



REPORTE SESIÓN 3

Fecha: 5 de octubre de 2006

Hora: 5:00 p.m.

Número de Reunión: 3

Tema(s): INTRODUCCION A LA PROGRAMACION ORIENTADA A OBJETOS

Expositor(es): SONIA VIZCARRA LARIOS, CARLOS H. SANCHEZ

Orden del día

Actividad 1: ANUNCIOS, PEDIR AUXILIO PARA EL EVENTO DEL 1 DE NOVIEMBRE, COOPERACION PARA GASTOS DE LA CELULA.

Actividad 2: INTRODUCCION A LA PROGRAMACION ORIENTADA A OBJETOS I

Reporte: SONIA

El objetivo de esta exposición es Describir el Paradigma de Orientación a Objetos incluyendo los conceptos relacionados al análisis, diseño y programación

PARADIGMAS DE PROGRAMACION

Estructurados (C, Pascal, Basic, etc.)

Funcionales (CAML)

Declarativos (Prolog)

Orientados a Objetos (C#, VB.NET, Smalltalk, Java)

Orientados a Aspectos

Híbridos (Lisp, Visual Basic)

Incomprensibles

Cada enfoque tiene sus ventajas y desventajas

Cada uno es más apropiado para ciertas cosas

QUE ES UN OBJETO?

Todo es un objeto... Informalmente, un objeto representa una entidad del mundo real

Definición Formal (Rumbaugh):

“Un objeto es un concepto, abstracción o cosa con un significado y límites claros en el problema en cuestión”

Un objeto posee (Booch):

✓ Estado

1. El estado de un objeto es una de las posibles condiciones en que el objeto puede existir
2. El estado normalmente cambia en el transcurso del tiempo
3. El estado de un objeto es implementado por un conjunto de propiedades (atributos), además de las conexiones que puede tener con otros objetos

✓ Comportamiento

1. El comportamiento de un objeto determina cómo éste actúa y reacciona frente a las peticiones de otros objetos
2. Es modelado por un conjunto de mensajes a los que el objeto puede responder (operaciones que puede realizar)
3. Se implementa mediante métodos

✓ Identidad

1. Cada objeto tiene una identidad única, incluso si su estado es idéntico al de otro objeto

OBJETOS Y CLASES

Una clase es una definición abstracta de un objeto

- Define la estructura y el comportamiento compartidos por los objetos
- Sirve como modelo para la creación de objetos

MODIFICADORES DE ACCESO

Permiten definir el nivel de acceso (visibilidad) de los miembros (atributos o métodos) de una clase

- Público: Cualquier clase puede "ver" los miembros públicos de otra clase
- Privado: Sólo la clase puede ver sus propios miembros privados

Existen otros dos modificadores para propósitos específicos (Paquete, Protegido)

Actividad 2: INTRODUCCION A LA PROGRAMACION ORIENTADA A OBJETOS II

Reporte: CARLOS H

QUE ES UML?

"UML es un lenguaje visual para especificar, construir y documentar sistemas" (OMG - Object Management Group)

Unified (UNIFICADO):

- El aporte de muchos métodos y notaciones
- Independiente de implementaciones, plataformas y lenguajes

Modeling (MODELADO):

- Los modelos son utilizados en todas las ingenierías

Language (LENGUAJE):

- Si hay gente, requieren comunicarse. Si se tienen que comunicar, se tienen que entender. Para entenderse necesitan un lenguaje común

PILARES DE LA ORIENTACION A OBJETOS

✓ ABSTRACCION

Ignorancia Selectiva

- La abstracción nos ayuda a trabajar con cosas complejas
- Se enfoca en lo importante
- Ignora lo que no es importante (simplifica)

Una clase es una abstracción en la que:

- Se enfatizan las características relevantes
- Se suprimen otras características

Una clase debe capturar una y solo una abstracción clave

✓ RELACIONES

Todo sistema abarca muchas clases y objetos

Los objetos contribuyen en el comportamiento de un sistema colaborando entre si

- La colaboración se logra a través de las relaciones

Existen dos tipos principales de relaciones

- Asociación (tener, puede tener)
- Agregación (es parte de...)

✓ HERENCIA

Es una relación entre clases en la cual una clase comparte la estructura y comportamiento definido en otra clase (Grady Booch)

Cada clase que hereda de otra posee:

- Los atributos de la clase base además de los propios
- Soporta todos o algunos de los métodos de la clase base

Una subclase hereda de una clase base

Herencia "Es-Un": herencia real, donde la subclase es un tipo específico de la superclase

✓ ENCAPSULAMIENTO

Principio que establece que los atributos propios de un objeto no deben ser visibles desde otros objetos

- Deben ser declarados como privados
 - Permite abstraer al resto del mundo de la complejidad de la implementación interna
 - Permite exponer el estado del objeto sólo a través del comportamiento que le hayamos definido mediante miembros públicos
- ¿Por qué es útil?
 - Punto de Control/Validación
 - Mejor respuesta ante los Cambios

CONCEPTO DEL DISEÑO ORIENTADO A OBJETOS (INTERFACES)

Recurso de diseño soportado por los lenguajes orientados a objetos que permite definir comportamiento

Permite que clases que no están estrechamente relacionadas entre sí deban tener el mismo comportamiento

La implementación de una interfaz es un contrato que obliga a la clase a implementar todos los métodos definidos en la interfaz

Se crean las interfaces que definen comportamiento

(POLIMORFISMO)

Es la propiedad que tienen los objetos de permitir invocar genéricamente un comportamiento (método) cuya implementación será delegada al objeto correspondiente recién en tiempo de ejecución.

El polimorfismo tiende a existir en las relaciones de herencia, pero no siempre es así.

EJEMPLOS

- La definición del método reside en la clase base
- La implementación del método reside en la clase derivada
- La invocación es resuelta al momento de ejecución

NO TAREAS