

UF1. PROJECTE

NF1. Programari i codi informàtic

M03. Programació bàsica

Entrega #1.1 – Hèctor Blanco García

Estructura d'un programa informàtic

Blocs d'un programa informàtic

Actualment es comprenen com a blocs, les parts que componen algún concepte, en aquest cas específic, el nostre programa informàtic. En el món de la informàtica es considera que existeixen dos blocs principals que componen el programari:

Els blocs també segueixen una sintaxi o estructura concreta, que sempre comença amb { } i és tancada amb un altre { }. Cada vegada que s'inicia un bloc nou, haurem d'obrir-lo i tancar-lo amb els corxets.

·El bloc de declaracions: Aquell on es detallen tots els objectes que s'empraran al programari (constants, variables, arxius...)

·El bloc d'instruccions: Aquell conjunt d'operacions i accions que s'han de dur a terme per tal que s'aconsegueixin els resultats esperats.

Tipus de dades simples i compostes.

Dades simples: Ocupen una casella de memòria, fa referència a un únic valor a la vegada. Dins d'aquest grup estan: enters, reals, caràcters i booleans.

Dades compostes: També conegudes com dades d'addició, són aquelles que poden almacenar més d'un valor en una comanda. En llenguatges com PHP, tenim, per exemple, el array.

Estructures de selecció, repetició i salt.

Abans de definir les estructures dintre del llenguatge en Java, hem d'especificar la pròpia definició d'una estructura informàtica.

Dintre de Java, una estructura es una compilació d'informació que conté variables i funcions, que permet al software reconèixer i executar un seguit d'instruccions per dur a terme l'acció establerta al codi.

En Java, hi ha diverses estructures que componen tot el seu llenguatge, però les que tractarem en el projecte són:

-Estructura alternativa o selectiva: Funciona d'una forma molt senzilla utilitzant sempre un condicional, és a dir, si es compleix una condició **x**, s'executara una acció, però si es compleix una condició **y**, se n'executarà una altra. Per posar un exemple més quotidià, si plou, no sortirem al carrer i ens mantindrem en casa, mentre que si fa sol, sortirem a

passar en lloc d'estar en casa.

Les instruccions que s'empren en les estructures selectives són: **if** i **else**.

La sintaxi en codi Java quedaria així:

```
if (condició)
    instrucció ;
else
    instrucció ;
```

-Estructura de repetició o loop: Aquesta estructura funciona de manera que repeteix un seguit d'instruccions determinades un nombre de vegades **n** que se li ha assignat al codi. La principal finalitat d'aquesta estructura es no repetir línies de codi innecessàries i fer un programari més compacte. Normalment s'introdueix una condició que si el codi detecta que es compleix, inicia el bucle. Les comandes que s'empren con **while** i **for**. La diferencia entre aquests dos es que en la condició while, s'executa el bucle fins que definim que la condició sigui falsa, ja que si no anul·lem la condició, es realitzarà un bucle infinit. En canvi, en el for, afegeix un increment i un decrement a la variable interposada com a condició i quan el bucle es tanca, la variable desapareix, deixant de formar part en la resta del codi.

Les seves sintaxis son:

```
while (condició)
{
    instruccions ;
}
```

```
for (inicializació ; condició ; increment)
{
    instruccions ;
}
```

-Estructura de salt: Les tres principals estructures de salt a Java són el **continue**, el

break i el **return**. Que son estructures de programari que permeten transferir el control d'una part específica del codi a un altra.

2. Tractament de cadenes.

Una cadena de text no deixa de ser un conjunt de caràcters alfanumerics, signes de puntuació i espais en blanc amb més o menys sentit. L'exemple més clar d'una cadena de text seria "Hola a tots".

En Java, totes aquestes cadenes de text serán representades per la classe `String` i `StringBuffer`.

Per tal de crear una cadena de forma explícita, seguirem la següent estructura:

```
String sMiCadena = new String("Cadena de Text");
```

Encara que si ho volem fer de forma implícita podem emprar la estructura:

```
String sMiCadena = "Cadena de Text";
```

En aquest cas, Java crearia una objecte `String` per tractar aquesta cadena.

Si volem construir una cadena `String` buida, hem d'utilitzar o bé "", utilitzar un `= new String()`.

Si volem volcar una cadena per pantalla emprarem la classe:

```
System.out.println("Cadena de Text");
```

o bé

```
String sMiCadena = new String("Cadena de Text");  
System.out.println(sMiCadena);
```