Departamento de Ingeniería Informática

Calidad en los Sistemas informáticos, 2018-2019

Seminario – Buenas prácticas de sintaxis

Pablo de la Torre Moreno
Departamento de Ingeniería Informática
21 de octubre de 2018

Índice

Conceptos básicos

Notación Pascal
(recomendada por Microsoft®)

```
float NumeroAleatorio();
int edad;
CADENA_DE_CONEXION;
```

Los ejemplos de esta presentación se harán en Pascal

Notación dromedario
(de uso general en las comunidades de software libre)

```
float numeroAleatorio();
int edad;
CADENA_DE_CONEXION;
```

Notación húngara

Consiste en añadir a cada nombre de variable (y sólo para las variables) un prefijo que indique su tipo / semántica.

```
int iEdad;
// en lugar de
int edad;
```

Charles Simonyi
(1948 –)

Sintaxis en modelo conceptual (1 de 2)

- Existen varias propuestas riterios sintácticos [Calero, 2010, cap. 8]
- Parte de la calidad en el desarrollo dirigido por modelos (MDD)
- Obtenidos de los trabajos de Lange, Unhelkar, Wüst, Berenbach y Ambler
- Es aplicable al modelo físico de objetos en base de datos

Sintaxis en modelo conceptual (2 de 2)

Clases	• Nombres únicos en notación Pascal • Multiplicidad, la que representa cada instancia
Atributos	• Sustantivos basados en dominio • Notación del proyecto (Pascal o dromedario) • Tipo y ámbito especificados
Métodos	• Sustantivos si su función es devolver un valor (función) • Verbos si ejecutan una acción (procedimiento) • Notación del proyecto (Pascal o dromedario) • Tipo y ámbito especificados
Multiplicidades	• Especificadas en rango en número, “N” o “*” • Sólo y obligatorio para asociaciones y agregaciones • Si no se indica, se considera 1
Asociaciones y composiciones	• Dibujadas con línea recta • Las cualificadas acaban en flecha • Las composiciones deben tener un diamante

Sintaxis en codificación (1 de 6)

- **Programar preferente –aunque opcionalmente– en inglés.**

- **Utilizar una variación simple de la notación húngara que incluya el ámbito**

Ventajas

- Indicación de tipo en lenguajes de tipificación dinámica (JavaScript, PHP…)
- Reconocimiento visual inmediato (`x = y` *vs.* `fX = iY`)
- Criterio común de denominación de variables

Desventajas

- Hay que acostumbrarse
- Abuso (`a2p2iAge`), según sus detractores, aunque improbable

Sintaxis en codificación (2 de 6)

Ámbito		
_	Privada (un guión bajo)	`_iAge`
__	Protegida (dos guiones bajos)	`__sName`
C	Constante	`_CsConnectionString`
S	Estática	`_Ssingleton`

Clases y tipos		
[nada]	Clase	`Coordinate`
E	Nombre de tipo enumerado	`EErrorType`
I	Nombre de interfaz (`Interface`)	`IPerson`

Sintaxis en codificación (3 de 6)

Variables de iteración		
i	Primer nivel de iteración	**for (int i = 0**
j	Segundo nivel de iteración	**for (int j = 0**
k	Tercer nivel de iteración (más allá, **z**, **t**…)	**for (int z = 0**

Tipo abstracto, puntero o sin signo		
a	Array (en el ejemplo, de enteros [**i**])	**aiAge**
a2	Array de array	**a2Coordinate**
…		
ht	Diccionario (*hash table*)	**htPerson**
qu	Cola (*queue*, en el ejemplo, de enteros [**i**])	**quiAttempt**
st	Pila (*stack*, en el ejemplo, de cadenas [**s**])	**stsDni**
p	Puntero (en el ejemplo, puntero a doble [**d**])	**pdCalification**
u	*Unsigned* (en el ejemplo, *unsigned int* [**ui**])	**uiYear**

Sintaxis en codificación (4 de 6)

Tipo simple, clase o enumerado		
b	Booleano (**`bool`** o **`boolean`**)	**`bIsCorrect`**
c	Carácter (**`char`**)	**`cLetter`**
d	Doble (**`double`**)	**`dAmount`**
e	Valor enumerado (**`enum`** o **`enumerate`**)	**`eErrorType`**
i	Entero (**`int`**)	**`iYear`**
f	Flotante (**`float`**)	**`fDivision`**
l	Entero largo (**`long`**)	**`lIdentifier`**
o	Objeto genérico (**`object`**)	**`oObject`**
r	Entero corto (**`short`**)	**`rLegs`**
s	Cadena (**`string`** o **`String`**)	**`sDescription`**
t	Fecha o fecha y hora (**`DateTime`**)	**`tBirthdate`**
y	Byte (**`byte`**)	**`yValue`**
[nada]	Objeto concreto (suele ser como la clase)	**`coordinate`**

Sintaxis en codificación (5 de 6)

Control		
but	Botón (**Button**)	**butSave**
cal	Calendario (**Calendar**)	**calBithdate**
chk	Chequeo (**CheckBox**)	**chkAuthorized**
cmb	Lista desplegable (**ComboBox**)	**cmbColor**
frm	Formulario (**Form**)	**frmMain**
grp	Grupo (**GroupBox**)	**grpBasicInformation**
lst	Lista (**ListBox**)	**lstColor**
lbl	Etiqueta (**Label**)	**lblTitle**
txt	Caja de texto (**TextBox**)	**txtName**

Más todas aquéllas que se requieran.

Sintaxis en codificación (6 de 6)

- Usar apropiadamente los tabuladores
  - El código de un **bloque** va a un nivel interior
  - El resto de una línea que no cabe va a un nivel interior
    - Se consideran entre 80 y 100 caracteres como un límite adecuado

Sintaxis SQL (1 de 6)

Inserción

```
INSERT INTO Mascota (Id, Id_TipoMascota, Nombre, Peso)
       VALUES (10, 2, 'Arturito', 25.3)
```

- Antes de insertar, recordar que:
  - Las cadenas van entre comillas simples
  - Si una cadena lleva una comilla simple, debe sustituirse por dos
  - El número de valores debe coincidir con el número de campos

Sintaxis SQL (2 de 6)

Actualización

```
UPDATE Mascota
       SET Nombre = 'Arturito', Peso = 26.0
       WHERE Id = 10
```

- Antes de actualizar, recordar que:
  - Las cadenas van entre comillas simples
  - Si una cadena lleva una comilla simple, debe sustituirse por dos
  - Se permiten varias tablas en los datos y en la condición

Sintaxis SQL (3 de 6)

Borrado

```
DELETE FROM Mascota
      WHERE Id = 10
```

- Antes de borrar, recordar que:
  - Si no se incluye la condición, se eliminan todos los registros
  - Se permiten varias tablas en la condición
  - Una vez borrado, no suele haber vuelta atrás
  - ¿Es mejor un campo `DeletionDate` en vez de un borrado físico?

```
UPDATE FROM Pet SET DeletionDate = NOW() WHERE Id = 10
```

Sintaxis SQL (4 de 6)

Consulta

```
SELECT Id, Id_TipoMascota, Nombre, Peso
       FROM Mascota
       WHERE Id = 10
       LIMIT 0, 10
```

- Antes de consultar, recordar que:
  - Las cadenas van entre comillas simples
  - Si una cadena lleva una comilla simple, debe sustituirse por dos
  - No usar el asterisco, para optimizar datos y clarificar código

Sintaxis SQL (5 de 6)

Consulta con múltiples tablas

```
SELECT Mascota.Id Id, TipoMascota.Nombre Tipo,
       Mascota.Nombre Nombre, Mascota.Peso Peso
       FROM Mascota
       INNER JOIN TipoMascota ON Mascota.Id_TipoMascota = TipoMascota.Id
       WHERE Mascota.Nombre LIKE 'A%'
       AND Mascota.Peso >= 100
```

- Antes de consultar, recordar que:
  - Se permiten varias tablas en los campos y en la condición
  - Debe tenerse cuidado con conflictos de nombres

Sintaxis SQL (6 de 6)

INNER JOIN vs SELECT IN

```
SELECT Mascota.Id Id, TipoMascota.Nombre Tipo,
       Mascota.Nombre Nombre, Mascota.Peso Peso, Color.Nombre Color
       FROM Mascota
       INNER JOIN TipoMascota ON Mascota.Id_TipoMascota = TipoMascota.Id
       INNER JOIN Color ON Mascota.Id_Color = Color.Id
       WHERE Mascota.Nombre LIKE 'A%'
       AND Mascota.Peso >= 100
```

```
SELECT Mascota.Id Id, TipoMascota.Nombre Tipo,
       Mascota.Nombre Nombre, Mascota.Peso Peso, Color.Nombre Color
       FROM Mascota
       WHERE Mascota.Id_TipoMascota IN (
              SELECT TipoMascota.Id FROM TipoMascota
              WHERE Mascota.Id_TipoMascota = TipoMascota.Id)
       AND Mascota.Id_Color IN (
              SELECT Color.Id FROM Color
              WHERE Mascota.Id_Color = Color.Id)
       AND Mascota.Nombre LIKE 'A%'
       AND Mascota.Peso >= 100
```

Referencias

- [Calero, 2010] Calero, C., Moraga, M. A., Piattini, M. G., *Calidad del producto y proceso software*, Ra-Ma, 2010
- [Ambler, 2005] Ambler, S. W., *The elements of UML 2.0 style*, Cambridge University Press, 2005
- [Berenbach, 2004] Berenbach, B., *Evaluation of large, complex UML analysis and design models*, 26th International Conference on Software Engineering (ICSE'04), 232-241
- [Lange, 2007] Lange, C., *Assessing and improving the quality of modeling*, Tesis doctoral, Universidad de Eindhoven
- [Unhelkar, 2005] Unhelkar, B., *Verification and validation for quality of UML 2.0 models*, John Wiley & Sons, Inc.
- [Wüst, 2005] Wüst, J., *The software design metrics tool for the UML* (http://www.sdmetrics.com)