

### TemperaturaAnual - temperatura

- En la classe TemperaturaAnual del projecte BlueJ temperatura:
- mostrarTemp mostra per pantalla la temperatura enregistrada cada dia de l'any.
- mostrarPrimeraMiniDia mostra la primera mesura mínima i quan es va produir.

```
public void mostrarTemp() {
    for (int i = 1; i < temp.length; i++) {
        System.out.println("Mes: " + i);
        for (int j = 1; j < temp[i].length; j++) {
            System.out.print("Dia: " + j);
            System.out.println(" Temp: " + temp[i][j]);
        }
    }
}</pre>
```

116

## Cerca en arrays bidimensionals

TemperaturaAnual - temperatura

UtilsMatrius - matrius

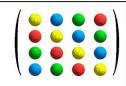
• En la classe TemperaturaAnual del projecte BlueJ *temperatura*, el métode mostrarPrimerDia40 mostra per pantalla la primera data en la que es van enregistrar 40º.

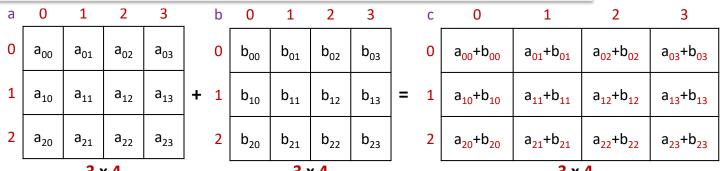


```
public void mostrarPrimerDia40() {
   int i = 1, j = 0;
   boolean trobat = false;
   while (i < temp.length && !trobat) {
        j = 1;
        while (j < temp[i].length && !trobat) {
            trobat = (temp[i][j] == 40);
            j++;
        }
        if (trobat) {
            System.out.print("40° mesurat el dia " + (j - 1));
            System.out.println(" del mes " + (i - 1));
        }
        else { System.out.println("Cap dia amb 40°"); }
}</pre>
```

## Exemple: Suma de matrius

En la classe UtilsMatrius del projecte BlueJ *matrius*, el métode sumaMat torna la matriu resultant de sumar les matrius a i b de igual dimensió.





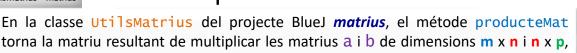
```
      3 x 4
      3 x 4

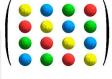
      m x n
      m x n
```

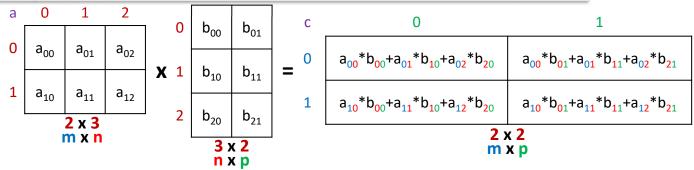


respectivament.

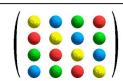
## Exemple: Producte de matrius





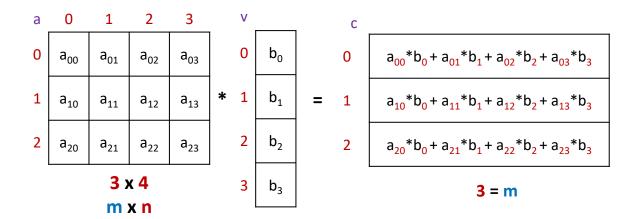


## Exercici: Producte de matriu per vector



### UtilsMatrius - matrius

En la classe <u>UtilsMatrius</u> del projecte BlueJ *matrius*, completa el métode matPerVec per tal que torne el producte d'una matriu a per un vector v. La dimensió d'a és m x n i la de v és n, m > 0, n > 0.





## Exercici: Transposada d'una matriu



UtilsMatrius - matrius

En la classe UtilsMatrius del projecte BlueJ matrius, completa el métode transpMat per tal que torne la transposada d'una matriu a donada. La dimensió d'a és m x n, m > 0, n > 0.

a 0 1 0  $a_{00}$  $a_{01}$  $a_{02}$  $a_{03}$ 3 x 4 1 a<sub>13</sub> a<sub>11</sub> a<sub>12</sub> a<sub>10</sub> m x n 2 a<sub>20</sub> a<sub>21</sub> a<sub>22</sub> a<sub>23</sub>

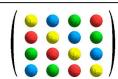
at	0	1	2	
0	a <sub>00</sub>	a <sub>10</sub>	a <sub>20</sub>	
1	a <sub>01</sub>	a <sub>11</sub>	a <sub>21</sub>	
2	a <sub>02</sub>	a <sub>12</sub>	a <sub>22</sub>	
3	a <sub>03</sub>	a <sub>13</sub>	a <sub>23</sub>	

4 x 3 n x m



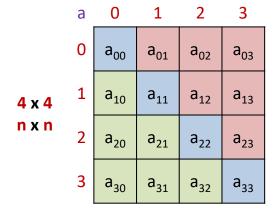


# Exercici: Transposada d'una matriu quadrada



UtilsMatrius - matrius

 En la classe UtilsMatrius del projecte BlueJ matrius, completa el métode transpMatQuad per tal que transpose la matriu quadrada a donada. La dimensió d'a és n x n, n > 0.



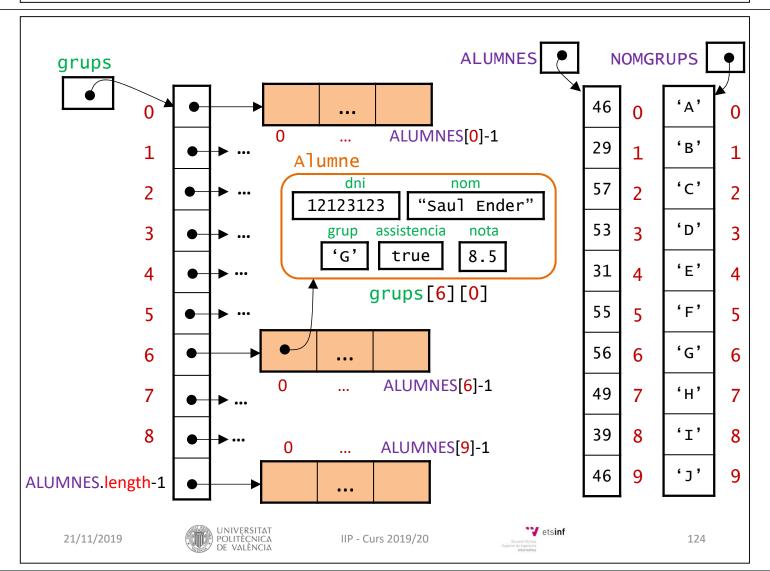
a <sup>t</sup>	0	1	2	3	
0	a <sub>00</sub>	a <sub>10</sub>	a <sub>20</sub>	a <sub>30</sub>	
1	a <sub>01</sub>	a <sub>11</sub>	a <sub>21</sub>	a <sub>31</sub>	4 x 4
2	a <sub>02</sub>	a <sub>12</sub>	a <sub>22</sub>	a <sub>32</sub>	n x n
3	a <sub>03</sub>	a <sub>13</sub>	a <sub>23</sub>	a <sub>33</sub>	





## Exemple: grups de primer curs

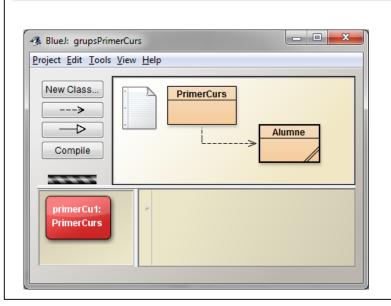
```
PrimerCurs - grupsPrimerCurs
class PrimerCurs {
  final int[] ALUMNES = \{46, 29, 57, 53, 31, 55, 56, 49, 39, 46\};
  final char[] NOMGRUPS = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J'};
  private Alumne[][] grups;
  public PrimerCurs(Scanner tec) {
        grups = new Alumne[ALUMNES.length][];
        for (int i = 0; i < grups.length; <math>i++) {
            grups[i] = new Alumne[ALUMNES[i]];
            for (int j = 0; j < grups[i].length; j++) {
                  grups[i][j] = new Alumne(tec);
                  grups[i][j].setGrup(NOMGRUPS[i]);
                                       class Alumne {
        }
                                         private long dni;
  }
                                         private double nota:
                                         private String nom;
                                         private boolean assistència;
}
                                         private char grup;
                    UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA
                                      }
   21/11/2019
                                 IIP - Cur
```

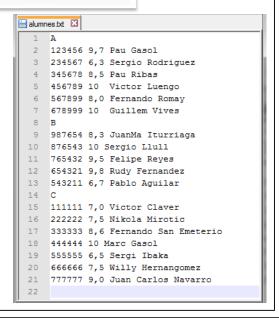


## Exemple: grups de primer curs

#### PrimerCurs - grupsPrimerCurs

- Per poder provar els mètodes de la classe PrimerCurs del projecte BlueJ
  grupsPrimerCurs, s'ha definit un constructor que llig les dades d'un fitxer de text
  (alumnes.txt), evitant d'aquesta manera tenir que teclejar-los en cada prova.
- Comenta les declaracions de les constants ALUMNES i NOMGRUPS i descomenta les que hi ha definides a continuació per a 3 grups de 6, 5 i 7 alumnes, respectivament.
- Crea un objecte PrimerCurs amb aquest constructor, passant-li com a paràmetre el nom del fitxer. Inspecciona'l i comprova que els alumnes se corresponen amb els del fitxer.





## Exemple: grups de primer curs

#### PrimerCurs - grupsPrimerCurs

 En la classe PrimerCurs del projecte BlueJ grupsPrimerCurs, el métode llistarAlumnes torna un llistat amb la informació de tots els alumnes d'un grup donat.

```
public String llistarAlumnes(char g) {
   int i = 0;
   while (i < NOMGRUPS.length && NOMGRUPS[i] != g) { i++; }
   String res = "";
   if (i < NOMGRUPS.length) {
      for (int j = 0; j < grups[i].length; j++) {
        res += grups[i][j] + "\n";
      }
   }
   else { res = g + "no és un grup vàlid"; }
   return res;
}</pre>
```



## Exemple: grups de primer curs

### PrimerCurs - grupsPrimerCurs

En la classe PrimerCurs del projecte BlueJ grupsPrimerCurs, el métode notaMinima torna el darrer alumne amb la nota mínima de tot primer.

```
public Alumne notaMinima() {
    double notaMin = Double.MAX VALUE;
    Alumne aMin = null;
    for (int i = 0; i < grups.length; i++) {
        for (int j = 0; j < grups[i].length; j++)
            if (grups[i][j].getNota() <= notaMin)
                notaMin = grups[i][j].getNota();
                aMin = grups[i][j];
    return aMin;
```

21/11/2019

PrimerCurs - grupsPrimerCurs



IIP - Curs 2019/20



- - X

Inspect

Get

: double[]

3

8.86000000000000001

8.014285714285714

Inspect

Close

127

## Exercicis: grups de primer curs

- En la classe PrimerCurs del projecte BlueJ grupsPrimerCurs, completa els següents métodes:
- mitjanaGrup, que torna un array amb la nota mitjana de cada grup.
- matrHonorGrup, que torna un llistat amb el primer alumne amb Matrícula d'Honor (nota 10) de cada grup.
- Crea un objecte PrimerCurs amb la informació del fitxer alumnes.txt, i prova els mètodes que has implementat.

