Entorns de Desenvolupament: M05

Paradigmes de programació

Fernando Porrino Serrano
Presentació elaborada en base al document original d'en Marcel García Vacas
Il·lustració de portada en base a imatges originals extretes de commons.wikimedia.org

Índex

- **Paradigmes de programació**
  - Imperatiu / estructurat
    - Teorema de l'estructura.
    - Disseny descendent.
  - Objectes
    - Abstracció.
    - Encapsulació.
    - Modularitat.
    - Jerarquia.
    - Polimorfisme.
  - Funcional
  - Lògic

Paradigmes de programació

- **Resulta complicat classificar els llenguatges de programació.**
- **Es pot trobar el mateix llenguatge a diverses categories.**
- **Existeix una forma de classificar-los en base a la seva filosofia de base o paradigmes:**
  - Imperatiu/estructurat.
  - Objectes.
  - Funcional.
  - Logic.

Font: pixabay.com

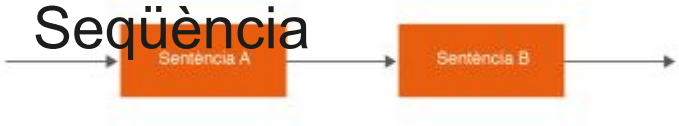
El paradigma imperatiu/estructurat

- **Es basa en donar ordres directes que, al ser executades de forma seqüencial, alteren les dades a memòria.**
- **Exemples de llenguatges:**
  - C
  - Cobol
  - Basic

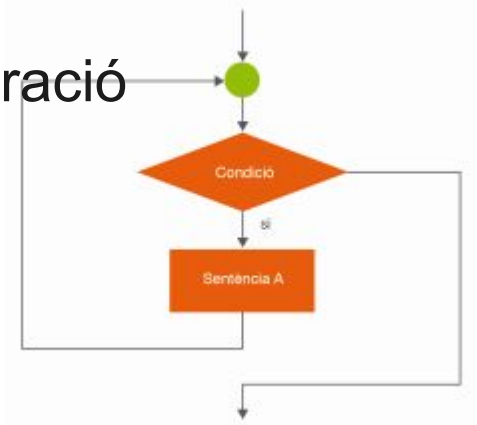
Font: pixabay.com

El paradigma imperatiu/estructurat

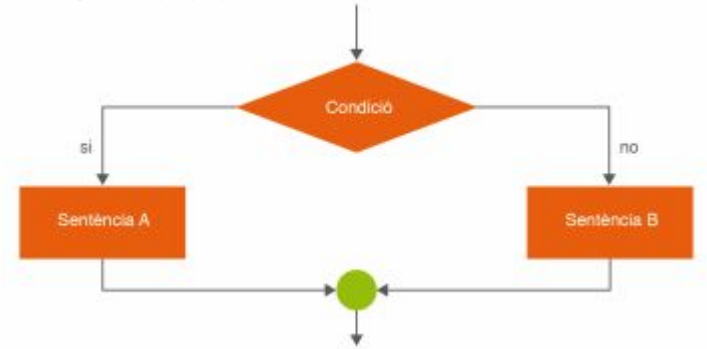
- **Teorema de l'estructura: qualsevol programa pot ser representat mitjançant tres tipus d'estructures de control:**
  - Seqüència



  - Iteració

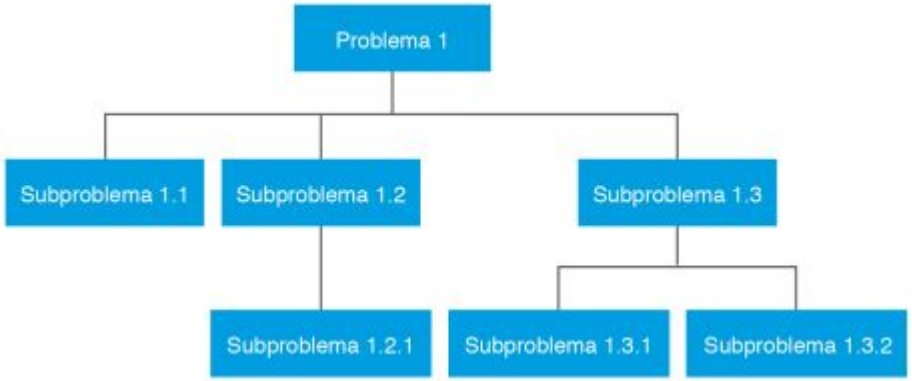


  - Selecció



El paradigma imperatiu/estructurat

- **Tècnica del disseny descendent o "top-down": modular un programa per a que cada peça (mètode) s'encarregui d'una tasca en concret.**



El paradigma imperatiu/estructurat

- **Un exemple:**

```
const float SOU BASE = 1.000;

Struct Administratiu
{
    string nom;
    string DNI;
    float Salari;
}

Struct Professor
{
    string nom;
    string DNI;
    int numHores;
    float salari;
}

void AssignarSalariAdministratiu (Administratiu administratiu1)
{
    administratiu1. salari = SOU_BASE * 10;
}

void AssignarSalariProfessor (Professor professor1)
{
    professor1. salari = SOU_BASE + (numHores * 12);
}
```

El paradigma d'objectes

- **Conegut com a Programació Orientada a Objectes (OOP).**
- **L'abstracció no es basa en procediments sino en objectes.**
  - **Els objectes representen quelcom del mon real.**
  - **Interactuen entre ells amb missatges (crides a mètodes).**

- **Exemples de llenguatges:**
  - C++
  - Java
  - C#

Font: pixabay.com

El paradigma d'objectes

- **Un objecte es pot descompondre en:**
  - Atributs
  - Mètodes
- **Es basa en la integració dels següents conceptes:**
  - Abstracció
    - Classes.
  - Encapsulació
    - Públic, protegit, privat.
  - Modularitat
    - Independència entre components.
  - Jerarquia
    - Herència, associació, composició, agregació.
  - Polimorfisme
    - Sobrecàrrega, sobreescriptura.

Font: pixabay.com

El paradigma d'objectes

- **Un exemple:**

```
class Treballador {
   private:
      string nom;
      string DNI;
   protected:
      static const float SOU_BASE = 1.000;
   public:
      string GetNom() {return this.nom;}
      void SetNom (string n) {this.nom = n;}
      string GetDNI() {return this.DNI;}
      void SetDNI (string dni) {this.DNI = dni;}
      virtual float salari() = 0;
}

class Administratiu: public Treballador {
   public:
      float Salari() {return SOU_BASE * 10};
}

class Professor: public Treballador {
   private:
      int numHores;
   public:
      float Salari() {return SOU_BASE + (numHores * 15);}
}
```

El paradigma funcional

- **Combinació de funcions matemàtiques per reduir-ne la complexitat.**
- **El resultat d'un càlcul és l'entrada del següent fins.**
- **Orientats a l'àmbit de la investigació i origen de les expressions lambda.**
- **Exemples de llenguatges:**
  - Lisp
  - Scala
  - Fulls de càlcul
  - F# (multiparadigma)

Font: pixabay.com

El paradigma funcional

- **Un exemple:**

```
> (defun factorial (n)
(if (= n 0)
1
(* n (factorial (- n 1)))))
FACTORIAL
> (factorial 3)
6
```

El paradigma lògic

- **S'apliquen les regles de la lògica per a inferir conclusions a partir de dades.**
- **Es creen una sèrie de regles o afirmacions (premises) que s'apliquen sobre una col·lecció de dades.**
- **Creat per a treballar amb IA, actualment en desús.**
- **Exemples de llenguatges:**
  - Prolog
  - Elf
  - Godel

Font: pixabay.com

El paradigma lògic

- **Un exemple:**

Exemple de desplegament pràctic del paradigma lògic

Determinarem si hem de prescriure al pacient estar a casa reposant al saber que es compleixen els següents fets: malestar i 39º de temperatura corporal.

Regles de la base de coneixement:

- R1: Si febre, llavors estar a casa en repòs.
- R2: Si malestar, llavors posar-se termòmetre.
- R3: Si termòmetre marca una temperatura > 37º, llavors febre.
- R4: Si diarrea, llavors dieta.

Si seguim un raonament d'encadenament cap endavant, el procediment seria:

```
1  Indicar el motor d'inferència, els fets: malestar i termòmetre
       marca 39.
2
3  <code>Base de fets = { malestar, termòmetre marca º39 }
```

El sistema identifica les regles aplicables: R2 i R3. L'algorisme s'inicia aplicant la regla R2, incorporant en la base de fets "posar-se el termòmetre".

```
1  Base de fets = { malestar, termòmetre marca º39, posar-se termò
       metre }
```

Com que no s'ha solucionat el problema, continua amb la següent regla R3, afegint a la base de fets "febre".

```
1  Base de fets = { malestar, termòmetre marca º39, posar-se termò
       metre, febre }
```

Com que no s'ha solucionat el problema, torna a identificar un subconjunt de regles aplicables, excepte les ja utilitzades. El sistema identifica les regles aplicables: R1, tot incorporant a la base de fets "estar a casa en repòs".

```
1  Base de fets = { malestar, termòmetre marca º39, posar-se termò
       metre, febre, estar a casa en repòs}
```

Com que repòs està a la base de fets, s'ha arribat a una resposta positiva a la pregunta formulada.

Gràcies per la vostra atenció!