

0101010  
0100101  
1101010

# ANNEX - EXPRESSIONS REGULARS

Programació –1er DAW/DAM

# 0. CONTINGUTS

---

- Introducció.
- Classes de caràcters
- Quantificadors
- Alternatives
- Delimitadors o Àncores
- Classes especials
- Expressions Regulars en Java
- Exemples

# 1. Introducció

- Les **expressions regulars** ens proporcionen **patrons** de **cerca** i/o **substitució**.
- També són conegudes com **regexp** o **regex**.
- Són **àmpliament utilitzades** al món de la programació:
  - *Validació d'entrades de dades.*
  - *Cerques en un fitxer,*
  - *Regles anti-spam,*
  - *Anàlisi de logs*
  - *Web Scraping*



# 1. Introducció

---

- Es tracta d'una **seqüència de caràcters** entre els quals s'inclouen **caràcters simples** i **caràcters especials** (meta-caràcters).
  - **Caràcters simples:** *abcdefghijklmnopqrstuvwxyz0123456*
  - **Caràcters especials:** `[]{}.*+\ $^()|?`
- Els **caràcters especials**, no s'interpreten literalment, sinó que tenen un significat especial dins de l'expressió.

***^[A-F][0-9]\$***

---- Meta-caràcters

---- Caràcters simples

# 1. Introducció

```
public class Example {  
  
    public static void main(String[] args) {  
  
        final String REGEXP_TIRADA_VALIDA ="^[A-F][0-9]$";  
        Scanner teclat = new Scanner(System.in);  
        System.out.println("On vols col·locar la fitxa [A0-F9]");  
        String tirada = teclat.next();  
        if (tirada.matches(REGEXP_TIRADA_VALIDA)) {  
            System.out.println("La tirada és vàlida");  
        }else{  
            System.out.println("La tirada és invàlida");  
        }  
    }  
}
```

**^[A-F][0-9]\$**

## 2. Classes de caràcters

.	Indica qualsevol caràcter.
[abc]	El String deu ser a,b o c
[abc][12]	El String ha de contenir les lletres a b o c seguides de 1 o 2
[^abc]	El símbol ^ dins dels claudàtors indica negació. En aquest cas, indica qualsevol cosa que no siga a, b o c
[a-z1-9]	Indica rang: <b>abcdefghijklmn....z o 12345...9</b>
[0-9a]	Indica qualsevol caràcter entre '0' i '9' i la 'a'
AB	Concatenació. A seguit de B
foo(?!bar)	Qualsevol coincidència amb <i>foo</i> no seguida per <i>bar</i>
foo(?=bar)	Qualsevol coincidència amb <i>foo</i> seguida per <i>bar</i>



## 2. Classes de caràcters. Exemples (I)

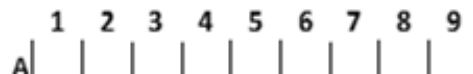
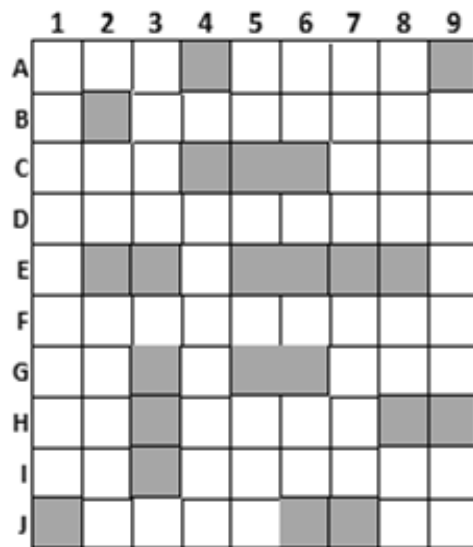
---

- **Exemples:**

- ◆ **a[lm]a** → Coincideix amb ala, ana, ama
- ◆ **co.o** → Coincideix amb coso, como, coro...
- **al[0-9a]** → Coincideix amb al0, al1, al2, al9, ala, ...

## 2. Classes de caràcters. Exemples (II)

- Imaginem que al joc **enfonsar la flota vull** validar que l'usuari introdueix una combinació de cel·la vàlida.
  - A1, A2, A3... A9, ... B1, B2, B3 ... B9, ... J1, J2, J3 ... J9.**



**Puc definir una expressió regular per validar que l'entrada de l'usuari compleix el patró**

Rang  
A,B,C...J

**$^[A-J][1-9]\$$**

Inici de  
cadena

Final de  
cadena

Rang  
1,2...9



### 3. Quantificadors

Els operadors de repetició són: **\***, **+**, **?**

<b>a*</b>	0 o més vegades la 'a'
<b>a+</b>	1 o més vegades la 'a'
<b>a?</b>	1 o cap vegada la 'a'
<b>(abc)+</b>	1 o més vegades la cadena 'abc'
<b>a{4}</b>	la cadena amb la 'a' exactament 4 vegades consecutives (aaaa)
<b>a{2,4}</b>	la cadena amb la 'a' entre 2 i 4 vegades consecutives
<b>a{4,}</b>	la cadena amb la 'a' almenys 4 vegades consecutives

## 4. Alternatives

---

- Podem expressar alternatives mitjançant el **meta-caràcter** "|".
  - **ab|ac**- la cadena 'ab' o la cadena 'ac'
  - **ab(cd|ef)**- la cadena 'abcd' o 'abef'

## 5. Delimitadors o àncores

---

- Les àncores indiquen que la ocurrència ha de produir-se **en llocs especials** dins de la **cadena de cerca**:
  - **\$** final de cadena
    - diar\$**: **Coincideix** amb; **diar**, custodiar
    - No coincideix** amb **diari**
  - **^** :començament de cadena
    - ^diar**: **Coincideix** amb **diari**, **diariament**
    - No coincideix** amb **estudiar**, custodiar

## 6. Classes de caràcters predefinits

- El caràcter de **barra diagonal inversa** **'\'** en una expressió regular indica que el caràcter que segueix és una **classe especial de caràcter**.

<b>\w</b>	Coincideix amb qualsevol caràcter alfanumèric [A-Za-z0-9_]	<b>Ex.</b> a,b,c,0,9,10,
<b>\W</b>	Coincideix amb qualsevol caràcter <b>no alfanumèric</b> [^A-Za-z0-9_]	<b>Ex.</b> %, (,), /
<b>\s</b>	Coincideix amb qualsevol <b>espai en blanc</b>	
<b>\S</b>	Coincideix amb qualsevol <b>espai no en blanc</b>	
<b>\d</b>	Coincideix amb <b>qualsevol digit</b>	<b>Ex.</b> 1,2,3,4,5...
<b>\D</b>	Coincideix amb <b>qualsevol no digit</b>	<b>Ex.</b> w,d,e,r,x,*-



# Activitat Prèvia

---

**Activitat 1.** Què **patrons** tracten de **trobar** les següents expressions regulars?

$^{\text{red}}[A-J]\text{-}^{\text{blue}}[1-9]^{\text{red}}\$$

$^{\text{red}}(1^{\text{blue}}[012]^{\text{red}}|^{\text{blue}}[1-9]^{\text{red}})^{\text{red}}\text{:}[0-5]^{\text{blue}}[0-9]^{\text{red}}\backslash^{\text{red}}s(am/pm)^{\text{blue}}\$$

$^{\text{red}}[a-zA-Z]^{\text{blue}}\backslash^{\text{red}}w\{4,15\}^{\text{red}}\$$

# Activitat Prèvia (solució parcial)

---

- Què **patrons** tracten de **trobar** les següents expressions regulars?

**^[a-zA-Z]\w{4,15}\$**

Un password que ha de:

- Començar amb un caràcter
- Contenir només lletres, números i guions baixos
- Tenir, entre 4 i 15 caràcters (tots dos inclusivament)

## 7. Expressions regulars a Java

### 7.1 La classe String

- La classe `String` presenta mètodes per treballar amb expressions regulars.
  - **public boolean** `matches(String regex)`: permet comprovar si la cadena compleix un patró determinat.

```
public static void main(String[] args) {  
    String cadena = "1212";  
    if (cadena.matches(".*1(?!2).*")) {  
        System.out.println("SI");  
    } else {  
        System.out.println("NO");  
    }  
}
```

## 7. Expressions regulars a Java

### 7.1 La classe String

- **public** String replaceAll(String regex, String replacement): Busca totes les coincidències del patró i les reemplaça per la cadena @replacement
- **public** String replaceFirst(String regex, String replacement): Busca la primera coincidència del patró de i la reemplaça per la cadena @replacement

```
public static void main(String[] args) {  
    // Cadena sobre la qual farem la substitució  
    String cadena1 = "En un lloc de la Manxa";  
    String cadena2 = cadena1.replaceAll("[a|u]", "e");  
    System.out.println(cadena2); // En en lloc de Le menche  
}
```



## 7. Expressions regulars a Java

### 7.1 La classe String

- `public String[] split(String regex)`: Torna un array que conté les parts resultants de dividir l'array atenent el patró especificat per `@regex`.

```
public static void main(String[] args) {  
    String string = "123-654321";  
    String[] parts = string.split("-[a6]");  
    String part1 = parts[0]; // 123  
    String part2 = parts[1]; // 54321  
}
```

# Activitats prèvies

---

**Activitat 2:** Escriu un mètode que valide la introducció d'un número de compte bancari. (0000–0000–00-0000000000)

- Haurà de disposar de:
  - **Codi del banc** que pertany en compte (4 dígit)
  - **Codi de la sucursal** (4 dígit)
  - **Número de control** que impedeix errors de teclat (2 dígit)
  - **Número de compte** bancari (10 dígit)

# Activitats prèvies

---

**Activitat 3.-**Escriviu una expressió regular que valide la introducció d'un NIF (00000000X). X pot valer: TRWAGMYFPDXBNJZSQVHLCKE.

**Activitat 4.** Escriviu una expressió regular que valide la introducció d'un codi postal espanyol (00000).

- Les primeres 2 xifres fan referència a la província (01-52)
- Les tres darreres fan referència al districte i poden anar des de 000 fins a 999

- Això és tot... de moment ;-)