

Programa curso JAVA BEGGINER

Clase 1 - Paradigmas de Programación

Programación estructurada o procedural. Concepto de algoritmo, proceso, subproceso, función, rutina, subrutina, subprocedimiento, procedimiento.

Programación orientada a objetos. Concepto de abstracción, encapsulamiento, modularidad, jerarquía

Ventajas y desventajas de los paradigmas anteriores

Diseño de algoritmos de software

JAVA como lenguaje orientado a objetos

Análisis orientado a objetos (con UML)

Diseño orientado a objetos (con UML)

Clase 2 – Entendiendo JAVA

Diferencia entre lenguajes compilados e interpretados

Clasificación de java en base al punto anterior

Arquitectura del framework de JAVA

Programación de la primer aplicación

Tipos de datos de java

Tipos primitivo y envoltorios

Conversión de datos

Casting de objetos

Clase 3 - La importancia del análisis y diseño.

Principales factores que encaminan al fracaso un proyecto de software.

La importancia del análisis previo del problema.

El diseño de la aplicación como punto de referencia

Buenas prácticas para el diseño de sistemas

Ejemplos prácticos de análisis y diseño de aplicaciones.

Clase 4 – Convenciones del lenguaje

Convenciones para el nombrado de clases, objetos, propiedad, métodos, etc.

Documentación del código con Java doc.

Generación de manuales de usuario a partir de Java doc.

Importancia de las convenciones a la hora de la reingeniería inversa.

Estandarización del proceso de desarrollo de software. Normas de calidad (CMMI)

Convenciones implícitas.

Clase 5 – *Ámbito de Variables y Objetos*

Definición de ámbito. visibilidad, interface, implementación.

Ejemplos de variables en distinto ámbito.

Como identificar el ámbito de una variable.

Evitar el abuso de variables globales.

Estructura de paquetes. Que son y cuando se usan.

Organización del framework de Java, paquetes más utilizados.

Clase 6 – *Relaciones entre clases*

Tipos de relaciones: agregación, composición y herencia

Identificación de relaciones entre clases

Codificació de los distintos ejemplos

Diagramación UML de las relaciones con Enterprise Architect

Importancia del diseño orientado a objetos.

Clase 7 – *Excepciones y Callbacks*

Que son la excepciones, como se utilizan.

Excepciones personalizadas.

Ejemplo de código + UML que describe la documentación de excepciones

Callbacks, relación y diferencias con las excepciones

Clase 8 – *Acceso a base de datos, programación en capas*

Objetos utilizados para el acceso a base de datos mediante JAVA

Conceptos de programación en capas con UML.

Ejemplos de código.