

Coding Standards

DevInt 23

1 Cabeceras

- **#define guard**

- * Todos los archivos .h deben tener define de guard declarados en mayúsculas.
- * Deben tener la terminación de "_H_" de la siguiente manera:

Ejemplo:

```
#ifndef ACTIONPACMAN_H_
#define ACTIONPACMAN_H_
..
#endif ACTIONPACMAN_H_
```

- **#define de variables**

- * Deben estar declarados en mayúsculas.
- * Deben estar declarados dentro del archivo Defines.h .
- * Si son palabras compuestas, cada palabra debe estar separada por barra baja (_).

Ejemplo:

```
#define NEURONAL_NETWORK "a "
```

- **Orden de las cabeceras**

1. Deben declararse las librerías propias.
2. Deben declararse los includes de la librería de C++.
3. Deben declararse todos los demás includes.
4. Deben declararse todos los usings.

- **#include archivos .h**

- * Todos los includes deben declararse entre los símbolos de mayor y menor "<>" .
- * En el caso de los que sean de la librería de C++, no deben contener el .h.

Ejemplo:

```
#include <NewbieParser.h>
#include <cstdio>
#include <cstring>
using namespace std
```

2 Clases

- Nombres de clases

- * El nombre debe ser un sustantivo.
- * La primera letra del nombre debe ser mayúscula y el resto minúscula.
- * En caso de que el nombre de la clase sea compuesto la primera letra de la siguiente palabra debe ser mayúscula.

Ejemplo: `class NewbieParser`

- * No usar barra baja (`_`) ni otros símbolos (`*`, `"`, `-`, `\`, *etc.*).

- Nombres de variables

- * Globales

- Al inicio del nombre de la variable se debe colocar una barra baja (`_`).
 - La primera letra del nombre debe ser minúscula.
 - En caso de que el nombre de la variable sea compuesto la primera letra de la siguiente palabra debe ser mayúscula.

Ejemplo: `string _nameTrainer`

- * Locales y parámetros

- La primera letra del nombre debe ser minúscula.
 - En caso de que el nombre de la variable sea compuesto la primera letra de la siguiente palabra debe ser mayúscula.

Ejemplo: `string nameTrainer`

- Nombres de métodos

- * El nombre del método debe reflejar una acción .
- * La primera letra del nombre debe ser mayúscula.
- * En caso de que el nombre del método sea compuesto la primera letra de la siguiente palabra debe ser mayúscula.

Ejemplo: `void GetParser()`

- * No usar barra baja(`_`) ni otros símbolos.

3 Formato

- Idioma

- * Tanto las cabeceras, clases, variables y métodos, deben estar en inglés.

- Indentación

- * Indentar con cuatro espacios.
- * No usar tabs en el código. Reemplazar la tecla TAB, por cuatro espacios en el editor de su preferencia .

- Llaves

- * Las llaves deben ser el primer y único carácter de su línea.

- Espacios

- * Debe existir un espacio entre una variable y su operador.
- * Entre la variable y el punto y coma no debe existir espacio.

Ejemplo: `string nameTrainer = "primus";`

- Declaración de punteros y referencias

- * Se debe poner los símbolos de punteros y referencias(* , &) a lado derecho del tipo de la variable, sin espacio entre estos.

Ejemplo: `Newbie* newbie = new Newbie();`
`string& a = b;`

- Comentarios

- * Los comentarios deben realizarse solamente para describir el código y deben ser claros.
- * Dentro de los comentarios no debe existir código, a menos que sea estrictamente necesario para describir el código.