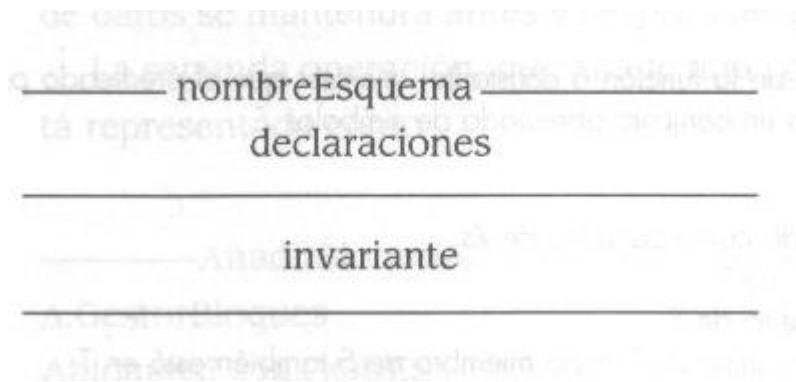


Métodos Formales – Lenguaje Z

Ingeniería de Software I

Lenguaje Z



Esquema: describe un espacio de estados del sistema y las operaciones que sobre el mismo se efectuarán

Es una unidad de texto matemático que describe algunas partes de un sistema.

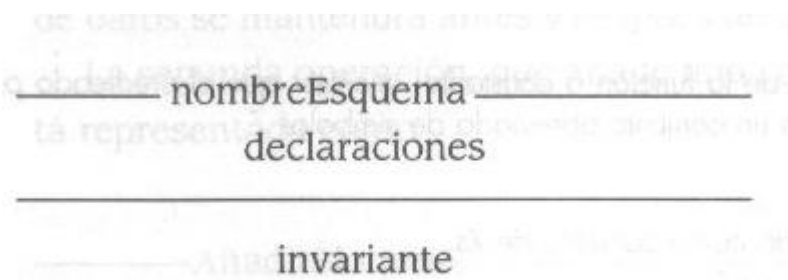
Parte declarativa: variables, tipos de datos

Parte predicativa: proposiciones, relaciones y cambios

Documento de especificación en Z

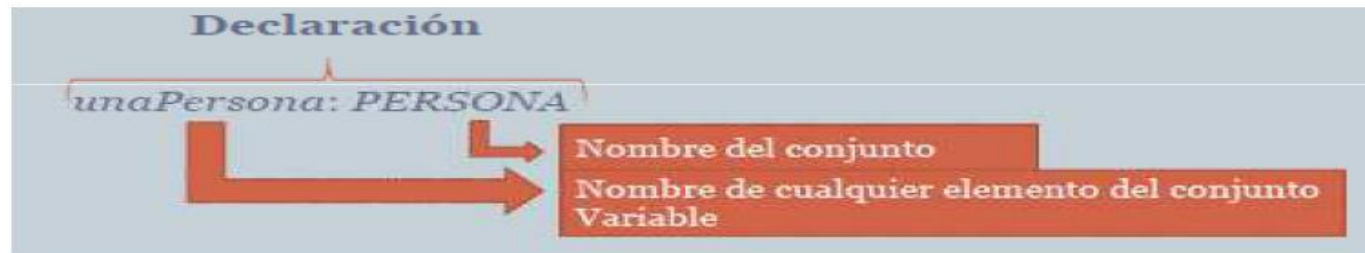
1. Definir los tipos utilizados
2. Definir el estado inicial del sistema
3. Definir las operaciones básicas
4. Definir tratamiento de errores
5. Definir las operaciones completas

Lenguaje Z: declaraciones



Declaraciones

- Sea *PERSONA* el conjunto de todas las posibles personas sobre el Universo. Si queremos referirnos a alguna en particular escribimos:



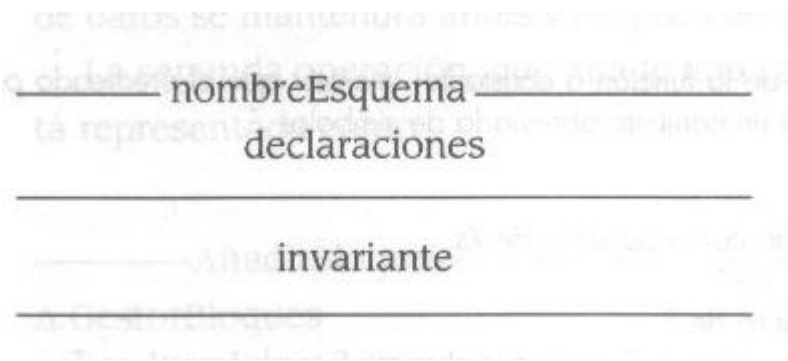
Una **variable** tiene un nombre (*unaPersona*), un tipo (*PERSONA*) y un valor extraído desde el tipo.

Lenguaje Z: declaraciones

Convencion:

- Siempre comenzar con una letra minuscua.
- Escribir como una combinacion de letras mayusculas y minusculas, no usar solo minusculas.
- No debe contener espacios en blanco, underscore, caracteres especiales.

Lenguaje Z: predicados



Predicados

- Los valores de las variables pueden ser restringidos por ***predicados y al satisfacerse un predicado se define un*** conjunto.
- Los tipos de predicados son:
 - =, ∈: igualdad y pertenencia
 - >, ≥, <, ≤ : relaciones
 - ∧, ∨ : conectivos logicos

Lenguaje Z: predicados

Contador

contador: \mathbb{N}

maximo: \mathbb{N}

contador \leq *maximo*

Especificación en Z

1. Definir los tipos utilizados (de datos,)

Tipos Basicos. Introduccion

- Para describir el ambito en el cual trabajamos, clasificamos los objetos en conjuntos llamados **tipos**.
- El unico tipo incorporado como parte de la notacion Z, es el conjunto de los enteros, \mathbb{Z} .
- Todo otro tipo es definido por nosotros.
- Para definir un *tipo basico* debemos centrarnos en lo esencial y los detalles los dejamos de lado. Esta focalizacion se denomina *abstraccion*.

Basicos o dados: depende de los conjuntos utilizados en las aplicaciones que se modelen

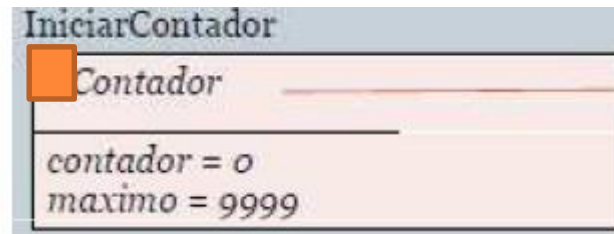
$[Tipo], [Libro, Persona, Autor, Tema]$

- Entre parentesis cuadrados y en orden alfabetico
- Se deben describir claramente. Por ejemplo; NOMBRE es el conjunto de todos los posibles nombre que cualquier persona pueda tener.

Especificación en Z

2. Definir el estado inicial del sistema

- El nombre de un esquema debe comenzar con mayuscula y no puede contener espacios en blanco.
- Si existiera mas de un predicado, estos se acoplan por medio de la conjuncion, actua por defecto.
- El esquema en que se describe las variables de un sistema y los predicados de estas variables se denomina ***esquema del estado del sistema***.
- ***Este representa el estado de un objeto***, la coleccion de valores almacenados en las variables en un instante de tiempo. Sus variables son llamadas ***variables de estado***.



Especificación en Z

3. Definir las operaciones básicas

ConsultaContador

$\exists \text{Contador}$

$\text{contador!} : \mathbb{N}$

$\text{contador!} =$

contador

- (Xi contador) incluye todas las variables y predicados definidos en el esquema Contador; los valores almacenados en estas variables no cambiarán.
- La declaración ***contador! : \mathbb{N}*** que el valor de salida de contador pertenece al conjunto \mathbb{N} . Usamos el símbolo (!) para la salida.
- El predicado ***contador! = contador*** nos dice que la salida del contador es la misma que contador.

Especificación en Z

3. Definir las operaciones básicas

Cambio en el estado del sistema

Click

$\Delta\text{Contador}$
$\text{contador} < \text{maximo}$
$\text{contador}' = \text{contador} + 1$
$\text{maximo}' = \text{maximo}$

Esquema *Click* nos muestra la actualización de la variable *contador* pero la variable *maximo* no tiene cambios.

• $\Delta\text{Contador}$ nos dice:

- Incluye todas las variables y predicados definidos en el esquema Contador.
- Los valores almacenados en alguna (o todas) de estas variables pueden cambiar.
- Toda variable seguida por el signo ' representa el estado despues de que una actualizacion se ha realizado.

Especificación en Z

4. Definir tratamiento de errores

$\frac{\text{Mensaje_Ok}}{\text{salida! : Mensajes}}$ salida! = Ok	$\frac{\text{Error_aniadir_curso}}{\exists \text{Curso}}$ $s? : \text{Estudiante}$ $\text{salida? : Mensajes}$ $(s? \notin \text{matriculado} \wedge \# \text{matriculado} = \text{Limite_clase} \wedge \text{salida!} = \text{sin_sitio})$ \vee $(s? \in \text{matriculado} \wedge \text{salida!} = \text{ya_matric.})$ \vee $(s? \in \text{matriculado} \wedge \# \text{matriculado} = \text{Limite_clase} \wedge \text{salida!} = \text{dos_errores})$
--	--

- Usamos el símbolo (?) para la entrada
- La declaración $s? : \text{Estudiante}$ indica que el valor de entrada pertenece al conjunto de Estudiante

Especificación en Z

5. Definir las operaciones completas

Esquema completo de la operación *Matricular*

$$\textit{Matricular} \equiv (\textit{Matriculado}_0 \wedge \textit{Mensaje_Ok}) \vee \textit{Error_aniadir_curso}$$

Bibliografía

- Métodos formales en la ingeniería del software. Manuel I. Capel.
- Introducción a la notación Z. Maximiliano Cristian.