Assignment Kit for Size Counting Standard

Personal Software Process for Engineers: Part I

The Software Engineering Institute (SEI)
is a federally funded research and development center
sponsored by the U.S. Department of Defense and
operated by Carnegie Mellon University.

Personal Software Process for Engineers: Part I

Assignment Kit for the Size Counting Standard

Overview

Overview

This assignment kit covers the following topics.

Prerequisites

Prerequisite reading
- Chapter 3

Objectives

The objectives of Size Counting standard are to
- define the size counting standards that are appropriate for the programming language and environment that you will use
- provide a basis for developing a coding standard
- prepare for developing a program to count program size

Size counting standard requirements

Size counting standard requirements

Produce and document a standard for counting program size for the language and environment that you will use in this course.

Submit the completed standard using the format in the template on page 11 with your program 2 assignment package.

Size Counting Standard Template

Definition Name:	Conteo de tamaño para programas Java (Programa 4a – Integración Simpson y distribución t)	Language:	Java
Author:	Jared Fernández González	Date:	30/11/2025

Count Type	**Type**	**Comments**
Physical/Logical	Logical LOC (líneas lógicas de código)	Se cuenta el número de sentencias lógicas Java. Una sentencia termina normalmente en ; o es un bloque de control completo (if, for, while, switch, do, try, catch, finally, return, throw). Una sentencia puede ocupar una o varias líneas físicas, pero se cuenta como 1. Varias sentencias en una misma línea cuentan como más de 1.
Statement Type	**Included**	**Comments**
Executable	Sí	Se incluyen todas las sentencias que ejecutan acciones: asignaciones, llamadas a métodos, return, throw, incrementos/decrementos, sentencias dentro de if, for, while, switch, etc. Por ejemplo, dentro de computeFinalValue, cada llamada a computeW(...), computeXi(...), computeFxi(...), y cada suma sobre dblFinalTerms cuenta como sentencias ejecutables.
Nonexecutable:		
Declarations	Sí	Se incluyen declaraciones de variables, atributos, constantes y firmas de métodos/constructores (por ejemplo: int intNumSeg;, private double dblFinalValue;, public double computeFinalValue(int intNumSegIn) { ... } cuenta la firma del método como 1 sentencia lógica).
Compiler Directives	No	Las sentencias package e import no se cuentan en el tamaño.
Comments	No	No se cuentan líneas de comentarios ni bloques de comentarios (//, /* ... */, /** ... */), aunque contengan código comentado. Por ejemplo, los comentarios que documentan GammaFunction o SimpsonIntegration no se cuentan.
Blank lines	No	No se cuentan líneas en blanco. Solo se usan para mejorar la legibilidad entre secciones (por ejemplo, entre atributos y métodos públicos en SimpsonIntegration).
Other elements	No	No se cuentan líneas que solo tienen llaves { o }, ni separadores, ni anotaciones aisladas. Las anotaciones (@Override, etc.) no se cuentan por sí solas; la sentencia que siguen es la que se cuenta.
Clarifications		**Examples/Cases**
Una sentencia puede ocupar varias líneas físicas pero cuenta como 1 línea lógica.		java\n// Ejemplo similar a un cálculo en SimpsonIntegration\n dblFinalValue = (dblW / 3.0)\n * dblSum;\n Esto cuenta como 1 sentencia lógica (una asignación).

Varias sentencias en la misma línea cuentan por separado.		java\nint intNumSeg = 10; dblW = dblX / intNumSeg; System.out.println(dblW);\n Aquí hay 3 sentencias lógicas: declaración/asignación de intNumSeg, asignación de dblW, y llamada a println.
Las estructuras de control se cuentan como una sentencia lógica por cada cabecera de control.		java\nif (intNumSegIn <= 0) {\n throw new IllegalArgumentException(\"El número de segmentos debe ser positivo.\");\n}\n El if (...) cuenta como 1 sentencia lógica. El throw ...; cuenta como otra sentencia lógica.
Las firmas de métodos y constructores se cuentan como una sentencia lógica cada una.		java\npublic SimpsonIntegration(int intDOFIn, double dblXIn)\n{\n this.intDOF = intDOFIn;\n this.dblX = dblXIn;\n}\n La firma public SimpsonIntegration(int intDOFIn, double dblXIn) cuenta como 1 sentencia lógica. Dentro del constructor hay 2 asignaciones, que son 2 sentencias lógicas más.
Las declaraciones múltiples cuentan como una sentencia por cada variable.		java\nint i, j, k;\n Esto se cuenta como 3 sentencias lógicas de declaración (una por cada variable). Si en el futuro decides tratarlas como 1 sola, deberás actualizar este estándar y mantener la regla siempre igual.
Los bucles for, while y do-while cuentan 1 por la cabecera del bucle más las sentencias del cuerpo.		java\nfor (int i = 0; i <= intNumSeg; i++)\n{\n dblSum += dblFinalTerms[i];\n}\n El for (...) se cuenta como **1** sentencia lógica. La asignación acumulativa dblSum += ...; es otra sentencia lógica.
Note		Este estándar se aplica a todas las clases del proyecto 5a: App, Output, GammaFunction, SimpsonIntegration, Logic y cualquier clase nueva que se añada al programa 5a.
Note		
Note		
Note		

