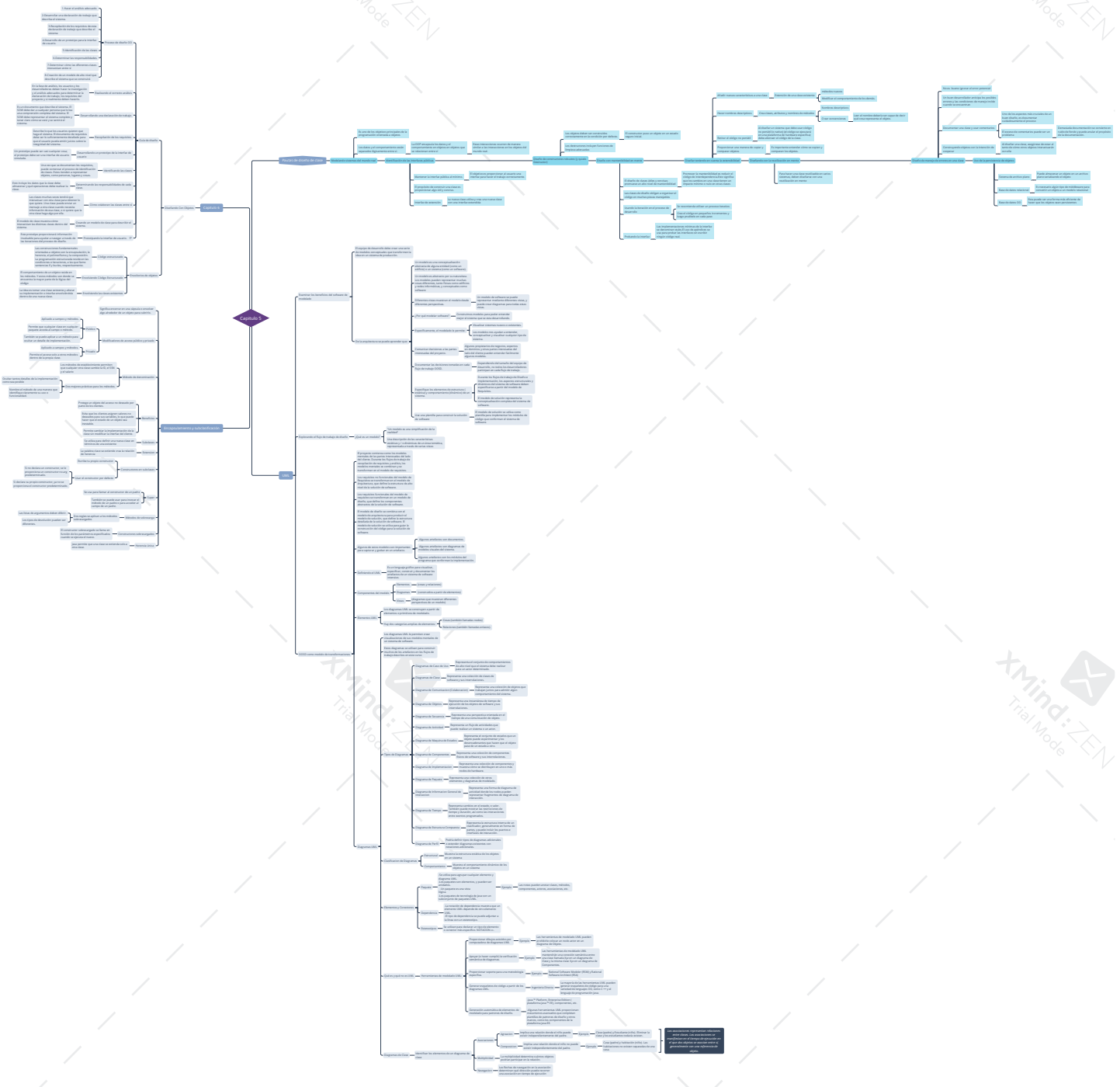
Capítulo 5
Capítulo 6
Diseñando Con Objetos
Fases de diseño
Proceso de diseño UML
1. Hacer el análisis adecuado
2.Desarrollar una declaración de trabajo que describa el sistema.
3.Recopilación de los requisitos de alto nivel de la declaración de trabajo que describa el sistema.
4.Diseñando de un prototipo para la interfaz de usuario.
5.Identificación de las clases
6.Determinar las responsabilidades
7.Determinar cómo las diferentes clases interactúan entre sí.
8.Creación de un modelo de alto nivel que describa el sistema que se construirá.
Diagramas UML
Diagramas de secuencia
En la fase de análisis, los usuarios les proporcionan a los analistas de sistemas una declaración del trabajo que describe el trabajo, los requisitos del proyecto y la interfaz deseo hacerlo.
Desarrollando el escenario analítico
Es un documento que describe el sistema. El documento de requisitos puede ser una declaración del trabajo, el SOW describe representar el sistema complejo y también es conocido como (por sus siglas en inglés) el sistema.
Desarrollando una declaración del trabajo
Describe lo que los usuarios quieren que haga el sistema. El documento de requisitos describe lo que el sistema debe hacer, pero lo que el usuario puede emitir juicios sobre la integridad del sistema.
Especificación de requisitos
Un prototipo puede ser vaca cualquier cosa. El prototipo debe ser una interfaz de usuario solamente.
Desarrollando un prototipo de la interfaz de usuario
Una vez que se documentan los requisitos, puede iniciarse el proceso de identificación de clases. Éstas tienden a representar objetos, conceptuales, lugares y cosas.
Identificando las clases
Este trabajo los dates que la clase debe contener y por supuestamente debe realizar la clase.
Determinando las responsabilidades de cada clase
Los clases resultan serse tendrá que interactuar con otra clase para obtener la información que necesita. Por ejemplo, el mensaje a una clase cuando necesita información de una clase, o a quien que le pide que realice una tarea.
Cómo colaborarán las clases entre sí
El modelo de clase muestra cómo interactúan las distintas clases dentro del sistema.
Después de modelo de clase para describir el sistema.
Este principio proporcionará información. Involucrarlo para ayudar a crearlo a través de los Entrevistas del proceso de diseño.
Prototipo(interfaz de usuario) 2
Encapsulamiento
Las construcciones fundamentales orientadas a objetos son el encapsulamiento, la herencia, el polimorfismo y la composición. La programación estructurada tendía en las construcciones de secuencia, selección y iteración. Es decir, requerían bucles.
Código estructurado
El comportamiento de un objeto recide en los métodos. Y estos métodos son dentro de la mayor parte de la lógica del objeto.
Encapsulando Código Estructurado
La lógica contiene varias clase anónimas y clases no implementadas o interfaz encerrándose dentro de una nueva clase.
Encapsulando las clases anónimas
Significa encerrar en una cápsula o envoltorio algo alrededor de un objeto para cubrirlo.
Polimorfismo
Aplicado a campos y métodos.
Permite que subtipar clase en cualquier propuesto acceda al campo o método.
También se puede aplicar a un método para ocultar un detalle de implementación.
Modificadores de acceso público y privado
Aplicado a campos y métodos.
Permitirá el acceso a una a métodos dentro de la propia clase.
Privado
Los métodos de establecimiento permiten que cualquier otra clase cambie la SI, el CSS y el valor.
Métodos de determinación
Ocultar varios detalles de la implementación como es posible.
Encapsulamiento
Siempre el método de una manera que cambiar la información no con o funcionalidad.
Encapsulamiento
Protege un objeto del acceso no deseado por parte de los clientes.
Protegido
Evita que los clientes asignen valores no deseados para sus variables. Si se puede cambiar el valor de un campo, se puede manipular.
Permite cambiar la implementación de la clase sin modificar la interfaz del cliente.
Subclases
Se utilizaran define una nueva clase en términos de una existente.
Herencia
La palabra clave se entiende crear la solución de herencia.
Herencia
Escribe la propia enumeración.
Construcciones en subclases
Es no declara un constructor, no le proporciona un constructor sin los predeterminado.
Uso al constructor por defecto
El declara ya un por consiguiente, ya no se proporciona el constructor predeterminado.
Uso al constructor
Se usa para llamar al constructor de un padre.
También se puede usar para invocar el constructor de un padre o para acceder al campo de un padre.
Super
Los tipos de argumentos deben diferir.
Los tipos de devolución pueden ser diferentes.
Una región en aplicar a los métodos sobrecargados.
Métodos de sobrecarga
El constructor sobrecargado se llama en otro constructor sobrecargado (con la palabra clave this) cuando se ejecuta el nuevo.
Constructores sobrecargados
Java permite que una clase se extienda solo a una clase.
Herencia única
Fases de diseño del ciclo de vida
Modelando(Clases del mundo real)
Es una de los objetivos principales de la programación orientada a objetos.
Los datos y el comportamiento están separados (figuramos arriba) y
Identificación de la interfaz pública
La UGP aseguraba los datos y el comportamiento en objetos que se relacionan arriba e
Hasta mencionamos ocurren de manera similar a las mencionadas arriba objetos que cambia real.
Diseño de componentes (modelo 3 capas interactuan)
Los objetos deben ser reutilizables correctamente en la solución por defecto.
El consumidor jeere un objeto en un estado seguro nivel
Los desarrolladores incluyen funciones de objetos adecuadas.
Diseño con reutilización
Diseñar con herencia
Diseñar con interfaces
Diseñar con composición
Diseñar con reutilización en mente
Diseñar con herencia
Diseñar con interfaces
Diseñar con composición
Diseñar con reutilización en mente
Diseñar con herencia
Diseñar con interfaces
Diseñar con composición
Diseñar con reutilización en mente
Diseñar con herencia
Diseñar con interfaces
Diseñar con composición
Diseñar con reutilización en mente
Diseñar con herencia
Diseñar con interfaces
Diseñar con composición
Diseñar con reutilización en mente
Diseño de manera los errores en una clase
Uso de la jerarquía de objetos
Diseñando
Diseñando
Diseñando