

Convenciones y reglas usadas en la codificación

A continuación se describen el conjunto de reglas que hay que seguir estrictamente a la hora de implementar el código de la aplicación. Es importante seguir estas reglas para que después no haya confusión entre los miembros del subgrupo de implementación. Estas reglas pueden ser susceptibles a cambios en caso de que haya desacuerdos y se quiera redefinir la forma de hacer las cosas.

Paquetes

- Por lo general, los nombres de los paquetes deben empezar por mayúscula. Si el nombre está compuesto por varias palabras, todas las palabras se escriben juntas, empezando la primera letra de cada palabra por mayúscula; y además, se omiten los artículos y las preposiciones.

Ejemplos:

1. El paquete para “el centro medico” → “CentroMedico”, ...

Clases

- Por lo general, los **nombres de las clases** deben empezar por mayúscula. Si el nombre está compuesto por varias palabras, todas las palabras se escriben juntas, empezando la primera letra de cada palabra por mayúscula; y además, se omiten los artículos y las preposiciones.

Ejemplos:

1. La clase para “el paciente” → “Paciente”
2. La clase para “la capa de base de datos” → “CapaBaseDatos”, “BaseDatos”, ...
3. La clase para “el punto y la recta” → “PuntoRecta”, “PuntoYRecta”, ...

EXCEPCIÓN: Para las clases relacionadas con la interfaz de usuario, el nombre debe comenzar por “UI_” y a continuación el nombre como se ha descrito antes.

Ejemplo:

1. La clase para la “interfaz de usuario del administrador” → “UI_Administrador”, ...

EXCEPCIÓN: Para las clases relacionadas con la interacción entre el mundo de los objetos y la base de datos, el nombre debe comenzar por “BD_” y a continuación el nombre como se ha descrito antes.

Ejemplo:

1. En el mundo de los objetos tenemos la clase “Paciente”, y para que este se comuniquen con la base de datos debe comunicarse con una clase intermedia, y esta ya se comunica con la capa de la base de datos. Un nombre sugerido para esa capa intermedia puede ser: “BD_Paciente”.

Así, el esquema de comunicación entre clases puede ser:

“Paciente” ----> “BD_Paciente” ---> “CapaBaseDatos”

- Si el **nombre de una clase** está compuesto por varias palabras, por lo general debe generarse su nombre de forma que muestre la mayor información posible, **pero** si el nombre generado resulta ser muy largo, algunas palabras se pueden sustituir por su inicial en mayúscula.

Ejemplo:

1. La clase para “la capa de base de datos” → “CapaBaseDatos”, “BaseDatos”, ...
Estos nombres no son muy largos, por lo que no habría que reducirlo en iniciales.
2. La clase para “el concursante de la casa del Gran Hermano” → “ConcurstanteCasaGH”
Este nombre ya es mas largo, y por eso se reducen a iniciales 2 palabras.

- Los **nombres de los miembros de una clase** (los miembros de la clase son las variables privadas de la clase) empiezan en minúscula, y si es un nombre compuesto, se juntan las palabras y cada palabra empieza por mayúscula (excepto la primera). Se omiten los artículos y preposiciones del nombre.

Ejemplo:

1. Si la clase "Paciente" tuviese los miembros para el dni, el nombre, apellidos, teléfono móvil, etc ... :
 dni → "dni"
 nombre → "nombre"
 apellidos → "apellidos"
 teléfono móvil → "telefonoMovil"

- Al igual que con los nombres de las clases, si el **nombre de un miembro de la clase** está compuesto por varias palabras, por lo general debe generarse su nombre de forma que muestre la mayor información posible, **pero** si el nombre generado resulta ser muy largo, algunas palabras se pueden sustituir por su inicial en mayúscula.
- Los **nombres de las funciones miembro de una clase** empiezan en mayúscula, y si es un nombre compuesto, se juntan las palabras y cada palabra empieza por mayúscula. Se omiten los artículos y preposiciones del nombre.

Ejemplo:

1. Si la clase "Paciente" tuviese las funciones miembro para obtener el nombre, obtener el dni, consultar el historial, etc ... :
 obtener el nombre → "ObtenerNombre", "getNombre", ...
 obtener el dni → "ObtenerDni", "getDni", ...
 consultar el historial → "Historial", "ConsultarHistorial", "GetHistorial", ...

- Al igual que con los nombres de las clases, si el **nombre de una función miembro de la clase** está compuesto por varias palabras, por lo general debe generarse su nombre de forma que muestre la mayor información posible, **pero** si el nombre generado resulta ser muy largo, algunas palabras se pueden sustituir por su inicial en mayúscula.
- A la hora de definir una función miembro, los **nombres de los parámetros de las funciones miembro (al igual que los nombres de las f. miembro)** empiezan en mayúscula, y si es un nombre compuesto, se juntan las palabras y cada palabra empieza por mayúscula. Se omiten los artículos y preposiciones del nombre.

Ejemplo:

1. Para una función que devuelva un paciente y haya que pasarle un identificador de paciente, y si el identificador fuese el dni del paciente:
 "Paciente ObtenerPaciente(String Dni)", ...
- Al igual que con los nombres de las clases, si el **nombre de un parámetro de una función miembro de la clase** está compuesto por varias palabras, por lo general debe generarse su nombre de forma que muestre la mayor información posible, **pero** si el nombre generado resulta ser muy largo, algunas palabras se pueden sustituir por su inicial en mayúscula.

Funciones

- Los **nombres de las variables locales de las funciones** (al igual que con los miembros de una clase) empiezan en minúscula, y si es un nombre compuesto, se juntan las palabras y cada palabra empieza por mayúscula (excepto la primera). Se omiten los artículos y preposiciones del nombre.
- Al igual que con los nombres de las clases, si el **nombre de una variable local** está compuesto por varias palabras, por lo general debe generarse su nombre de forma que muestre la mayor información posible, **pero** si el nombre generado resulta ser muy largo, algunas palabras se pueden sustituir por su inicial en mayúscula.