

Optimización y documentación (Entornos de desarrollo)

Refactorización. Concepto.
Limitaciones. Patrones de
refactorización más usuales.
Refactorización y pruebas.
Herramientas de ayuda a la
refactorización

La refactorización consiste en aplicar transformaciones en el código para mejorar la estructura sin que cambie el comportamiento ni la funcionalidad del programa. El objetivo que se persigue es mejorar la estructura interna y limpiar el código para minimizar la aparición de errores.

Hay una pequeña diferencia respecto a la optimización, cuando se optimiza, se persigue una mejora en el rendimiento, por ejemplo mejorar la velocidad de ejecución de la aplicación.

Patrones de refactorización:

- Renombrado de clase, método, paquete, etc.
- Sustituir bloques de código por un método.
- Mover clases de un paquete a otro.
- Realizar borrado seguro.

Analizadores de código: los analizadores reciben un código fuente y lo procesan para mostrar posibles mejoras.

Ejercicio

Prueba de concepto simple de refactorización: pedir a 10 usuarios que complimenten unos datos siempre y cuando la edad lo permita

[crayon-6a9d4f69d21cc472131128/]

Control de versiones. Estructura de las herramientas de control de versiones. Repositorio. Herramientas de control de versiones

Para gestionar las distintas versiones que se van generando durante el desarrollo de una aplicación se utilizan las herramientas de Control de Versiones.

Los sistemas de Control de Versiones están formados por:

- Repositorios
- Módulos
- Revisión
- Etiquetas
- Rama

Algunas de las órdenes que se pueden ejecutar sobre un sistema de Control de Versiones son:

- Checkout
- Update
- Commit
- Abort

Documentación. Uso de comentarios.

Alternativas

Documentar un código sirve para explicar el funcionamiento del mismo, punto por punto, de forma que cualquier persona que acceda a la información podrá entender la finalidad del código.