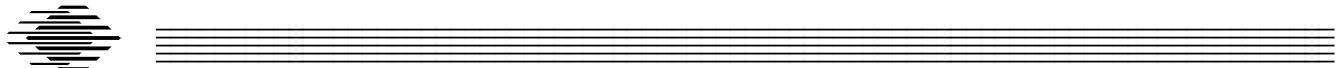


# **Assignment Kit for Size Counting Standard**



## **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.

This material approved for public release.  
Distribution is limited by the Software Engineering Institute to attendees.

# Personal Software Process for Engineers: Part I

## Assignment Kit for the Size Counting Standard

### Overview

---

#### Overview

This assignment kit covers the following topics.

| Section                                   | See Page |
|---|----------|
| Prerequisites                             | 2        |
| Objectives                                | 2        |
| Size counting standard requirements       | 3        |
| Example 1: Pascal size counting standard  | 4        |
| Example 2: C++ size counting standard     | 6        |
| Example 3: another size counting standard | 8        |
| Evaluation criteria and suggestions       | 10       |
| Size counting standard template           | 11       |

---

#### 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 Language: Java  
 Java (Regresion Lineal)

Author: Jared Fernández González Date: 27/11/2025

| Count Type  | Type                                   | Comments   |
|---|--|--|
| Physical/Logical  | Logical LOC (líneas lógicas de código) | Se cuenta el número de <b>sentencias lógicas Java</b> . 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 <b>más de 1</b> . |
| 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.  |
| Nonexecutable:  |  |  |
| Declarations  | Sí                                     | Se incluyen declaraciones de variables, atributos, constantes y firmas de métodos/constructores (por ejemplo: int n;, private double sumX; public void logic1a() { ... } cuenta la declaración del método como 1 sentencia lógica).  |
| Compiler Directives   | No                                     | Java no usa directivas de compilador tipo #define, #include como C/C++. 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.  |
| Blank lines   | No                                     | No se cuentan líneas en blanco. Solo se usan para mejorar la legibilidad.  |
| 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\nsuma = sumaX\n+ sumaY\n+ sumaXY;\nEsto cuenta como <b>1</b> sentencia lógica.  |
| Varias sentencias en la misma línea cuentan por separado.                         |  | java\nint n = 0; n++; System.out.println(n);\nAquí hay <b>3</b> sentencias lógicas: declaración/asignación, incremento, llamada a println.   |

|   |  |  |
|---|--|--|
| Las estructuras de control se cuentan como una sentencia lógica por cada cabecera de control. |  | java\nif (x > 0) {\n    suma += x;\n}\nEl if (...) cuenta como <b>1</b> sentencia lógica. La asignación suma += x; es otra sentencia lógica. Total: <b>2</b> .   |
| Las firmas de métodos y constructores se cuentan como una sentencia lógica cada una.          |  | java\npublic void logic1a() {\n    ...\n}\nLa firma public void logic1a() se cuenta como <b>1</b> sentencia lógica. El contenido del método se cuenta según las sentencias que tenga.  |
| Las declaraciones múltiples cuentan como una sentencia por cada variable.                     |  | java\nint a, b, c;\nEsto se cuenta como <b>3</b> sentencias lógicas de declaración (una por cada variable), si se decide distinguirlas a nivel PSP; alternativamente puedes definir que cuenta como 1 (debes elegir una regla y mantenerla siempre). |
|   |  |  |
|   |  |  |
|   |  |  |
|   |  |  |
|   |  |  |
|   |  |  |
| Note  |  | Este estándar se aplica a todas las clases del proyecto: App, Input, Output, Data, Logic, EstimacionCorLineal y cualquier clase nueva que se añada.  |
| Note  |  |  |
| Note  |  |  |
| Note  |  |  |