**Definición de un estándar para contar líneas de código (LOC)**

Objetivos del ejercicio:

1. Definir el estándar para contar LOC de lenguaje que estas usando.
2. Proveer las bases necesarias para desarrollar un estándar de codificación
3. Preparar lo necesario para hacer un programa que cuente LOC

En práctica debes Producir un estándar para contar LOC lógicas para el lenguaje de progracaión que estas usando en tus proyectos de PSP.

El resultado esperado es un estándar documentado usando las siguientes tablas, que discutimos en clase,

LOC COUNTING STANDARD TEMPLATE

Definition Name: C++ Código standard Language: C++

Author: Adrian Jesus Lora Molina Date: 17-06-21

|  |  |  |
| --- | --- | --- |
| **Count Type** | **Type** | **Comments** |
| Physical/Logical | Lógico |  |
| **Statement Type** | **Included** | **Comments** |
|  |  |  |
| Executable | Si |  |
| Nonexecutable | No |  |
| Declarations | Si | Declaraciones de variables, etc. |
| Compiler Directives | No | Particularmente por los problemas de PSP, esto es simple. Donde agrego “include guards” en cada archivo de C++, agregando las directivas de compilador aumentara los recuentos de LOC de manera inaceptable |
| Comments | No | No apropiado—No se añade complejidad. Ningún comentarios debería añadirse al LOC |
| On own lines |  |  |
| With source |  |  |
| Banners |  |  |
| Blank lines |  |  |
| **Clarifications** |  | **Examples/Cases** |
| Nulls | No |  |
| Empty statements | No | “;;”, lone ;’s, etc. |
| Begin…end | No | Misma lógica que en C++. Begin/end |
| Test conditions | No |  |
| Expression evaluation | No |  |
| End symbols | No | Como fue mencionado—no se delimitará declaraciones ; , no se contarán |
| Then, else, otherwise | No |  |
| Elseif | Si | “else if” será considerado como una declaración |
| Keywords | Si | Palabras claves solitarias (inspect, from, until, loop, etc) |
| Labels |  | Branch destinations when on separate lines |

|  |  |
| --- | --- |
| Propósito | Guiar el desarrollo de los programas de C++. Antes de recuento, los archivos de origen deben ser procesadas por sangría GNU con opciones predeterminadas lo que hará cumplir muchas de las reglas delineadas. |
| Contenido de archivo | Un archivo de origen y un archivo de encabezado por clase. Clases que pueden ser completamente descritas en un archivo encabezado no requieren un archivo de código. Funciones cuales están autónomas no requieren un archivo de encabezado separado. |
| Comentarios (general) | Cuando se usan los comentarios debería describir brevemente la intención del código en lugar de su función interno. El énfasis debe ser puesto en la claridad del código; si unas secciones del código parece que necesita una explicación para que funcione debería reestructurar para simplificar |
| Comentarios de clase | Cada clase debería tener una explicación breve descripción en el encabezado, preferiblemente una o dos líneas de comentarios antes de la clase |
| Comentarios de métodos/funciones | Cada método y función debe tener una sola línea en la declaración de clase que describe brevemente su intención. |
| Comentarios bloqueados | Los comentarios bloqueados pueden usarse con C++, pero ningún código debe existir en la misma línea que un comentario de bloqueo. |
| Declaración de método/función | Se declarará el tipo de valor return en la línea antes de la declaración de método/función |
| Contractos | Cada método/función que tiene restricciones operativas debería reflejarse en las macros REQUIRE, ENSURE y CHECK, usados desde el paquete yakIcons *contract.cpp,* ie. REQUIRE (num\_elements > 1) |
| Identificadores | Identificadores deberán describirse extensivamente. Todos los identificadores deben estar en minúsculas, ie number\_of\_entries, en lugar de NumberOfEntries. Solo puede haber un identificador por |
| Nombres de descriptores de acceso | Nombres de descriptores de acceso deben representar el valor, usualmente representado como sustantivo, ie “standard\_deviation” en lugar de “get\_standard\_deviation” |
| Tamaño del método | Métodos deben tener una obvia o documentada razón para ser mas largos que 35 LOC físicos. |
| Llaves (begin/end) | Llave abiertas y cerradas van a estar en una línea separada y con sangría de un nivel |