

Capítulo I. Introducción

1. Los inconvenientes de la programación clásica
2. La orientación a objetos
 - 2.1. El nacimiento de una manera nueva de construir aplicaciones
 - 2.2. Reutilización del código
3. Lenguajes de programación orientada a objetos
 - 3.1. Evolución histórica de los lenguajes de programación
 - 3.2. Evolución de los lenguajes de programación orientada a objetos
 - 3.3. Características básicas de los lenguajes de programación orientada a objetos

Capítulo II. Clases y objetos

1. Criterios de calidad
 - 1.1. Factores de calidad internos
 - 1.2. Factores de calidad externos
 - 1.2.1. Reutilización del código
 - 1.2.2. Escalabilidad de la aplicación
 - 1.2.3. Portabilidad del código
 - 1.2.4. Usabilidad
 - 1.2.5. Funcionalidad
 - 1.3. Otros factores de calidad
 - 1.3.1. Precisión
 - 1.3.2. Robustez
 - 1.3.3. Compatibilidad
 - 1.3.4. Rendimiento
 - 1.3.5. Oportunidad
 - 1.4. El papel de la orientación a objetos
2. El principio de modularidad
 - 2.1. Requerimientos de modularidad
 - 2.1.1. Descomposición
 - 2.1.2. Composición
 - 2.1.3. Comprensión
 - 2.1.4. Continuidad
 - 2.1.5. Protección
 - 2.2. Reglas de modularidad
 - 2.2.1. Mapeo directo
 - 2.2.2. Limitación del número de interfaces
 - 2.2.3. Limitación del tamaño de las interfaces
 - 2.2.4. Facilidad de reconocimiento de las interfaces
 - 2.2.5. Ocultación de la información
3. Clases y objetos
 - 3.1. Los papeles de la clase y del objeto
 - 3.2. La clase como estructura estática
 - 3.2.1. Atributos
 - 3.2.2. Métodos u operaciones
 - 3.2.3. Visibilidad de los atributos y métodos
 - 3.2.4. La clase como estructura modular
 - 3.2.5. Representación de una clase con UML
 - 3.3. El objeto como estructura dinámica
 - 3.4. Extensión de los atributos y de los métodos de una clase
 - 3.4.1. Los métodos constructor y destructor de una clase
 - 3.5. Sobrecarga
 - 3.6. Otros aspectos sobre los lenguajes de programación orientada a objetos

Capítulo III. Abstracción y clasificación

1. Complejidad y abstracción
 - 1.1. La complejidad inherente al software

- 1.2. La abstracción en el desarrollo de software
2. Los principios de la clasificación
 - 2.1. Modelar pensando en los objetos
 - 2.2. Modelar pensando en las clases
 - 2.3. Identificación de las operaciones
3. Relaciones entre clases
 - 3.1. Propiedades de las asociaciones
 - 3.1.1. Cardinalidad
 - 3.1.2. Navegabilidad
 - 3.1.3. Roles
 - 3.2. Tipos de asociaciones
 - 3.2.1. Asociaciones reflexivas
 - 3.2.2. Asociaciones de agregación
 - 3.2.3. Asociaciones de composición
 - 3.3. El concepto de clase asociativa
 - 3.4. Relación de generalización/especialización
4. Técnica simple para identificar clases
 - 4.1. Descripción de la técnica
 - 4.2. Ejemplo práctico
5. Tipos de clases
 - 5.1. Clases abstractas
 - 5.2. Clases parametrizadas
 - 5.3. Clases finales

Capítulo IV. Estructuras de objetos

1. Instanciación
 - 1.1. La gestión de la memoria
2. Recorridos
 - 2.1. Matrices y listas
 - 2.2. Objetos y estructuras de datos
 - 2.3. Recursividad
3. Delegación

Capítulo V. Herencia y polimorfismo

1. Definición de herencia
 - 1.1. Herencia por especialización
 - 1.2. Herencia por generalización
 - 1.3. Herencia múltiple
 - 1.3.1. Duplicidad de atributos y métodos
2. Consideraciones sobre la herencia
 - 2.1. Añadir métodos nuevos
 - 2.2. Implementar métodos abstractos
 - 2.3. Sobreescritura de métodos
 - 2.4. Concepto de 'casting'
3. Errores de la herencia
 - 3.1. Creación de superclases poco generales
 - 3.2. Uso de subclases en lugar de una superclase
4. Polimorfismo
5. Implementación de la herencia y el polimorfismo con Java
 - 5.1. Implementación de la herencia
 - 5.1.1. Interfaces y herencia múltiple
 - 5.2. Implementación del polimorfismo

Capítulo VI. Un ejemplo práctico

1. Enunciado
2. Resolución
 - 2.1. Identificación de las clases
 - 2.2. Creación de diagramas parciales de clases

- 2.3. Creación del diagrama de clases
- 2.4. Asignación de atributos y métodos
 - 2.4.1. Clase Persona
 - 2.4.2. Clase Estudiante
 - 2.4.3. Clase Pregunta
 - 2.4.4. Clase PreguntaTextual
 - 2.4.5. Clase PreguntaMultiopción
 - 2.4.6. Clase Opción
 - 2.4.7. Clase Test
 - 2.4.8. Clase TestContestado
 - 2.4.9. Clase PreguntaContestada
 - 2.4.10. Clase Aplicación
- 2.5. Construcción del diagrama completo
- 2.6. Inclusión de la navegabilidad en el diagrama de clases
- 2.7. Codificación

Capítulo VII. El lenguaje de programación Java

BLOQUE I: Java como lenguaje de programación estructurada

- 1. Introducción
 - 1.1. Origen del lenguaje
 - 1.2. Características del lenguaje
 - 1.3. Palabras reservadas
 - 1.4. Convenciones en el nombre de variables y funciones
 - 1.5. El primer programa en Java
- 2. Tipos de datos
 - 2.1. Números
 - 2.1.1. Números enteros
 - 2.1.2. Números en coma flotante
 - 2.2. Caracteres
 - 2.3. Tipos enumerados
 - 2.4. Constantes
 - 2.5. El tipo booleano
- 3. Operadores
 - 3.1. Operador de asignación
 - 3.2. Operadores aritméticos
 - 3.3. Operadores relacionales
 - 3.4. Operadores lógicos
 - 3.5. Operadores a nivel de bit
 - 3.6. Operadores equivalentes
 - 3.7. Operador condicional
 - 3.8. Precedencia de operadores
- 4. Matrices
- 5. Bloques de instrucciones
 - 5.1. Bloques condicionales
 - 5.1.1. Bloque if-else
 - 5.1.2. Bloque else if
 - 5.1.3. Bloque switch
 - 5.2. Bloques iterativos
 - 5.2.1. Bloque while
 - 5.2.2. Bloque do-while
 - 5.2.3. Bloque for
 - 5.3. Sentencia break
 - 5.4. Sentencia continue
- 6. Funciones
 - 6.1. La definición de funciones
 - 6.2. Los parámetros de entrada
 - 6.3. El valor de retorno
 - 6.4. Un ejemplo de función
 - 6.5. La invocación de funciones

- 7. Visibilidad de variables
- 7.1. Variables locales
- 7.2. Variables globales

BLOQUE II: Java como lenguaje de programación orientada a objetos

- 1. Definición de clases
 - 1.1. La clase
 - 1.2. Los atributos de una clase
 - 1.3. Métodos de una clase
 - 1.3.1. El método constructor
 - 1.3.2. El método destructor
 - 1.3.3. El resto de métodos
 - 1.4. Uso de la palabra reservada "this"
 - 1.5. Métodos estáticos
 - 1.6. Sobrecarga de métodos
- 2. Las relaciones entre clases
 - 2.1. Cardinalidad
 - 2.2. Número exacto
 - 2.3. Rango de valores
 - 2.4. Valores indefinidos
 - 2.5. Navegabilidad
 - 2.6. Roles
- 3. Librería de clases
 - 3.1. La clase String
 - 3.2. La clase ArrayList
- 4. Las excepciones
 - 4.1. Creación de una excepción
 - 4.2. Lanzamiento de excepciones
 - 4.3. Tratamiento de excepciones

Glosario

Bibliografía