



# Continguts

## **Programació procedimental**

- Definició**
- Top\_down anàlisis**
- Exemple**
- Exemple -Procediments**

# P. procedimental - Definició

## Què és la programació procedimental?

Es un estil de programació derivat de la programació estructurada i que es basa en el concepte de procediments que es criden els uns als altres.

*Procedural Programming may be the first programming paradigm that a new developer will learn. Fundamentally, the procedural code is the one that directly instructs a device on how to finish a task in logical steps.*

*This paradigm uses a linear top-down approach and treats data and procedures as two different entities. Based on the concept of a procedure call, Procedural Programming divides the program into procedures, which are also known as routines or functions, simply containing a series of steps to be carried out.*

*Simply put, Procedural Programming involves writing down a list of instructions to tell the computer what it should do step-by-step to finish the task at hand.*

# P. procedural – Top-Down

Per definir els procediments es segueix una aproximació basada en l'anàlisi TOP-DOWN.

1) So, let's say that you want to create a program that plays checkers, but you don't know where to start. Well, when you're working with top-down systems you do.

You just write: **PlayCheckers( );**

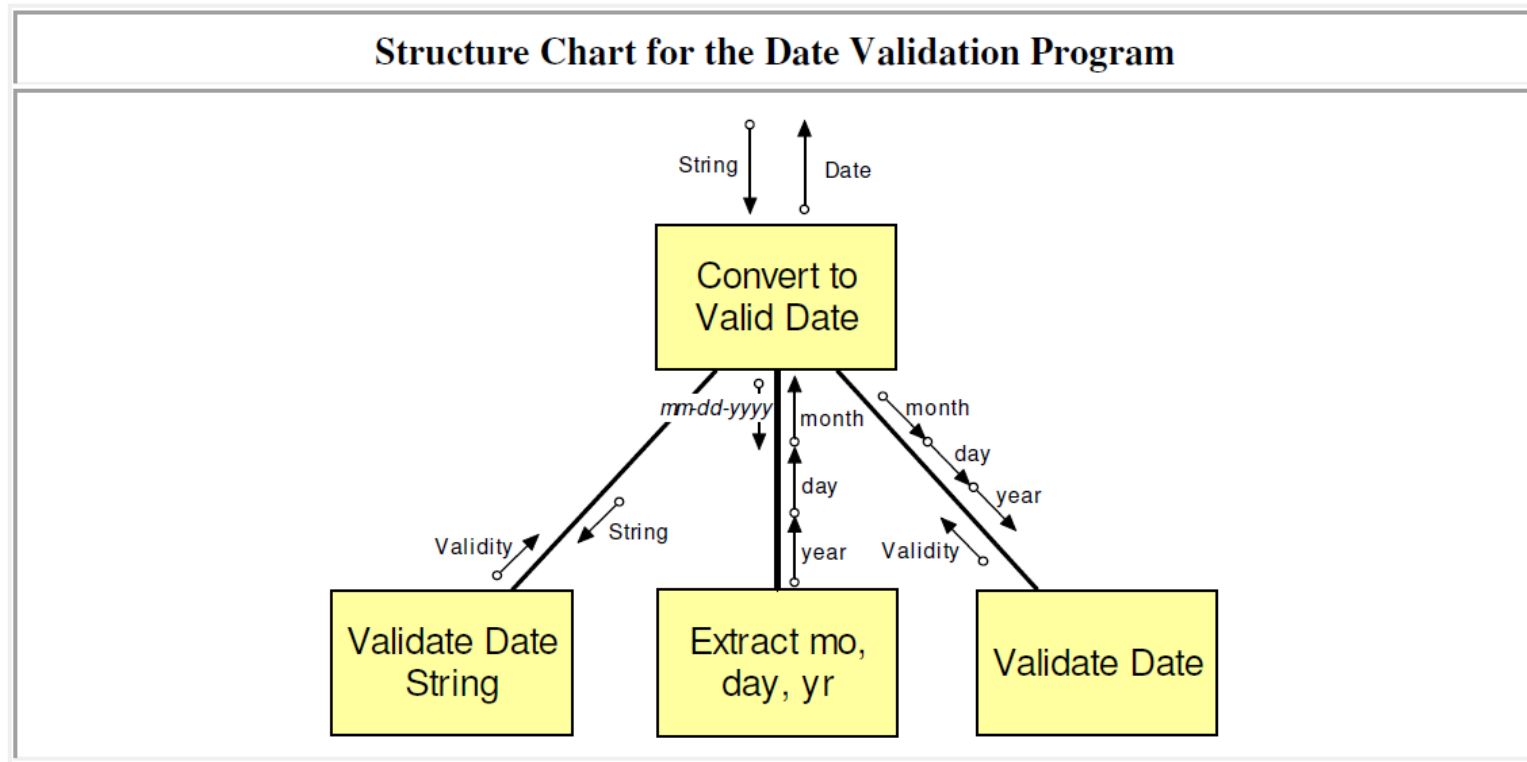
2) Of course, that doesn't mean anything yet, as you don't have a 'play checkers' procedure. But you have defined the problem and can now move to the next level down.

```
PlayCheckers()  
{  
    setupboard();  
    do to end of game {  
        player1Move();  
        player2Move();  
    }  
};
```

# P. procedimental – Exemple (1)

Exemple de programa per convertir un string a data, descomposat en procediments seguint el mètode TOP-DOWN.

- El programa rep com entrada un string i retorna una data



# P. procedimental – Exemple (2)

- La definició d'alt nivell del programa principal és la següent:

## Method Specifications for the Date Validation Program

```
public int [] convertToValidDate( String date )
/* Convert the given string to a date on the Gregorian
 *   calendar.
 * Precondition: none.
 * Postcondition: If the given date is a true Gregorian
 *   date in the form mm-dd-yyyy or mm/dd/yyyy, it returns
 *   an array with a[0]=mm, a[1]=dd and a[2]=yyyy.
 *   It returns null if:
 *       date is null
 *       date doesn't have the indicated form.
 *       mm-dd-yyyy is not a true Gregorian date.
 */
```

- Es tracta d'una definició feta a partir d'indicar **Precondicions**, es a dir condicions prèvies per poder executar el programa.
- Postcondicions** quin serà el resultat d'executar el programa.

# P. procedimental – Exemple (3)

- La definició d'alt nivell dels subprogrames és la següent:

```
public boolean isValidDateFormat( String date )
/* Determine if the given date has valid format;
 *   specifically, ##-##-#### or ##/##/####.
 * Precondition: date is not null.
 * Postcondition: Returns true iff date has valid format.
 */

public int [] extractMoDayYr( String date )
/* Extract integers from the given date representing
 *   its month, day and year.
 * Precondition: date has form mm-dd-yyyy or mm/dd/yyyy.
 * Postcondition: Returns a[0]= mm, a[1]= dd, a[2]= yyyy.
 */

public boolean isValidDate( int month, int day, int year )
/* Determine if month/day/year is a true Gregorian date.
 * Precondition: none.
 * Postcondition: returns true iff month/day/year is a true
 *   Gregorian date.
 */
```

# P. procedural – Procediments (1)

- L'algorisme del programa principal és el següent:

Algorithm for `convertToValidDate`

1. *if* the date string is a null pointer, return null
2. *check* the date string for validity.
3. *if* not valid, return null
4. *extract* the month, day and year.
5. *check* that month, day and year make a real date
6. *if* not, return null
7. *return* month, day, year in an array

# P. procedural – Procediments (2)

- El procediment de validar que l'string que conté la data té un format vàlid podria ser semblant al següent (tenint en compte que estem acceptant formats de data molt concrets)

*Algorithm for isValidDateFormat*

1. check that position 0 is a digit
2. check that position 1 is a digit
3. check that position 2 is a - or /
4. check that position 3 is a digit
5. check that position 4 is a digit
6. check that position 5 is a - or /
7. check that position 6 is a digit
8. check that position 7 is a digit
9. check that position 8 is a digit



# P. procedural – Procediments (3)

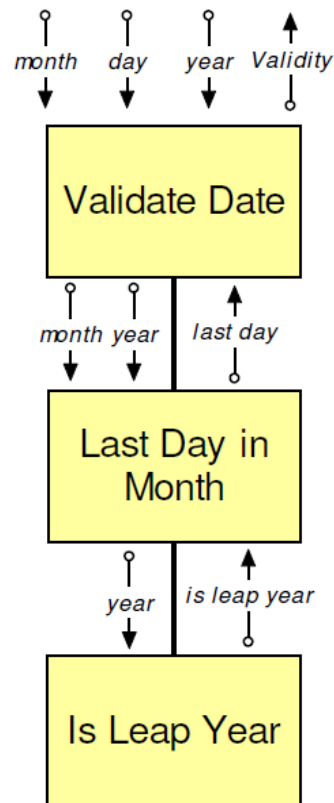
- S'ha comprovat que el format és correcte i se n'ha extret 3 parts: dia, mes i any. El procediment de validar que és tracta d'una data vàlida seria:

Algorithm for isValidDate

1. if year < 1900 then  
    return false  
end if
2. if month is not 1 through 12 then  
    return false  
end if
3. if day is not 1 through whatever is appropriate for the month (and year if month is February) then  
    return false  
end if
4. return true

# P. procedimental – Procediments (4)

- A l'hora de comprovar que es tracta d'una data vàlida s'ha identificat un procediment d'alt nivell «fer el que correspongui segons mes». El podem descomposar en els següents procediments::



L'entrada del procediment principal és un mes, un dia i un any i retorna vàlid/no vàlid

Per ajudar a saber si una data es vàlida hi ha un procediment que donat un mes i un any retorna el darrer dia del mes

Cal tenir present que hi ha anys de traspàs, per això també es defineix un procediment que donat un any retorna si és de traspàs/o no.

# Enllaços de consulta

Procedural programming

[.../procedural-programming/](#)

Python Programming Tutorial with Top-Down Approach

[.../python-programming-tutorial-with-topdown-approach/](#)

Difference between procedural programming and object-oriented programming:

[.../procedural-programming-vs-object-oriented-programming](#)