Guidelines:Coding_Documentation_es

Lea también las normas de estilo de programación

Para la documentación del código, nos valemos de la herramienta Doxygen, por ser multilenguaje, multiplataforma y bastante común entre proyectos de Software libre.

La mejor manera de comenzar es leyendo el manual de Doxygen de la página web, en concreto la sección de documentación del código es la más útil, pudiendo obviar el resto. Sin embargo, debido a la flexibilidad de Doxygen, aquí se resume el estilo que deben de seguir para que los comentarios del código y la documentación generada sea consistente.

Documentación básica

Asegúrese de haber leido antes la documentación de Doxygen.

- **La documentación debe situarse en los ficheros de código, no de cabecera**.
  La documentación debe de añadirse siempre que sea posible en los ficheros de código, no en los de cabecera. De este modo se mantienen las cabeceras limpias y pequeñas. *Excepciones a esta norma* son los espacioes de nombre, las clases y los miembros de las clases públicos y privados, que pueden documentarse en la cabecera.
- **Se usa el estilo de código de C++**
  La documentación debe hacerse con comentarios con el estilo de C, esto es, entre `/*` y `*/`. Los comentarios de C++ de una sola linea con el símbolo `//`, aunque válidos, no son correctos para hacer nuestra documentación.
- **Uso del estilo de Javadoc**
  Para una sintaxis más clara, el estilo es el mismo que Javadoc, así que si se está familiarizado con este sistema de documentación, el uso de Doxygen será más sencillo. **Tenga en cuenta al leer la documentación de Doxygen que dado que se usa el formato de Javadoc, las órdenes comienzan con `@` y no con `\`.**

Normas comunes

Para cualquiera de los bloques de documentación se usan la siguientes normas.

- Comienza un bloque de código con `/**` y un salto de linea.
- Comienza cada bloque del código con un *. Estos símbolos deben de alinearse al primera * del comienzo del comentario.
- La segunda linea del comentario debe ser una descripción breve del termino a documentar. **Debe de terminar en un punto para que la descripción sea procesada correctamente**.

- Termina la descripción breve con una linea en blanco.
- A continuación, de forma opcional una descripción más detallada.
- Se cierra el comentario en una nueva linea.

Ejemplo:

```
/**
 * A variable. Just it.
 *
 * A more elaborate description of the variable. It may
 * contains a large amount of lines, so please don't
 * put the description in a single line.
 */
int var;
```

Funciones

Para comentar una función, hay que tener en cuenta, además:

- Justo después de la descripción breve (o la detallada si se ha incluido) hay que describir los parámetros. Doxygen utiliza la marca `@param`, seguida del nombre del parámetro y la descripción del mismo.
- Después del último comentario una linea en blanco.
- Por último, con la marca `@return` se describe el valor devuelto.
- Cualquier otro comando de Doxygen irá después de este último, añadiendo justo antes una linea de comentario en blanco.

Ejemplo:

```
/**
 * A normal member taking two arguments and returning an integer value.
 *
 * This is an incredible function that will make people happier and stop
 * wars in the world.
 *
 * @param a an integer argument.
 * @param s a constant character pointer.
 *
 * @return The test results.
 *
 * @exception MyException throwed if any error happens.
 * @see otraFuncion
 */
int
testMe (int a, const char *s) {
   ....
}
```

Clases

Las clases en los lenguajes de programación orientados a objetos, deben de documentarse en su declaración, es decir, en los ficheros de cabecera. Los atributos y funciones **privados no deben documentarse**, los **públicos y protegidos** sí, y debe de hacerse **en el fichero fuente**. Ejemplo:

```
/**
 * This is a class to manage something.
 *
 * Many things can be done with an object of this class.
 */
class MiClase {
private:
       int atr;
       int f ();
protected:
       int bbb;
       int g ();
public:
       char *id;
       char *getId ();
};
```

Tipos estructurados

La declaración de una estructura o un enumerado, deben de documentarse de la siguiente manera.

- Al principio de la definición se añadirá la documentación sobre el tipo.
- Cada campo o valor se documentará únicamente con un comentario breve, a menos que sea absolutamente necesario.
- Para que la declaración sea más legible y que los comentarios no estorben para ello, se documentará con el formato de Doxygen para los elementos anteriores al comentario. Es decir, el comentario se pondrá en la misma linea que la definición del campo, con el formato `/**<` y `*/`.

Ejemplo:

```
/**
 * A struct that represents a car.
 *
 * The structs can have a long description. The fields should not.
 */
struct MyStruct {
     int wheels; /**< Number of wheels. */
     char *id;   /**< Car ID. */
     int cv;     /**< Motor power in CV. */
}
```

Ejemplo de enumerados:

```
/**
 * Numbers enumerator.
 */
typedef enum {
      CERO, /**< Zero */
      UNO,  /**< One */
      DOS   /**< Two */
} numeros;
```