

<

Paradigmas de Programación

/>

Clase 7

1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 1 0 1

</ Clase 7 - Programación Funcional

- Características de la Programación Funcional.
- Definición de Funciones
- Ejemplos
- Listas
- Operaciones sobre listas
- Ejemplos de listas

</Características de Programación Funcional

- Un programa en este paradigma es un conjunto de definiciones de funciones o ecuaciones matemáticas, cuya evaluación permite obtener el resultado de una consulta.
- El sistema operativo evalúa las funciones y devuelve un resultado a modo de calculadora.

</Características de Programación Funcional

- Las variables no se usan como celdas de memoria cuyo contenido puede variar durante la ejecución del programa, sino como variables matemáticas cuyo valor se liga al argumento de la función y no cambia.
- La programación se realiza a través de scripts.

</Definición de funciones

Se debe definir un script o ambiente en el que se detalla:

- Tipo de la función
- Ecuación que la describe

cuad::Int->Int *tipo de la función*

cuad(x) = x*x *definición o ecuación*



</Tipos de datos

Standard

- Int
- Bool
- Char
- a

Derivado

- Par ordenado (char, num), (num,num), etc.
- Lista: [], [X] [X:Xs], [Xs:Xss]
- Función

No existen las estructuras de control.

- ***Se pueden incluir condiciones en las funciones.***
- ***Para repetir se usa recursión.***

</Formas de definir funciones

Por ecuaciones simples

$\text{doble}(x) = 2 * x$

Por análisis de casos o guardas

$\text{min}(x, y) \mid x < y = x$
 $\mid \text{otherwise} = y$

Con definiciones locales

$\text{potCuarta}(x) = \text{cuad}(x) * \text{cuad}(x)$
 $\text{where } \text{cuad}(x) = x * x$

</Formas de definir funciones

Recursivas

`fact(0) = 1`

`fact(x) = x * fact(x-1)`

De alto orden (composición de funciones) usando:

Map, Filter, Take, Take while

*(Se ven con listas, no se pueden usar en el
parcial).*

</Evaluación de Funciones

Normal: primero se evalúa la función y después se simplifican los argumentos

```
cuad(5 + 4)
(5 + 4) * (5 + 4)
9 * (5 + 4)
9 * 9
81
```

</Evaluación de Funciones

Aplicativo: Primero se simplifican los argumentos y luego se evalúa la función.

```
cuad(5 + 4)
cuad(9)
9 * 9
81
```

Tanto el método normal como el aplicativo llevan al mismo resultado.

</Ejercicios

Eleve a la cuarta un número

</Ejercicios

Eleve a la cuarta un número. Definición
Local

</Ejercicios

Calcule el máximo valor entre 2 números y
retorne 0 si son iguales

</Ejercicios

Dado cualquier valor retorne 5

</Ejercicios

Que indique el signo de su argumento

</Ejercicios

Que devuelve el or exclusivo de sus argumentos