El Principio de Sustitución de Liskov

Esta es la segunda de mis columnas del *cuaderno de ingeniería* para *The C++ Report.* Los artículos que aparecerán en esta columna van a enfocarse en el uso de C++ y OOD, y abordaran cuestiones de ingeniería de software.

Introducción

Los mecanismos primarios detrás del principio Abierto-Cerrado son la abstracción y el polimorfismo. En lenguajes estáticos tipados como C++ , una de sus mecanismos clave que soporta abstracción y polimorfismo es la herencia. Es mediante el uso de la herencia que podemos crear clases derivadas que conforman las interfaces abstractas polimórficas definidas por funciones virtuales puras en sus clases base abstracta.

El Principio de sustitución de Liskov

Funciones que usan punteros o referencias a clases bases deben poder usar objetos de clases derivadas sin saberlo.