



PROGRAMACIÓN FUNCIONAL

Conceptos Preliminares

Conceptos Preliminares

- ◆ Conceptos de programación y programa
- ◆ Propiedades de los programas
- ◆ Modelo de Computación
- ◆ Sintaxis y semántica
- ◆ Características de la Programación Funcional

Ejercicio

◆ Consigna:

discuta y escriba como definiría los conceptos de *programación* y *programa*, y las características deseables de los programas

◆ Características

- ◆ grupal (3 ó 4 personas)
- ◆ nominal
- ◆ tiempo: 15 minutos

Puesta en común

- ◆ Consigna:

contraste la definición escrita por su grupo con la de otros grupos, intentando obtener una síntesis

- ◆ Características:

- ◆ discusión general
- ◆ tiempo: 15 minutos

Definiciones

◆ Programación:

- ◆ es una tarea que comprende los siguientes puntos
 - ◆ analizar problemas
 - ◆ diseñar soluciones para ellos que puedan ejecutarse
 - ◆ codificar dichas soluciones
 - ◆ verificar propiedades deseadas de las mismas

◆ Programa:

- ◆ descripción de una solución a un problema, que puede ejecutarse de alguna manera para obtener una instancia particular de dicha solución.

Programación

- ◆ ¿Cuáles son los dos aspectos fundamentales?
 - ◆ transformación de información
 - ◆ interacción con el medio
- ◆ Ejemplos:
 - ◆ calcular el promedio de notas de examen
 - ◆ cargar datos de un paciente en su historia clínica
- ◆ Este curso se concentrará en el primero de estos aspectos.

Preguntas

- ◆ ¿Cuáles propiedades de un programa son importantes?
 - ◆ eficiencia
 - ◆ terminación
 - ◆ corrección
 - ◆ equivalencia
 - ◆ claridad
 - ◆ generalidad
 - ◆ modificabilidad
 - ◆ simplicidad
- ◆ ¿En cuáles debería focalizarse un programador?
¿Por qué?

Propiedades

- ◆ Si podemos probar fácilmente equivalencia de programas, podemos
 - ◆ reemplazar un programa por otro más eficiente
 - ◆ usar un programa correcto para ver que otro lo es
 - ◆ ver que no alteramos el significado al modificarlo
- ◆ ¿Qué necesitamos para poder probar equivalencia de programas con sencillez?

Preguntas

- ◆ ¿Cómo saber cuándo dos programas son iguales?
- ◆ Ejemplo:
 - ◆ ¿Son equivalentes ' $f(3)+f(3)$ ' y ' $2*f(3)$ '?
 - ◆ ¿Siempre?
 - ◆ ¿Sería deseable que siempre lo fueran?
¿Por qué?

Ejemplo

- ◆ ¿Qué imprime este programa?

```
Program test;
```

```
var x : integer;
```

```
function f(y:integer):integer;
```

```
begin x := x+1; f :=x+y; end;
```

```
begin x := 0; writeln(2*f(3)); end;
```

- ◆ ¿Y con 'f(3)+f(3)' en lugar de '2*f(3)'?

Preguntas

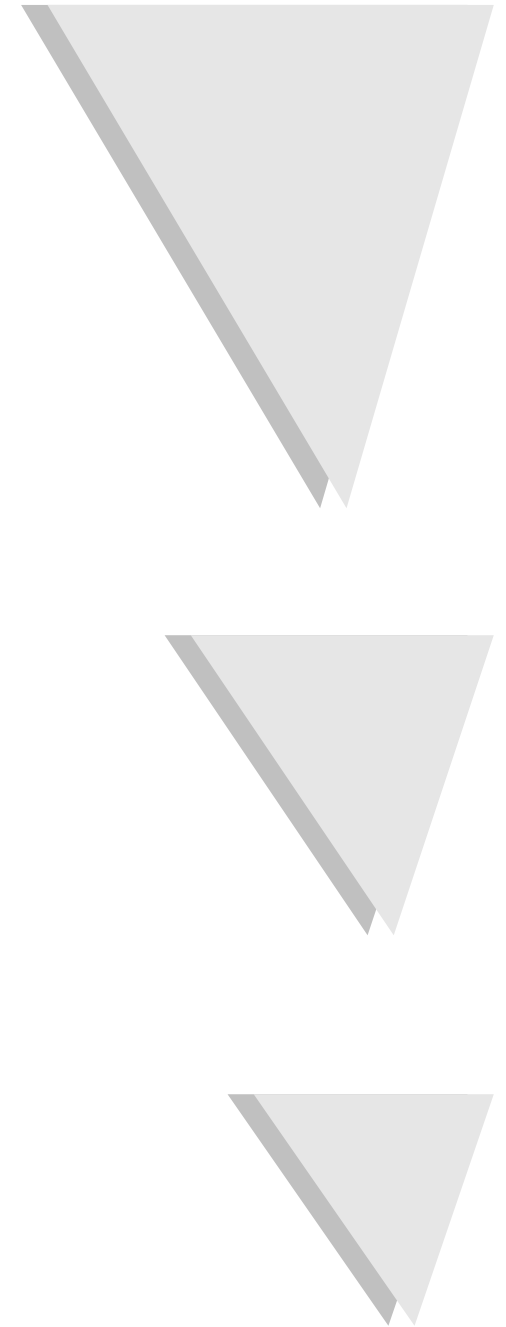
- ◆ ¿Qué conclusiones saca del ejemplo anterior?
- ◆ En particular:
 - ◆ ¿Qué representa una variable?
 - ◆ ¿Qué es estado de un programa?
 - ◆ ¿Qué es flujo de control?
- ◆ ¿Cuál es el *modelo de computación* utilizado?

Definiciones

- ◆ Modelo de computación
 - ◆ representación de la ejecución de un programa mediante un sistema formal
- ◆ Sistema formal
 - ◆ un lenguaje formal (sintaxis) más un aparato deductivo (reglas de transformación)
- ◆ Semántica
 - ◆ asignación de significado a la sintaxis de manera tal que se respeten las reglas del sistema formal

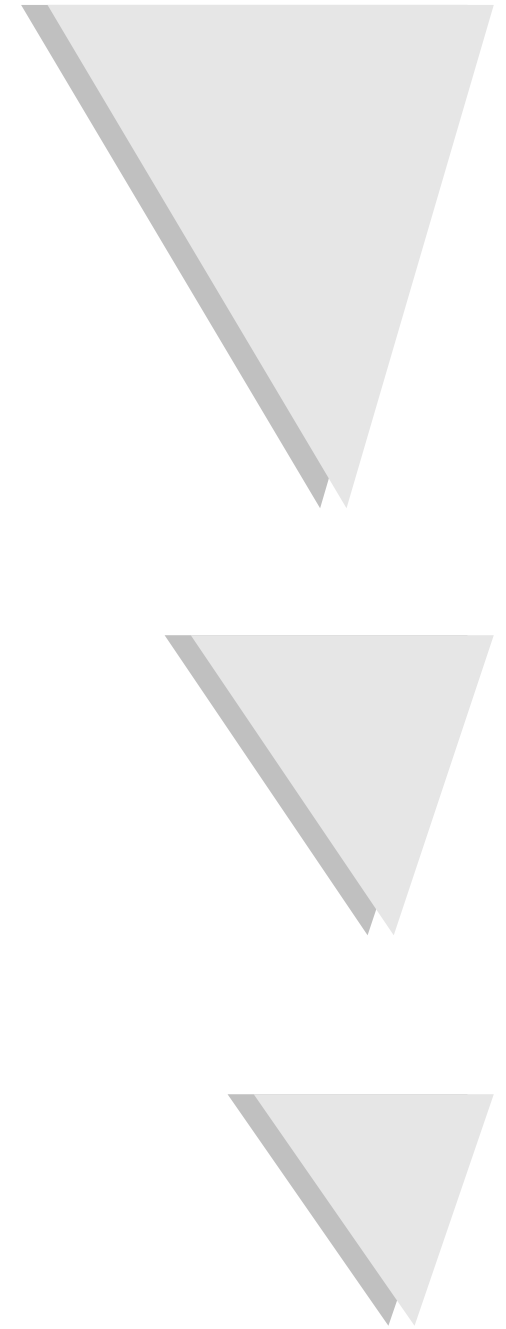
Exposición

- ◆ Razonamiento ecuacional
 - ◆ ¿cómo obtenerlo?
 - ◆ ¿Qué debería representar una variable?
- ◆ Valores y expresiones
 - ◆ TRANSPARENCIA REFERENCIAL
 - ◆ ¡Las funciones también son valores!
 - ◆ Clasificando expresiones: TIPOS



Exposición

- ◆ Programa funcional (script)
- ◆ Definiciones en funcional
 - ◆ Ecuaciones orientadas
 - ◆ Formas de definición
 - ◆ Expresiones especiales
- ◆ Sintaxis especial para ciertos tipos



Exposición

- ◆ Computación:
 - ◆ evaluación, reducción, simplificación
 - ◆ expresiones canónicas o normales
 - ◆ órdenes de reducción
 - ◆ no-terminación
- ◆ Dualidad denotacional/operacional de las expresiones y funciones

