



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Escola Tècnica Superior d'Enginyeria Informàtica



Tema 5. Estructures de control: selecció

Introducció a la Informàtica i la Programació (IIP)

Curs 2019/20

Departament de Sistemes Informàtics i Computació



Continguts

Duració: 3 sessions

1. Introducció: la necessitat d'una estructura de selecció per a programar
2. Instruccions condicionals
 - Instrucció `if...else...`
 - simple
 - general (niuada i múltiple)
 - l'operador condicional o ternari
 - Instrucció `switch`



- **Crea** una carpeta **Tema 5** dins de la teua carpeta **W:\IIP**
- **Descarrega** (del Tema 5 de PoliformaT) els fitxers **exemplesT5.jar** i **exercicisT5.jar** en **Tema 5**
- Des de l'opció **Projecte** de **BlueJ**, usa l'opció **Open ZIP/JAR...** per tal d'obrir-los com projectes **BlueJ** i prepara't per usar-los

Introducció: la necessitat d'una estructura de selecció per a programar

- Per resoldre problemes reals quasi sempre hi ha que prendre **decisions** en funció de les circumstàncies i, aleshores, **escollir** una d'entre diverses **alternatives**.
- Fins ara hem considerat programes que són simples **seqüències d'instruccions**. Però en un programa, molt a sovint, és necessari prendre decisions en funció de les dades d'entrada.
- La seqüència d'instruccions pot alterar-se depenent de les dades d'entrada o de resultats intermedis que es generen durant l'execució.
- Les **estructures de control** que permeten canviar el **fluxe d'execució** d'un programa són **selecció** i **repetició** (iteració).
- A continuació, es presenten les instruccions que proporciona Java per prendre decisions: **instruccions de selecció** o **condicionals**.

Per què instruccions de selecció?

Exemple 1: validar dades

```
public class TimeInstant {  
    private int hours, minutes;  
    /** TimeInstant corresponent a les hh hores i mm minuts.  
     * Precondicio: 0 <= hh < 24, 0 <= mm < 60 */  
    public TimeInstant(int hh, int mm) { hours = hh; minutes = mm; }  
    ...  
}
```

```
public class Test4 {  
    private Test4() { }  
    public static void main(String[] args) {  
        Scanner teclat = new Scanner(System.in);  
        System.out.println("Lectura de teclat d'una hora.");  
        System.out.print(" -> Introduiu les hores (entre 0 i 23): ");  
        int hor = teclat.nextInt();  
        System.out.print(" -> Introduiu els minuts (entre 0 i 59): ");  
        int min = teclat.nextInt();  
        TimeInstant hUsuari = new TimeInstant(hor, min);  
        ...  
    }  
}
```

Què passa si les **dades** llegides de teclat
NO són vàlides?

Per què instruccions de selecció?

Exemple 1: validar dades

```
public class TimeInstant {  
    private int hours, minutes;  
    /** TimeInstant corresponent a les hh hores i mm minuts.  
     * Precondicio: 0 <= hh < 24, 0 <= mm < 60 */  
    public TimeInstant(int hh, int mm) { hours = hh; minutes = mm; }  
    ...  
}
```

Necessitem instruccions que ens permeten
“prendre decisions”

```
public class Test4 {  
    private Test4() { }  
    public static void main(String[] args) {  
        Scanner teclat = new Scanner(System.in);  
        System.out.println("Lectura de teclat d'una hora.");  
        System.out.print(" -> Introduiu les hores (entre 0 i 23): ");  
        int hor = teclat.nextInt();  
        System.out.print(" -> Introduiu els minuts (entre 0 i 59): ");  
        int min = teclat.nextInt();  
        TimeInstant hUsuari;  
        SI (hor i min son vàlides)  
        ALESHORES hUsuari = new TimeInstant(hor, min);  
        SINO informa a l'usuari i ...  
        ...  
    }  
}
```

Per què instruccions de selecció?

Exemple 2: màxim de dos

```
public class TimeInstant {
    private int hours, minutes;
    ...
    /** Torna un valor negatiu si el TimeInstant (this) és anterior a
     *  altre, un valor positiu si el TimeInstant (this) és posterior
     *  a altre i zero si són iguals */
    public int compareTo(TimeInstant altre) {
        return this.aMinuts() - altre.aMinuts();
    }
}
```

```
public class Test4 {
    private Test4() { }
    public static void main(String[] args) {
        ...
        TimeInstant hUsuari = new Instant(hor, min);
        TimeInstant hUTC = new Instant();
        ...
        int res = hUsuari.compareTo(hUTC);
        TimeInstant max;
        ...
    }
}
```

Com emmagatzemar en una variable **max**
el més gran dels dos instants?

Per què instruccions de selecció?

Exemple 2: màxim de dos

**Necessitem instruccions que ens permeten
“prendre decisions”**

```
public class TimeInstant {
    private int hours, minutes;
    ...
    /** Torna un valor negatiu si el TimeInstant (this) és anterior a
     *  altre, un valor positiu si el TimeInstant (this) és posterior
     *  a altre i zero si són iguals */
    public int compareTo(TimeInstant altre) {
        return this.aMinuts() - altre.aMinuts();
    }
}
```

```
public class Test4 {
    private Test4() { }
    public static void main(String[] args) {
        ...
        TimeInstant hUsuari = new Instant(hor, min);
        TimeInstant hUTC = new Instant();
        ...
        int res = hUsuari.compareTo(hUTC);
        TimeInstant max;
        SI (res > 0) ALESHORES max = hUsuari; SINO max = hUTC;
        ...
    }
}
```

Instrucció condicional if

- L'estructura condicional més simple en Java és la següent:

```
if (B) { S; }
```

on B és una condició i S és una instrucció o un bloc qualsevol d'instruccions.

- Execució**

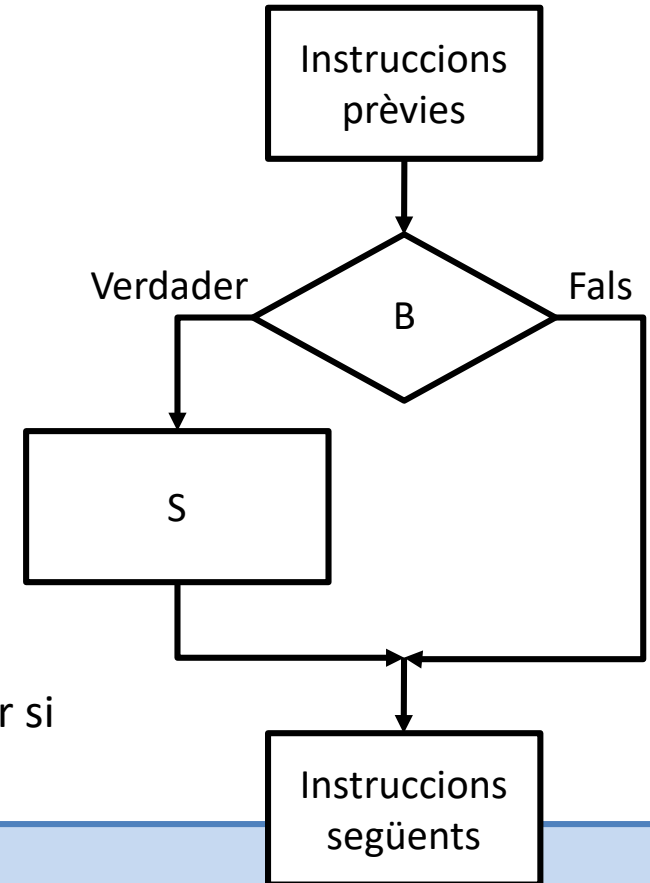
- Avaluar la condició B.
- Si B és verdadera, executar S.
- Continuar amb la següent instrucció a la condicional.

- Exemple:** dades mèdiques; segons el gènere, preguntar si està embarassada:

Si bloc d'instruccions
{ } obligatòries

Si una única instrucció
{ } NO obligatòries
però **RECOMANADES**
segons les **convencions**
d'escriptura de **codi Java**
(**Checkstyle**)

```
Huma h = new Huma();  
...  
System.out.print("Quina edat tens? ");  
...  
if (h.getGenere() == 'F') {  
    System.out.print("Estàs embarassada? ");  
    ...  
}  
System.out.print("Ets al·lèrgic/a a algun medicament? ");  
...  
UN  
PO  
DE
```



Instrucció condicional if ... else ...

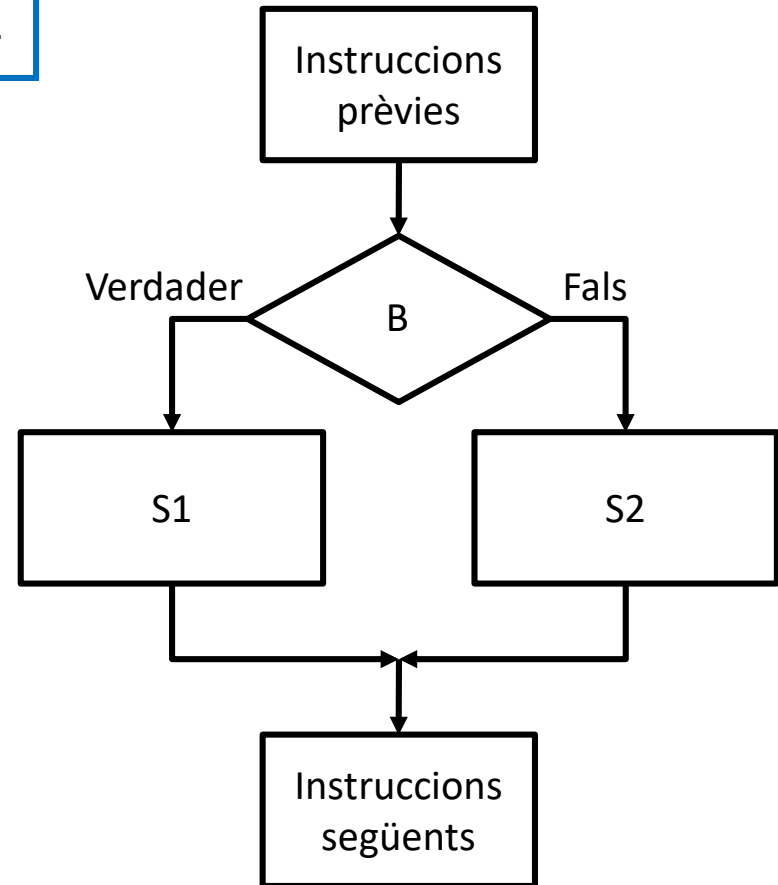
- La seua forma general és

```
if (B) { S1; } else { S2; }
```

on B és una condició i S1, S2 són instruccions o blocs d'instruccions qualsevols.

- Execució**

1. Avaluar la condició B .
2. Si B és verdadera, executar S1.
3. Si B és falsa, executar S2.
4. Continuar amb la següent instrucció a la condicional.



Instrucció condicional if ... else ...

- **Exemple:** obtenir en **max** el major de 2 números x i y

➤ **Versió 1:**

1 comparació

```
if (x > y) { max = x; }  
else { max = y; } /* x <= y */
```

➤ **Versió 2:**

2 comparacions

```
if (x >= y) { max = x; }  
else { max = y; } /* x < y */
```

➤ **Versió 3:**

1 comparació

```
max = y; // hipòtesi inicial: y és el major  
if (x > y) { max = x; }
```

Quina és la millor?

La(les) que **menor nº de comparacions** requerisca(quen) en el Pitjor dels Casos

Instrucció condicional if ... else ...

Exemple: màxim
de dos números

BlueJ:exemplesT5

```
public class UtilMeu {
    /** Torna el major de dos enters x i y, o y si x és igual a y */
    public static int max(int x, int y) {
        int max = y;
        if (x > max) { max = x; }
        return max;
    }
    /** Torna el major de dos reals x i y, o y si x és igual a y */
    public static double max(double x, double y) {
        if (x > y) { return x; }
        else { return y; }
    }
}

import java.util.Locale; import java.util.Scanner;
public class TestUtilMeu {
    private TestUtilMeu() { }
    public static void main(String[] args) {
        Scanner teclat = new Scanner(System.in).useLocale(Locale.US);
        System.out.print("Introdueix dos valors reals: ");
        double num1 = teclat.nextDouble();
        double num2 = teclat.nextDouble();
        double max = UtilMeu.max(num1, num2);
        System.out.printf(Locale.US, "El màxim és %4.2f\n", max);
    }
}
```



- La classe **Bonometro** representa el títol de transport amb el que se pot viatjar al metro.
 - Cada bonometro té el nombre de tiquets disponibles i tots els bonometros comparteixen una quantitat per defecte de recàrrega que és 10.
 - Un bonometro es pot carregar amb un nombre donat de viatges o per defecte; es pot consultar el nombre de viatges disponible i es pot recarregar.
 - El mètode **picar** comprova si queden viatges, en aquest cas actualitza el nombre de tiquets i torna el missatge:
"Bono amb xxx tiquets".
Si no queden viatges torna el missatge:
"Bono esgotat. Recàrrega'1 JA!"

Instrucció condicional if ... else ...

Exemple: Bonometro

BlueJ:exemplesT5

```
public class BonoMetro {
    public static final int BONO_10 = 10;
    // Un BonoMetro TÉ UN ...
    private int numTiquets;

    public BonoMetro(int n) { numTiquets = n; }

    public BonoMetro() { this(BONO_10); }

    public int getNumTiquets() { return numTiquets; }

    public void recarregar(int quantitat) { numTiquets += quantitat; }

    public String toString() {
        return "Bono amb " + numTiquets + " viatges";
    }

    public String picar() {
        if (numTiquets > 0) {
            numTiquets--;
            return this.toString();
        }
        else { return "Bono esgotat. Recàrrega'l JA!"; }
    }
}
```



Instrucció condicional `if ... else ...` **niuada**

- Els blocs d'instruccions poden ser a la seua vegada instruccions condicionals: instrucció condicional **niuada**.

Quina instrucció s'executa en funció dels valors de B1, B2 i B3?

```
if (B1) {  
    if (B2) { i1; }  
    else { i2; }  
}
```

```
if (B1) {  
    if (B2) { i1; }  
}  
else { i2; }
```

L'else s'associa al darrer
if que no tinga else

```
if (B1) {  
    if (B2) { i1; }  
    else { i2; }  
}  
else {  
    if (B3) { i3; }  
    else { i4; }  
}
```

B1 \ B2	true	false
true	i1	i2
false	-	-

B1 \ B2	true	false
true	i1	-
false	i2	i2

B1	B2	B3	instrucció
true	true	-	i1
true	false	-	i2
false	-	true	i3
false	-	false	i4

Instrucció condicional if ... else ...

Exercici 3 del llibre d'IIP

Blue:exercisT5

- En una tenda d'electrodomèstics, per liquidació, s'apliquen distints descomptes en funció del total de les compres realitzades:
 - Si $\text{total} < 500 \text{ €}$, no s'aplica descompte.
 - Si $500 \text{ €} \leq \text{total} \leq 2000 \text{ €}$, s'aplica un descompte del 30%.
 - Si $\text{total} > 2000 \text{ €}$, aleshores s'aplica un descompte del 50%.
- Per tal d'implementar el problema en Java, s'ha d'usar **una única variable** (`total` de tipus `double`) que emmagatzema el total de les compres realitzades (abans d'aplicar el descompte) i també el total a cobrar (després d'aplicar el descompte corresponent). És a dir, la variable `total` és, a la vegada, dada i resultat. S'ha de resoldre el problema amb **una única instrucció condicional (niuada)**. Completar la instrucció si la condició és `total >= 500` o `total <= 2000`.

```
if (total >= 500)
```

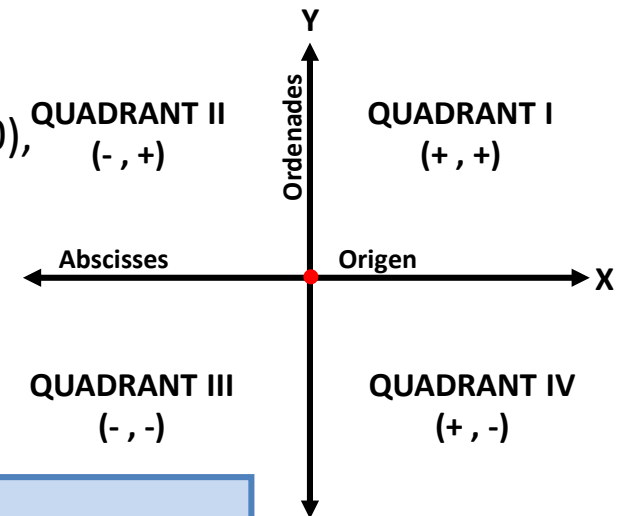
```
if (total <= 2000)
```

Instrucció condicional if ... else ...

- Donat p un objecte Punt representant un punt (x, y), on està situat al pla cartesià?

- Anàlisi per casos:**

- si $x = 0$, aleshores,
 - si $y = 0$, el punt està a l'origen de coordenades (0, 0),
 - si $y \neq 0$, el punt està a l'eix d'ordenades Y.
- si $x \neq 0$, aleshores,
 - si $y = 0$, el punt està a l'eix d'abscisses X,
 - si $y \neq 0$, el punt està a qualsevol dels 4 quadrants.



BlueJ:exemplesT5

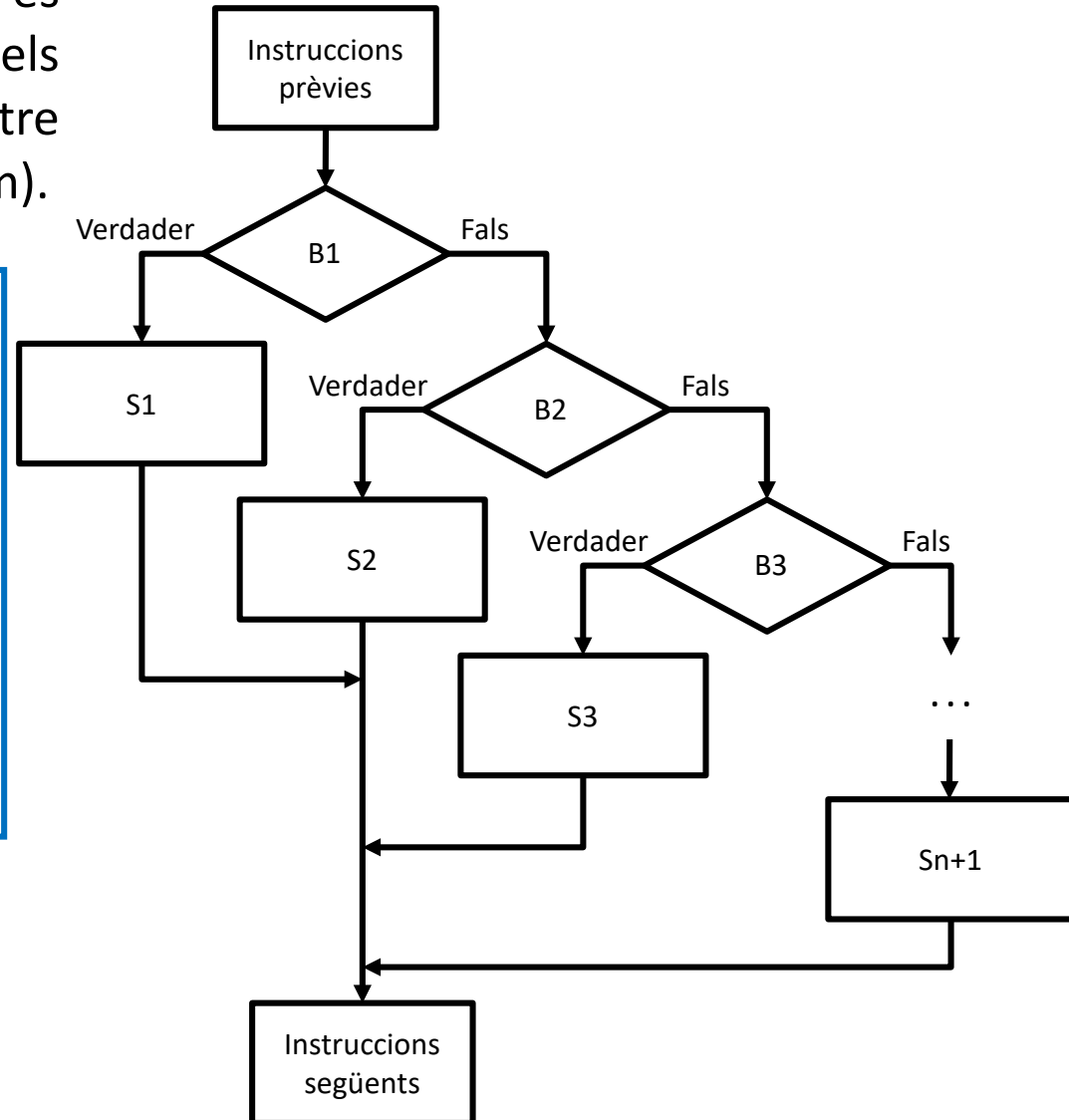
```
public class Punt
```

```
public static final int QUADRANTS = 0;
...
public static final int EIX_ABSCISSES = 5;
public static final int EIX_ORDENADES = 6;
public static final int CENTRE_COORDENADES = 7;
private double x;
private double y;
...
public int posicioEnPla() {
    if (x == 0) {
        if (y == 0) { return CENTRE_COORDENADES; }
        else { return EIX_ORDENADES; }
    } else {
        if (y == 0) { return EIX_ABSCISSES; }
        else { return QUADRANTS; }
    }
}
```


Instrucció condicional if ... else ... múltiple

- Una instrucció if...else... es diu **múltiple** quan cadascun dels else s'associa a un altre if...else... (excepte l'últim).

```
if (B1) { S1; }  
else if (B2) { S2; }  
else if (B3) { S3; }  
...  
else if (Bn) { Sn; }  
else { Sn+1; }
```



Instrucció condicional if ... else ... múltiple

Quina instrucció s'executa en funció dels valors de B1, B2 i B3?

```
if (B1) { i1; }  
else if (B2) { i2; }  
else if (B3) { i3; }  
else { i4; }
```

B1	B2	B3	instrucció
true	-	-	i1
false	true	-	i2
false	false	true	i3
false	false	false	i4

BlueJ:exemplesT5

```
public class TestIfsPunt
```

```
public static void main(String[] args) {  
    ...  
    int pos = p1.posicioEnPla();  
    System.out.print("El punt està a ");  
    if (pos == Punt.CENTRE_COORDENADES) {  
        System.out.println("l'origen de coord. (0,0)");  
    } else if (pos == Punt.EIX_ORDENADES) {  
        System.out.println("l'eix d'ordenades Y");  
    } else if (pos == Punt.EIX_ABSCISSES) {  
        System.out.println("l'eix d'abscisses X");  
    } else { System.out.println("qualsevol dels 4 quadrants"); }  
}
```

Instrucció condicional if ... else ...

Exemple: comparar
dos números reals

BlueJ:exemplesT5

```
public class TestIfsPunt {
    private TestIfsPunt() { }

    public static void main(String[] args) {
        Punt p1 = new Punt(4, 5);
        Punt p2 = new Punt();
        final String MSG_DIST_ORIG = " distància de l'origen ";
        double dop1 = p1.distOrigen();
        double dop2 = p2.distOrigen();
        if (dop1 > dop2) {
            System.out.println("p1 està a major"
                               + MSG_DIST_ORIG + "que p2");
        } else if (dop1 < dop2) {
            System.out.println("p2 està a major"
                               + MSG_DIST_ORIG + "que p1");
        } else {
            System.out.println("p1 i p2 estan a igual"
                               + MSG_DIST_ORIG);
        }
    }
}
```

Instrucció condicional if ... else ... múltiple

Exemple: convertir una nota a l'equivalent en lletra

- Anàlisi per casos:

- Si $0 \leq \text{nota} < 5$, suspens.
- Si $5 \leq \text{nota} < 7$, aprovat.
- Si $7 \leq \text{nota} < 9$, notable.
- Si $9 \leq \text{nota} < 10$, excel·lent.
- Si $\text{nota} = 10$, matrícula d'honor.
- En qualsevol altre cas, error.

BlueJ:examplesT5

```
public class UtilMeu
```

```
public static String deNotaALletra(double nota) {  
    String notaLletra;  
    if (nota < 0.0 || nota > 10.0) { notaLletra = "Error"; }  
    else if (nota < 5.0) { notaLletra = "Suspens"; }  
    else if (nota < 7.0) { notaLletra = "Aprovat"; }  
    else if (nota < 9.0) { notaLletra = "Notable"; }  
    else if (nota < 10.0) { notaLletra = "Excel·lent"; }  
    else { notaLletra = "Matrícula d'Honor"; }  
    return notaLletra;  
}
```

Instrucció condicional if ... else ...

Exemple: comprovar si una data és anterior a un altra

BlueJ:exemplesT5


```
public class Data
```

```
public boolean esAnterior(Data d) {  
    boolean anterior = false;  
    if (any < d.any) { anterior = true; }  
    else if (any == d.any) {  
        if (mes < d.mes) { anterior = true; }  
        else if (mes == d.mes && dia < d.dia) { anterior = true; }  
    }  
    return anterior;  
}
```

```
public boolean esAnterior(Data d) {  
    return (any < d.any  
        || (any == d.any  
            && (mes < d.mes || (mes == d.mes && dia < d.dia))));  
}
```

Instrucció condicional if ... else ...

Exercici 4 del llibre d'IIP

 Blue:exercicisT5

- El mètode estàtic **exercici4** de la classe **Condicionals**, donats **x** de tipus **int** i **c** de tipus **char**, mostra un missatge per pantalla en funció dels valors de **x** i **c** segons la instrucció:

```
if (x < 0 && c == 'x') { System.out.println("Cas 1"); }  
else if (x < 0 && c != 'x') { System.out.println("Cas 2"); }  
else if (x >= 0 && c == 'y') { System.out.println("Cas 3"); }  
else if (x >= 0 && c != 'y') { System.out.println("Cas 4"); }
```
- Reescriu** el codi amb la següent estructura, col·locant les condicions i instruccions **System.out.println()** adequades, de forma que donats qualssevol **x**, **c**, el resultat escrit en l'eixida estàndard coincidisca:

```
if (x < 0) {  
  
}  
else {  
  
}
```

Instrucció condicional if ... else ...



Classe Astre

- Completa en la classe **Astre** el mètode **visibleAmb** per tal que torne un **String** que descriga la forma en la que l'astre pot ser observat, tenint en compte el següent:

Brillantor	Visibilitat
< 5	"a simple vista"
$\geq 5 \text{ i } < 7$	"amb prismàtics"
$\geq 7 \text{ i } \leq 25$	"amb telescopi"
> 25	"amb grans telescopis"

Instrucció condicional if ... else ...

Exemple: comparar

- En la classe **Menjar**, el mètode **compareTo** indica si un menjar és *més* sa que un altre, és a dir, si aporta *menys calories* que l'altre i, a igualtat de calories, si se prepara en un *temps menor* que l'altre. Torna un nº negatiu, zero o positiu, segons el cas.

BlueJ:exemplesT5

```
public class Menjar {
    // Un Menjar TÉ UNES ...
    private double calories; private int minutsPrep;
    public Menjar(double c, int m) { calories = c; minutsPrep = m; }

    ...
    public int compareTo(Menjar altre) {
        // Primera versió:
        int res = 0;
        if (calories < altre.calories) { res = -1; }
        else if (calories > altre.calories) { res = 1; }
        else if (minutsPrep < altre.minutsPrep) { res = -1; }
        else if (minutsPrep > altre.minutsPrep) { res = 1; }
        return res;
    }
}
```

Class Math

```
static  signum(double d)
double
```

Returns the signum function of the argument; zero if the argument is zero, 1.0 if the argument is greater than zero, -1.0 if the argument is less than zero.

```
// Segona versió:
int res = minutsPrep - altre.minutsPrep;
if (calories < altre.calories) { res = -1; }
else if (calories > altre.calories) { res = 1; }
return res;
```

```
// Tercera versió:
double difCalories = Math.signum(calories - altre.calories);
int res = minutsPrep - altre.minutsPrep;
if (difCalories != 0) { res = (int) difCalories; }
return res;
```


Instrucció condicional if ... else ...



Classe Astre

- Completa en la classe **Astre** el mètode **mesBrillant** per tal que torne **1** si l'**Astre** actual és més brillant en magnitud absoluta que un **Astre** donat, **0** si tenen la mateixa magnitud absoluta i **-1** si el **Astre** donat és més brillant en magnitud absoluta que l'actual. Cal notar que s'usa la magnitud absoluta dels dos astres ja que, només situant-los a la mateixa distància, pot comparar-se la seua lluminositat.

Instrucció condicional if ... else ...

Exercicis 10, 11 i 22 del llibre d'IIP

Blue:exercicisT5

- **Completa** el mètode estàtic **noNegatiu** en la classe programa **NegPos** per tal que donats dos números enters, **num1** i **num2**, torne el valor lògic **true** si el producte dels dos números és positiu o zero i torne **false** si el producte dels dos números és negatiu. No hi ha que calcular el producte.

num1 * num2		num2		
		+	-	0
num1	+	+	-	0
	-	-	+	0
	0	0	0	0

- **Completa** el mètode estàtic **deMajorAMenor3** en la classe programa **Condicionals** per tal que, donats tres valors enters distints qualssevol, els mostre per pantalla ordenadament, de major a menor.
- **Completa** el mètode estàtic **simplifica** en la classe programa **Condicionals** per tal que, donats tres valors enters **a**, **b** i **c**, implemente distintes solucions a l'anàlisi per casos següent, fent ús de: operadors curtcircuitats i instruccions condicionals.

```
a > b -> true
a < b -> false
a == b i a > c -> true
a == b i a < c -> false
a == b i a == c -> false
```

L'operador condicional o ternari

- Java (com els llenguatges C i C++) introdueix un operador ternari, paregut a una instrucció condicional:

`exprbool ? expr1 : expr2`

on `exprbool` és qualsevol expressió de tipus `boolean` i `expr1` i `expr2` són expressions qualsevols del mateix tipus.

NO RECOMANAT el seu ús segons les convencions d'escriptura de **codi Java (Checkstyle)**

- Execució**
 - Avaluar `exprbool`.
 - Si `exprbool` és `true`, aleshores el valor de tota l'expressió és el valor de l'expressió `expr1`.
 - En cas contrari, el valor de tota l'expressió és el valor de l'expressió `expr2`.
- Exemples:**

```
int a, b, max;
...
max = a > b ? a : b;
...
```

```
...
if (a > b) { max = a; }
else { max = b; }
...
```

```
...
System.out.println("És anterior l'hora " + hUsuari + " a l'hora UTC " + hUTC + "? "
    + ((difMinuts < 0) ? "SÍ" : "NO"));
...
```

Instrucció condicional switch

```
switch (expressió) {  
    case val1:  
        [SC1] [break;]  
    case val2:  
        [SC2] [break;]  
    . . . . .  
    . . . . .  
    case valn:  
        [SCn] [break;]  
    [default:  
        [SCn+1] [break;]]  
}
```

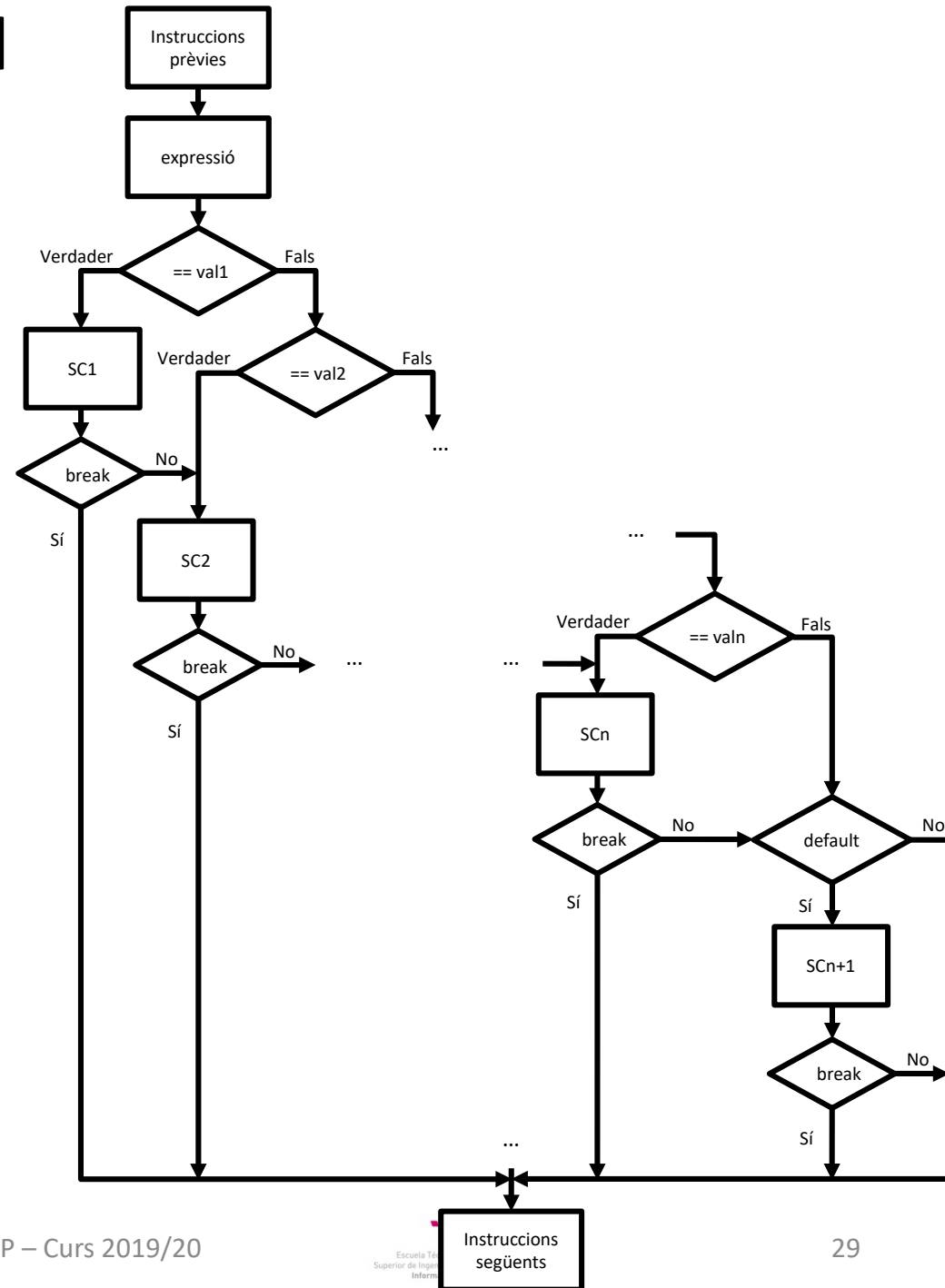
default RECOMANAT
segons les **convencions**
d'escriptura de **codi Java**
(**Checkstyle**)

- expressió és una expressió de qualsevol tipus simple (excepte float i double) o un String (des de la versió 7 de Java).
- val1, val2, ..., valn són valors compatibles amb el d'expressió.
- SC1, SC2, ..., SCn+1 són seqüències d'instruccions qualsevols.
- Les components que apareixen entre claudàtors són opcionals.

Instrucció condicional switch

• Execució

1. Avaluar expressió.
2. Comparar el resultat amb el de cadascun dels valors dels case.
3. Si algun coincideix, executar tot el codi des d'eixe case (incloent-ne codi associat a altres etiquetes case posteriors) fins que:
 - Es troba un break.
 - Acaba el switch.
4. Si cap valor coincideix, executar les instruccions associades a l'etiqueta default (si existeix) fins el final del switch.
5. Continuar amb la instrucció que segueix al switch.



Instrucció condicional switch

Exercicis 7 i 8 del llibre d'IIP

- Indica què es mostra per pantalla després d'executar el següent fragment de codi. Per què?

```
switch (2) {  
    case 1:  
        System.out.println(1); break;  
    case 2:  
        System.out.println(2);  
    case 3:  
        System.out.println(3); break;  
    default:  
        System.out.println(4);  
}
```

- Indica què es mostra per pantalla després d'executar el següent fragment de codi si `primOpcio = 1`. Per què? I si `primOpcio = 2`?

```
switch (primOpcio + 1) {  
    case 1:  
        System.out.println("Amanida"); break;  
    case 2:  
        System.out.println("Paella"); break;  
    case 3:  
        System.out.println("Emperador");  
    case 4:  
        System.out.println("Gelats"); break;  
    default:  
        System.out.println("Bon profit!");  
}
```

Instrucció condicional switch

Exemple: gestió d'un menú

BlueJ:examplesT5

```
import java.util.Locale; import java.util.Scanner;
public class MenuPunt {
    private MenuPunt() { }
    private static int menu(Scanner teclat) {
        System.out.println(" MENÚ");
        System.out.println("1. Distància a l'origen d'un punt");
        System.out.println("2. Moure un punt aleatòriament");
        System.out.println("3. Dades d'un punt");
        System.out.print("\nTria una opció: ");
        int opc = teclat.nextInt();
        return opc;
    }
    public static void main(String[] args) {
        Scanner tec = new Scanner(System.in);
        Punt p = new Punt(1.5, 3.0);
        int opcio = menu(tec);
        switch (opcio) {
            case 1:
                double d = p.distOrigen();
                System.out.printf(Locale.US, "Distància = %.2f\n", d); break;
            case 2:
                p.moureAleat();
                System.out.println("Nou punt = " + p.toString()); break;
            case 3:
                System.out.println(p.toString()); break;
            default:
                System.out.println("Opció no vàlida");
        }
    }
}
```

Instrucció condicional switch

Exemple: a quina estació pertany un mes?

- **Anàlisi per casos:** cada estació dura aproximadament 3 mesos, podem agrupar:
 - Primavera: març, abril i maig (mesos 3, 4 i 5).
 - Estiu: juny, juliol i agost (mesos 6, 7 i 8).
 - Tardor: setembre, octubre i novembre (mesos 9, 10 i 11).
 - Hivern: desembre, gener i febrer (mesos 12, 1 i 2).

BlueJ:examplesT5

```
public class UtilMeu
```

```
public static String deNumMesAEstacio(int numMes) {  
    String estacio = "";  
    switch (numMes) {  
        case 1: case 2: case 12:  
            estacio = "Hivern"; break;  
        case 3: case 4: case 5:  
            estacio = "Primavera"; break;  
        case 6: case 7: case 8:  
            estacio = "Estiu"; break;  
        case 9: case 10: case 11:  
            estacio = "Tardor"; break;  
        default:  
            estacio = "Error"; break;  
    }  
    return estacio;  
}
```

```
if (numMes==12 || numMes==1 || numMes==2) {  
    estacio = "Hivern";  
} else if (numMes >= 3 && numMes <= 5) {  
    estacio = "Primavera";  
} else if (numMes >= 6 && numMes <= 8) {  
    estacio = "Estiu";  
} else if (numMes >= 9 && numMes <= 11) {  
    estacio = "Tardor";  
} else { estacio = "Error"; }
```


Instrucció condicional switch

Exemple: comprovar si una data és correcta

BlueJ:examplesT5

public class Data

```
public boolean esCorrecta() {
    boolean correcta = false;
    if (any > 0 && (mes >= 1 && mes <= 12) && (dia >= 1 && dia <= 31)) {
        int diesMes;
        switch (mes) {
            case 2:
                if (esBixest()) { diesMes = 29; }
                else { diesMes = 28; }
                break;
            case 4:
            case 6:
            case 9:
            case 11:
                diesMes = 30; break;
            default:
                diesMes = 31; break;
        }
        if (dia <= diesMes) { correcta = true; }
    }
    return correcta;
}
```

Instrucció condicional switch



Classe Astre

- Completa en la classe **Astre** el mètode **toString** (que sobreescriu al d'**Object**) per tal que torne un **String** amb el següent format:

“nom: tipus (brillantor, distancia)”

Per exemple, **“Sirius: Estel (-1.42, 8.70)”**

Tots els números han d'arredonar-se a només dues xifres decimals i el tipus de l'astre ha d'apareixer com **“Estel”**, **“Nebulosa”** o **“Galàxia”**. S'ha d'utilitzar la instrucció **switch**.

Classe TestAstre

- En el mètode **main** de la classe **TestAstre**, mostrar un missatge per pantalla indicant quin és l'**Astre** més brillant en magnitud absoluta dels dos que s'han creat.

Instrucció condicional switch

Exercicis 21 i 25 del llibre d'IIP

Blue:exercisT5

- **Completa** la classe programa **Calculadora** per tal que, després de llegir de teclat dos operands, **double num1** i **num2**, i un operador, **char op**, simule una calculadora, considerant que els càlculs possibles són del tipus **num1 op num2**, sent **op** un operador d'entre: +, -, * i /.
- **Completa** la classe programa **AutoEscola** per tal que calcule la tarifa d'una autoescola tenint en compte el tipus de carnet (A, B, C o D) i el nombre de pràctiques realitzades.



Tipus de carnet	Tarifa de les matrícules	Preu per pràctica
A	150 €	15 €
B	325 €	21 €
C	520 €	36 €
D	610 €	50 €



Instruccions condicionals

Exercicis de CAP



- Nuevo `toString()` para la clase `Medicamento` (clau CCDFH4ai)
- `Fecha` con `esFestivo` (clau CCDFJ4ai)
- Piedra, Papel o Tijera (clau CCDFG4ai)
- El mètode `posicioEnPla` de la classe `Punt` (clau DCDFL4ai)

Exercici 13 del llibre d'IIP

Exercici 26 del llibre d'IIP

Exercici 12 del llibre d'IIP