

Orientación a Objetos

ÍNDICE

| | |
|---|-----------|
| 1. OBJETOS | 3 |
| 1.1. TIPOS DE OBJETOS | 3 |
| 1.2. MÉTODOS | 4 |
| 1.3. ENCAPSULADO | 4 |
| 1.4. MENSAJES | 5 |
| 1.5. CLASE | 6 |
| 1.6. HERENCIA | 6 |
| 2. ANÁLISIS DE LA ESTRUCTURA DE OBJETOS | 7 |
| 2.1. OBJETOS Y TIPOS DE OBJETOS | 7 |
| 2.2. ASOCIACIONES DE OBJETOS | 7 |
| 2.3. JERARQUÍAS DE GENERALIZACIÓN | 8 |
| 2.4. JERARQUÍAS COMPUESTAS | 8 |
| 2.5. DIAGRAMAS DE RELACIÓN ENTRE LOS OBJETOS | 9 |
| 2.6. ESQUEMAS DE OBJETOS | 9 |
| 2.7. EJERCICIOS | 9 |
| 3. ANÁLISIS DEL COMPORTAMIENTO DE OBJETOS | 11 |
| 3.1. ESTADOS DE UN OBJETO | 11 |
| 3.2. EVENTOS. | 11 |
| 3.3. TIPOS DE EVENTOS | 12 |
| 3.4. EL CICLO VITAL DE UN OBJETO | 13 |
| 3.5. INTERACCIONES ENTRE TIPOS DE OBJETOS | 14 |
| 3.6. OPERACIONES | 15 |
| 3.7. FUENTES EXTERNAS DE EVENTOS | 15 |
| 3.8. REGLAS DE ACTIVACIÓN | 15 |
| 3.9. CONDICIONES DE CONTROL | 16 |
| 3.10. SUBTIPOS Y SUPERTIPOS DE EVENTOS | 16 |
| 4. DISEÑO DE LA ESTRUCTURA Y COMPORTAMIENTO DE UN OBJETO | 18 |
| 4.1. CLASE | 18 |
| 4.2. DIFERENCIA ENTRE OPERACIÓN Y MÉTODO | 19 |
| 4.3. HERENCIA DE CLASE | 19 |
| 4.4. HERENCIA MÚLTIPLE | 19 |
| 4.5. SELECCIÓN DEL MÉTODO | 20 |
| 4.6. POLIMORFISMO | 20 |
| 4.7. NOTACIÓN. | 21 |
| 5. DOCUMENTACIÓN DEL ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS | 22 |
| 5.1. DOCUMENTACIÓN DEL ANÁLISIS | 22 |
| 5.1.1. Diagramas a Incluir | 22 |
| 5.1.2. Documentación Asociada a los Diagramas | 22 |
| 5.2. DOCUMENTACIÓN DEL DISEÑO | 22 |
| 5.2.1. Diagramas a incluir | 22 |
| 5.2.2. Documentación Asociada a los Diagramas | 23 |

1. OBJETOS

Las personas tienen una idea clara de lo que es un objeto: conceptos adquiridos que nos permiten sentir y razonar acerca de las cosas del mundo. Un objeto podría ser real o abstracto, por ejemplo una organización, una factura, una figura en un dibujador, una pantalla de usuario, un avión, un vuelo de avión, etc.

En el análisis y diseño orientados a objetos (OO), interesa el comportamiento del objeto. Si se construye software, los módulos de software OO se basan en los tipos de objetos. El software que implanta el objeto contiene estructuras de datos y operaciones que expresan dicho comportamiento. Las operaciones se codifican como métodos. La representación en software OO del objeto es entonces una colección de tipos de datos y objetos.

Entonces, dentro del software orientado a objeto, un objeto es cualquier cosa, real o abstracta, acerca de la cual almacenamos datos y los métodos que controlan dichos datos.

Un objeto puede estar compuesto por otros objetos. Estos últimos a su vez también pueden estar compuestos por otros objetos. Esta intrincada estructura es la que permite construir objetos muy complejos.

1.1. TIPOS DE OBJETOS

Los conceptos que poseemos se aplican a tipos determinados de objetos. Por ejemplo, empleado se aplica a los objetos que son personas empleadas por alguna organización. Algunas instancias de empleado podrían ser Juan Pérez, José Martínez, etc. En el análisis orientado a objetos, estos conceptos se llaman tipos de objetos; las instancias se llaman objetos.

Así, un tipo de objeto es una categoría de objeto, mientras que un objeto es una instancia de un tipo de objeto.

En el mundo de las bases de datos existen los tipos de entidad, como cliente o empleado. Existen muchas instancias de cada tipo de entidad (como Juan Pérez o José Martínez para el tipo de entidad empleado). Del mismo modo, en OO se define tipos de objetos e instancias de tipo de objeto.

Sin embargo, el término objeto tiene diferencias fundamentales con el término entidad, ya que la entidad sólo se refiere a los datos, mientras que objeto se refiere a los datos y a los métodos mediante los cuales se controlan a los propios datos. En OO, la estructura de datos y los métodos de cada tipo de objeto se manejan juntos. No se puede tener acceso o control de la estructura de datos excepto mediante los métodos que forman parte del tipo de objeto.

1.2. MÉTODOS

Los métodos especifican la forma en que se controlan los datos de un objeto. Los métodos en un tipo de objeto sólo hacen referencia a la estructura de datos de ese tipo de objeto. No deben tener acceso directo a las estructuras de datos de otros objetos. Para utilizar la estructura de datos de otro objeto, deben enviar un mensaje a éste. El tipo de objeto empaqueta juntos los tipos de datos y su comportamiento.

Un objeto entonces es una cosa cuyas propiedades están representadas por tipos de datos y su comportamiento por métodos.

Un método asociado con el tipo de objeto factura podría ser aquel que calcule el total de la factura. Otro podría transmitir la factura a un cliente. Otro podría verificar de manera periódica si la factura ha sido pagada y, en caso contrario, añadir cierta tasa de interés.

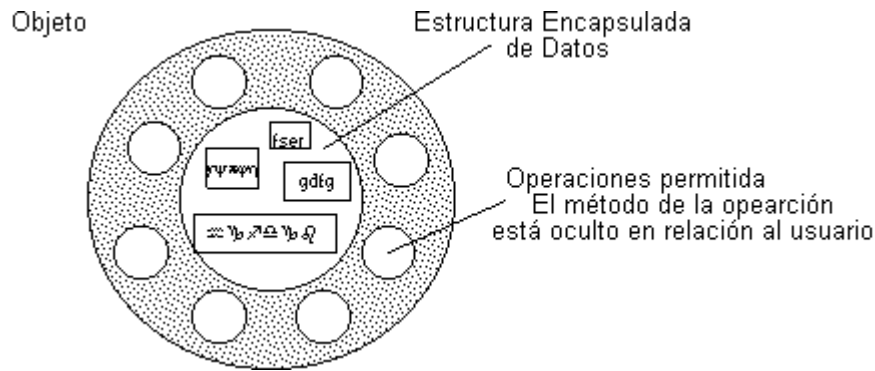
1.3. ENCAPSULADO

El empaque conjunto de datos y métodos se llama encapsulado. El objeto esconde sus datos de los demás objetos y permite el acceso a los datos mediante sus propios métodos. Esto recibe el nombre de ocultamiento de información. El encapsulamiento evita la corrupción de los datos de un objeto. Si todos los programas pudieran tener acceso a los datos de cualquier forma que quisieran los usuarios, los datos se podrían corromper o utilizar de mala manera. El encapsulado protege los datos del uso arbitrario y no pretendido.

El encapsulado oculta los detalles de su implantación interna a los usuarios de un objeto. Los usuarios se dan cuenta de las operaciones que puede solicitar del objeto, pero desconocen los detalles de cómo se lleva a cabo la operación. Todos los detalles específicos de los datos del objeto y la codificación de sus operaciones están fuera del alcance del usuario.

Así, encapsulado es el resultado (o acto) de ocultar los detalles de implantación de un objeto respecto de su usuario.

El encapsulado, al separar el comportamiento del objeto de su implantación, permite la modificación de ésta sin que se tengan que modificar las aplicaciones que lo utilizan.

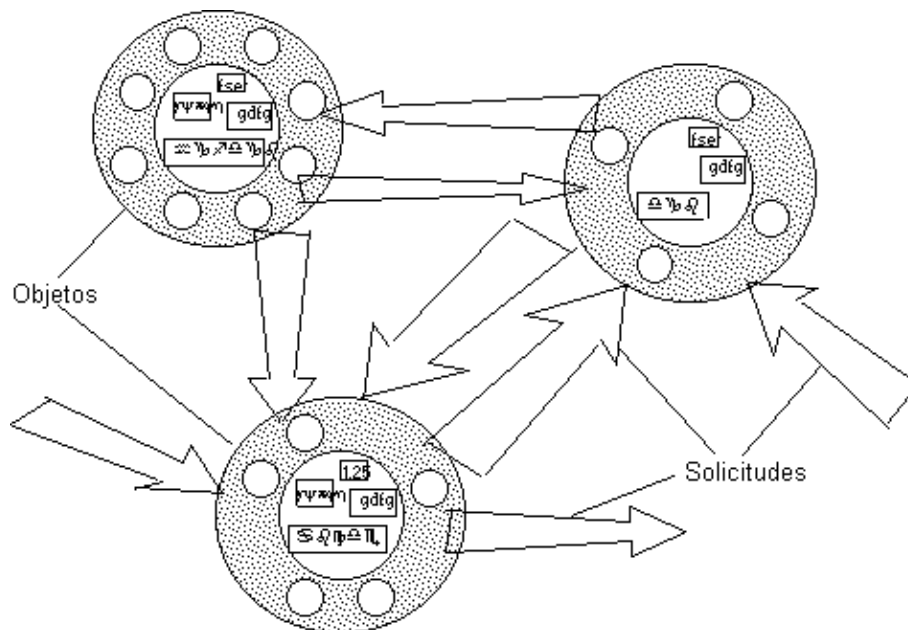


1.4. MENSAJES

Para que un objeto haga algo, le enviamos una solicitud. Esta hace que se produzca una operación. La operación ejecuta el método apropiado y, de manera opcional, produce una respuesta. El mensaje que constituye la solicitud contiene el nombre del objeto, el nombre de una operación y, a veces, un grupo de parámetros.

La programación orientada a objetos es una forma de diseño modular en la que con frecuencia el mundo se piensa en términos de objetos, operaciones, métodos y mensajes que se transfieren entre tales objetos. Un mensaje es una solicitud para que se lleve a cabo la operación indicada y se produzca el resultado.

Los objetos pueden ser muy complejos, puesto que pueden contener muchos subobjetos, éstos a su vez pueden contener otros, etc. La persona que utilice el objeto no tiene que conocer su complejidad interna, sino la forma de comunicarse con él y la forma en que responde.



1.5. CLASE

El término clase se refiere a la implantación en software de un tipo de objeto. El tipo de objeto es una noción de concepto. Especifica una familia de objetos, sin estipular la forma en que se implanten. Los tipos de objetos se especifican durante el análisis OO. Así, una clase es una implantación de un tipo de objeto. Especifica una estructura de datos y los métodos operativos permisibles que se aplican a cada uno de sus objetos.

1.6. HERENCIA

Un tipo de objeto de alto nivel puede especializarse en tipos de objeto de bajo nivel. Un tipo de objeto puede tener subtipos. Por ejemplo, el tipo de objeto persona puede tener subtipos estudiante y empleado. A su vez, el tipo de objeto estudiante puede tener como subtipo estudiante de pregrado y estudiante de postgrado, mientras que empleado puede tener como subtipo a académico y administrativo. Existe de este modo una jerarquía de tipos, subtipos, sub-subtipos, etc..

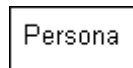
Una clase implanta el tipo de objeto. Una subclase hereda propiedades de su clase padre; una sub-subclase hereda propiedades de las subclases; etc. Una subclase puede heredar la estructura de datos y los métodos, o algunos de los métodos, de su superclase. También tiene sus métodos e incluso tipos de datos propios.

2. ANÁLISIS DE LA ESTRUCTURA DE OBJETOS

El análisis de la estructura de objetos (AEO) define las categorías de los objetos que percibimos y las formas en que los asociamos.

2.1. OBJETOS Y TIPOS DE OBJETOS

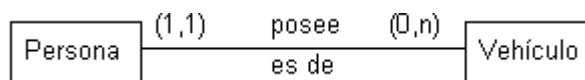
En el análisis se trata de identificar los tipos de objeto más que los objetos individuales en un sistema. Los tipos de objetos se definen en base a la comprensión del analista de nuestro mundo. Un objeto puede categorizarse de variadas formas.



Representación para Tipo de Objeto (Persona).

2.2. ASOCIACIONES DE OBJETOS

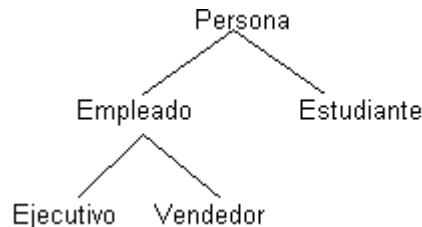
Es importante modelar la forma como los objetos se asocian entre sí. Además es necesario identificar el significado de la asociación y la cantidad de objetos con los que un objeto dado puede y debe asociarse (cardinalidad).



Representación para la Asociación entre dos Tipos de Objetos. Un objeto del tipo persona posee cero o muchos objetos del tipo vehículo. Un objeto del tipo vehículo es de un y sólo un objeto del tipo persona.

2.3. JERARQUÍAS DE GENERALIZACIÓN

Una de las vías de sentido común por las que el hombre organiza su volumen de conocimiento es el de las jerarquías, de lo más general a lo más específico.

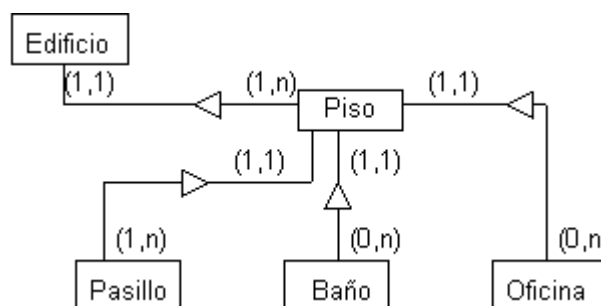


Representación de una Jerarquía de generalización, para el tipo de objeto Persona.

En las jerarquías se habla de subtipo o especialización de un supertipo o generalización. En el caso anterior, persona es el supertipo para Empleado y Estudiante, que son sus subtipos. Por otra parte, Empleado es el supertipo para los subtipos Ejecutivo y Vendedor. Los subtipos (niveles inferiores de la jerarquía) heredan las características de sus supertipos, además, cada instancia de un tipo de objeto lo es también de sus supertipos.

2.4. JERARQUÍAS COMPUESTAS

Un objeto se denomina complejo si está formado por otros. Las jerarquías Compuestas permiten realizar agregaciones de objetos.



Representación de una Jerarquía Compuesta.

Un objeto del tipo edificio se compone de a lo menos un objeto del tipo piso. A su vez un objeto del tipo piso se compone de a lo menos un objeto del tipo pasillo, podría tener varios (o ninguno) objetos del tipo baño y oficina.

2.5. DIAGRAMAS DE RELACIÓN ENTRE LOS OBJETOS

Los tipos de objetos están relacionados con otros tipos de objeto. Por ejemplo, un empleado trabaja en una sucursal, o un cliente realiza un pedido de varios productos.

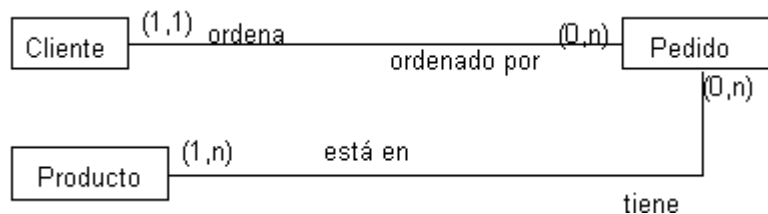


Diagrama de Relación entre objetos.

Un objeto del tipo cliente puede ordenar muchos objetos del tipo pedidos, y un objeto del tipo pedido es ordenado por un y sólo un objeto del tipo cliente. Un objeto del tipo producto está en muchos o ningún objeto del tipo pedido, mientras que un objeto del tipo pedido tiene al menos un objeto del tipo producto.

2.6. ESQUEMAS DE OBJETOS

La comprensión de un modelo suele ser más sencilla si los tipos de objetos y relaciones se presentan mediante un diagrama de relación entre objetos; los supertipos y subtipos se presentan en un diagrama de jerarquías de generalización y las estructuras compuestas en un diagrama compuesto. Sin embargo, para los usuarios más sofisticados puede ser útil presentarlo todo en un mismo diagrama, el que se denomina esquema de objetos.

2.7. EJERCICIOS

a:

Diseñe la estructura de un sistema que maneje la información de productos en una librería, que vende libros técnicos y novelas.

Las novelas se clasifican como de ciencia ficción, romance, misterio, juveniles y policiales. Los libros técnicos se clasifican como de ingeniería, ciencias naturales o ciencias sociales.

Cada libro tiene un título, uno o más autores, una editorial, un año de edición y un formato (tapas duras o edición económica). Los libros técnicos tienen además un código ISBN y capítulos, los que tratan una o más materias.

La librería obtiene los libros por medio de proveedores que representan a una o más editoriales. De cada libro se tiene un stock (que puede ser cero). Al venderse un libro, el stock se actualiza. Si un cliente requiere un libro cuyo stock es cero, se puede realizar un encargo por parte del cliente. Esto significa que se pide el libro a un proveedor de la editorial del libro.

b:

El jefe de una organización de desarrollo de software desea administrar el desarrollo de los proyectos.

Cada proyecto de desarrollo de software está a cargo de un jefe de proyecto, el cual lidera y coordina al equipo de proyecto, conformado por un grupo de análisis y diseño, un grupo de construcción y un representante de los usuarios que solicitó el software.

Cada proyecto contempla el desarrollo de una o más softwares (aplicaciones) que se componen a su vez de elementos de software (aplicaciones módulo). Cada elemento de software tiene un nombre, un responsable del análisis y diseño (miembro de ese grupo), un responsable de la construcción (miembro del grupo de construcción), una fecha de actualización (la última) y un plan de test (a desarrollar por el equipo de SQA).

Un plan de test es una declaración narrativa de las pruebas a aplicar al elemento de software para aprobarlo o no. El equipo SQA pertenece a la organización, pero debe ser independiente del equipo de desarrollo del software que contiene al elemento a evaluar.

Se desea poder manipular esta información y generar reportes de todos los proyectos en desarrollo.

Utilice las técnicas orientadas al objeto revisadas en clases.

3. ANÁLISIS DEL COMPORTAMIENTO DE OBJETOS

En el análisis del comportamiento de objetos (ACO) realizamos esquemas de eventos que muestran eventos, la secuencia en que ocurren y cómo los eventos cambian el estado de los objetos.

3.1. ESTADOS DE UN OBJETO

Un objeto puede existir en varios estados. Por ejemplo, un objeto reservación aérea puede ser una instancia de alguno de los siguientes tipos de objeto:

- Reservación solicitada,
- Reservación en lista de espera,
- Reservación confirmada,
- Reservación cancelada,
- Reservación satisfecha,
- Reservación archivada.

Tales tipos de objetos suelen percibirse como estados posibles del ciclo vital de un objeto. Sin embargo, un objeto puede tener una gran variedad de perspectivas de ciclos vitales. Por ejemplo, el mismo objeto reservación aérea también puede tener los siguientes estados relacionados con el pago:

- Reservación no liquidada,
- Reservación con un pago de depósito,
- Reservación totalmente pagada,
- Reservación reembolsada.

Así, el estado de un objeto es la colección de asociaciones que tiene un objeto.

3.2. EVENTOS

El mundo está lleno de eventos: una coneja tiene conejitos, llega el pesado del vecino en forma inesperada, un cliente solicita un préstamo, el servidor se cae, se termina la tarea, etc. En el análisis orientado a objetos el mundo se describe en términos de los objetos y sus estados, así como los eventos que modifican esos estados. Un evento produce un cambio en el estado de un objeto. Los eventos sirven como indicadores de los instantes en que ocurren los cambios de estado.

Para saber de los cambios y reaccionar adecuadamente ante ellos, debemos entender y modelar los eventos.

3.3. TIPOS DE EVENTOS

El analista no necesita conocer cada evento que ocurra en una organización: sólo los tipos de eventos. Por ejemplo, el tipo de evento *reservación en lista de espera confirmada* es la colección de eventos donde un objeto cambia de una *reservación en lista de espera* a una *reservación confirmada*.

Los tipos de eventos indican los cambios sencillos en el estado de un objeto; por ejemplo, cuando se deposita dinero en una cuenta bancaria o se actualiza el sueldo de un trabajador. Básicamente, los tipos de eventos describen las siguientes formas de cambios de estado:

- Un objeto se crea.
- Un objeto se termina.
- Un objeto se clasifica como una instancia de un tipo de objeto.
- Un objeto se desclasifica como una instancia de un tipo de objeto.
- Un objeto cambia de clasificación.
- Un atributo de un objeto se cambia.

Los objetos pueden asociar un objeto con otro. Por ejemplo, en la mayoría de las organizaciones, cuando un objeto se clasifica como empleado, debe estar asociado con un departamento. Un evento clasificará al objeto como empleado. Otro evento creará una asociación entre el objeto empleado y un objeto Departamento (las asociaciones son objetos como los demás).

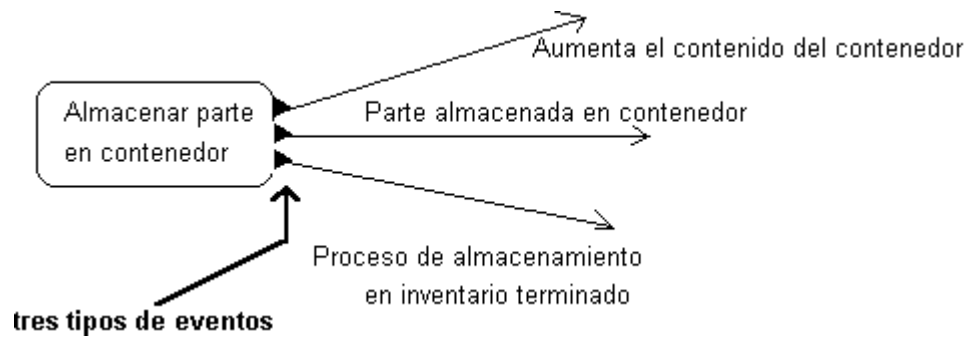
Algunos eventos requieren que antes ocurran otros. Por ejemplo, antes de cerrar un departamento, todos los empleados deben ser asignados a otra parte, las oficinas que ocupaban deben tener otro uso, etc..

Algunas veces, un evento puede provocar la reacción encadena de otros eventos. Por ejemplo, el cambio de circuito a las conexiones de un avión, puede exigir cambios a varios otros objetos.

Una operación hace que los eventos ocurran. Dibujamos la operación como un cuadro con esquinas redondeadas, puesto que los eventos indican los puntos en el tiempo en que se da el cambio de estado de un objeto. Los tipos de eventos se representan como triángulos negros llenos, generalmente unidos a la caja de operación.

- ▶ La gata tuvo gatitos
- ▶ Se solicita reservación aérea
- ▶ Trabajo terminado

Según el área que se modele, puede ocurrir más de un evento al terminar una operación, y cada uno de estos puede activar operaciones independientes.



3.4. EL CICLO VITAL DE UN OBJETO

La mayoría de los objetos tienen un ciclo vital en el que una sucesión de eventos pueden ocurrirle y cada uno de éstos modifica su estado.

En este análisis, se dibuja un diagrama que muestre el ciclo vital de un objeto, incluyendo los estados posibles de los objetos, además de los cambios de estado permisibles. Este se denomina diagrama de reja.

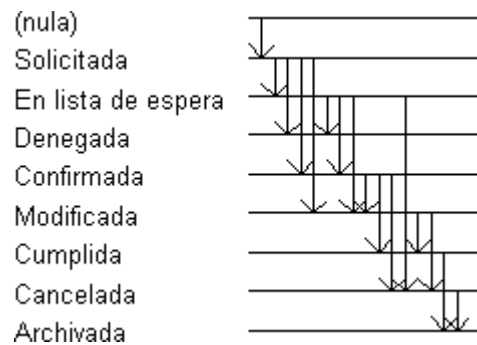
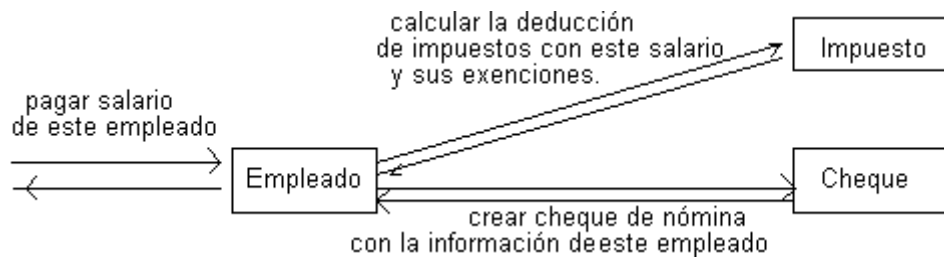


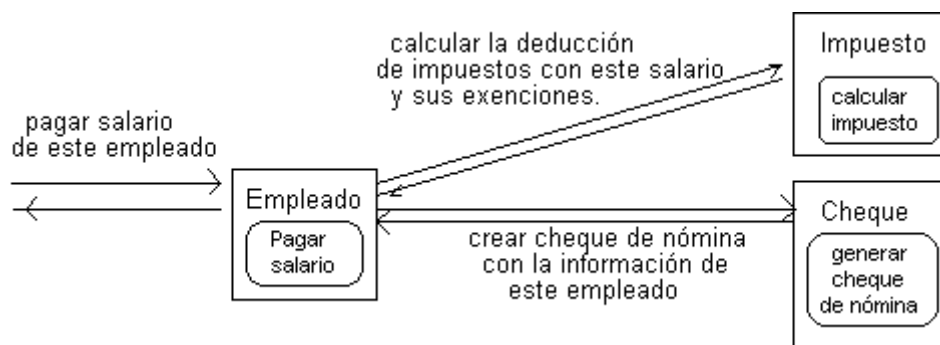
Diagrama de rejilla que muestra los estados posibles de un objeto reservación aérea. Las líneas horizontales representan estados y las verticales muestran las transiciones entre estados.

3.5. INTERACCIONES ENTRE TIPOS DE OBJETOS

La mayoría de los procesos requieren la interacción de varios objetos.



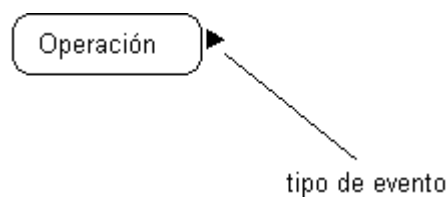
En esta otra figura, se desarrolla el diagrama anterior para mostrar las operaciones necesarias.



3.6. OPERACIONES

En el análisis OO, una operación se refiere a una unidad de procesamiento que puede ser solicitada. El procedimiento se implanta mediante un método. El método es la especificación de cómo llevar a cabo la operación. A nivel de programa, el método es el código que implanta la operación.

Las operaciones se invocan. Una operación invocada es una instancia de una operación. Una operación puede o no cambiar el estado de un objeto, si lo cambiara ocurriría un evento.



3.7. FUENTES EXTERNAS DE EVENTOS

Los eventos son cambios de estado que un sistema debe conocer y reaccionar ante ellos de algún modo. Muchas de las operaciones que producen estos eventos son externas al sistema.

3.8. REGLAS DE ACTIVACIÓN

Cuando ocurre un evento, lo normal es que el cambio de estado active el llamado a una o más operaciones. Por ejemplo, si se retiran bienes de un almacén y la cantidad baja de cierto nivel, ello puede activar una operación para volver a realizar un pedido.

Las reglas de activación definen la relación entre la causa y el efecto. Siempre que ocurra un evento de cierto tipo, la regla de activación invoca a una operación ya definida.

Un tipo de evento puede tener varias reglas de activación, cada una de las cuales invoca a su operación en paralelo. Las operaciones paralelas pueden producir diferentes cambios de estado en forma simultánea.

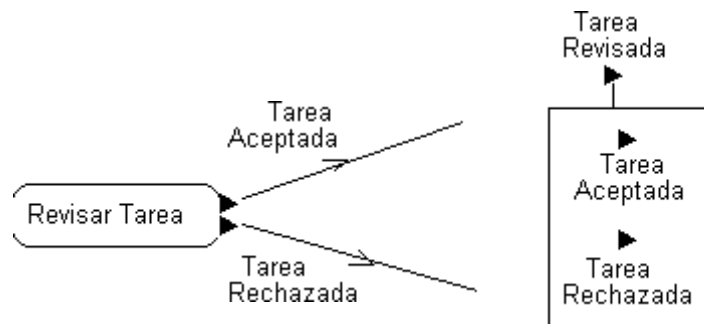
3.9. CONDICIONES DE CONTROL

Una operación puede ser invocada por una o varias reglas de activación. Sin embargo, antes de invocar de hecho a la operación se puede verificar una condición de control. Si el resultado de evaluación de la condición es verdadera se invoca su operación, en otro caso no.

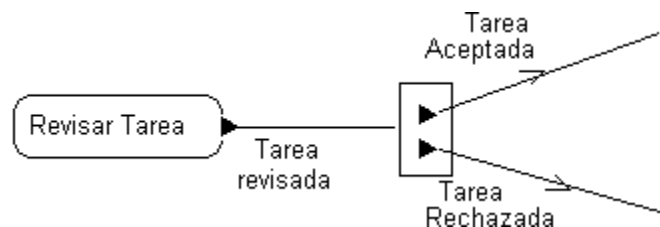


Las condiciones de control también pueden actuar como puntos de sincronización para el procesamiento en paralelo, pues garantizan que un conjunto de eventos esté completo antes de proceder con una operación.

3.10. SUBTIPOS Y SUPERTIPOS DE EVENTOS



Los eventos pueden dividirse en subtipos mediante diagramas independientes; o bien, es posible expresar la misma información en un diagrama ampliado.



La operación revisar tarea produce dos eventos: tarea aceptada o tarea rechazada. Sólo se puede dar uno de estos tipos de evento al revisar una tarea.

Aquí tarea revisada es un supertipo de tarea aceptada y tarea rechazada, que son los subtipos. Siempre se entiende que existe una relación de exclusividad entre los subtipos.

Las particiones de eventos no son operaciones independientes que coordinan las condiciones de bifurcación para las formas de activación ajenas, sino que indican los objetivos y distintos subobjetivos de las operaciones a las que están asociadas.

4. DISEÑO DE LA ESTRUCTURA Y COMPORTAMIENTO DE UN OBJETO

En el diseño de la estructura y comportamiento de objetos se identifican los componentes siguientes:

- Clases que se implantarán. Los tipos de objetos en el AEO serán la guía en esta decisión.
- Estructuras de Datos que utilizará cada clase. Se puede hacer un diagrama para representar la estructura de datos.
- Operaciones que ofrecerá cada clase y cuáles serán sus métodos. Se enumeran las operaciones y se especifican los métodos.
- Forma de Implantación de la herencia de clases y efecto sobre las especificaciones de los datos y operaciones.
- Identificación de variantes de clases ("igual que, excepto...").

4.1. CLASE

En el análisis de estructura de objetos, se identificaron tipos de objetos; en el diseño de estructura de objetos nos centramos en la implantación de esos tipos de objetos.

Clase es la implantación de un tipo de objeto. Especifica la estructura de datos y los métodos operativos permitidos que se aplican a cada uno de sus objetos.

La clase especifica la estructura de datos de cada uno de sus objetos y las operaciones que se utilizan para tener acceso a los objetos. La especificación de cómo se llevan a cabo las funciones de una clase se llama método. Los objetos se pueden utilizar exclusivamente con métodos específicos.

Una instancia de una clase, almacena sus datos dentro de él. Se tiene acceso a los datos y se les modifica sólo mediante operaciones permisibles. Esta restricción al acceso se debe al encapsulado. El encapsulado protege los datos del uso arbitrario o no permitido. El acceso o la actualización directa de los datos de un objeto por parte del usuario violaría el encapsulado.

Los usuarios observan el "comportamiento" del objeto en términos de las operaciones que se pueden aplicar a los objetos, así como los resultados de tales operaciones. Estas operaciones forman la interfaz del objeto con sus usuarios.

4.2. DIFERENCIA ENTRE OPERACIÓN Y MÉTODO

Las operaciones son procesos que se pueden solicitar como unidades. Los métodos son especificaciones del procedimiento de una operación dentro de una clase. Es decir, la operación es el tipo de servicio solicitado y el método es su código de programación.

Por ejemplo una operación asociada con la clase pedido podría ser aquella que calcule el total del pedido. El método especificaría la forma de calcular el total. Para esto, el método podría obtener el precio de cada artículo del pedido al enviar una solicitud a los objetos artículo asociados. A su vez, cada objeto artículo regresaría su precio al método pedido mediante un método de la clase artículo.

Los métodos de una clase controlan solamente a los objetos de esa clase. No pueden tener acceso directo a las estructuras de datos de un objeto en una clase distinta. Para utilizar las estructuras de datos en una clase diferente, deben enviar una solicitud a ese objeto.

4.3. HERENCIA DE CLASE

La generalización es una noción conceptual. La herencia de clase (que sólo se conoce como herencia) es una implantación de la generalización. La generalización establece que las propiedades de un tipo se aplican a sus subtipos.

La herencia de clase hace que la estructura de datos y operaciones sean disponibles para su reutilización por parte de sus subclases. La herencia de las operaciones de una superclase permite que las clases compartan código. La herencia de la estructura de datos permite la reutilización de la estructura.

4.4. HERENCIA MÚLTIPLE

En la herencia múltiple, una clase puede heredar estructuras de datos y operaciones de más de una superclase.

Por ejemplo supóngase que existe un tipo de objeto cuenta, que tiene como subtipos a los tipos de objetos cuenta de cliente y cuenta vencida. A su vez, cuenta de cliente tiene como a subtipo a cuenta de cliente vencida y cuenta vencida también tiene como subtipo a cuenta de cliente vencida.

4.5. SELECCIÓN DEL MÉTODO

Cuando se envía una solicitud a un objeto, el software selecciona los métodos por utilizar. El método no se almacena en el objeto, pues esto causaría una réplica múltiple y pérdida de espacio. En vez de esto, el método se asocia con la clase. El método puede no estar en la clase de la que el objeto es una instancia, sino en una superclase.

En ese caso, el mecanismo de selección buscará la operación en su superclase y en todas las superclases de la jerarquía hasta que lo encuentre, nivel por nivel. Si la encuentra, selecciona la operación. Si la operación no se encuentra en ningún nivel de la superclase, se considera inválida la fuente de la solicitud.

De esta forma, los usuarios sólo deben especificar lo que se debe hacer, dejando que sea el mecanismo de selección el que determine la forma de localizar la operación y la ejecute. El mecanismo de selección deja en manos de la aplicación OO el problema de localizar la operación y la ejecute.

4.6. POLIMORFISMO

Uno de los objetivos principales de las técnicas OO es utilizar otra vez el código. Sin embargo, algunas de las operaciones requieren adaptación para resolver necesidades particulares.

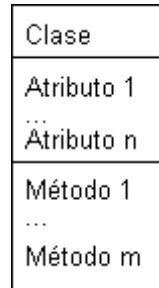
Esta necesidad, se da generalmente entre superclases y subclases, donde una subclase es una especialización de su superclase, y puede requerir alcanzar los mismos objetivos, pero con distintos mecanismos. Por ejemplo, una superclase rectángulo podría tener una operación área cuyo objetivo es calcular el área del rectángulo, definida como la multiplicación de los largos de dos lados contiguos. A su vez, la clase cuadrado es una subclase de rectángulo que también tiene una operación área cuyo objetivo es calcular el área del cuadrado, pero que está definida especialmente para los objetos del tipo cuadrado como la multiplicación del largo de uno de sus lados por si mismo.

El fenómeno recién descrito se conoce como polimorfismo, y se aplica a una operación que adopta varias formas de implantación según el tipo de objeto, pero cumple siempre el mismo objetivo.

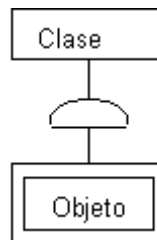
Una de las ventajas del polimorfismo es que se puede hacer una solicitud de una operación sin conocer el método que debe ser llamado. Estos detalles de la implantación quedan ocultos para el usuario; la responsabilidad descansa en el mecanismo de selección de la implantación OO.

4.7. NOTACIÓN

Para representar una clase, sus atributos y sus métodos:



Para representar una instancia de una clase:



Objeto es una instancia de la Clase. La media luna representa una jerarquía de generalización (se lee Objeto 'es un' Clase).

5. DOCUMENTACIÓN DEL ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS

5.1. DOCUMENTACIÓN DEL ANÁLISIS

5.1.1. Diagramas a Incluir

1. Diagrama de Jerarquías de Generalización
 2. Diagrama de Jerarquías Compuestas
 3. Diagrama de Relación de Objetos
 4. Ciclo Vital (por Tipo de Objeto)
 5. Diagrama de Interacción entre objetos
 6. Diagrama de Operaciones y eventos
- [los diagramas 1,2 y 3 pueden integrarse en uno: Esquema de Objetos]

5.1.2. Documentación Asociada a los Diagramas

1. Documentación de Tipo de Objeto

Nombre:

Supertipos:

Subtipos:

Descripción:

Compuesto por:

Operaciones:

2. Documentación de Operaciones

Nombre:

Descripción:

Solicitud:

Algoritmo:

5.2. DOCUMENTACIÓN DEL DISEÑO

5.2.1. Diagramas a incluir

1. Diagrama de Jerarquías de Generalización
2. Diagrama de Jerarquías Compuestas
3. Diagrama de Relación de Objetos
4. Diagrama de Interacción entre objetos

[1, 2 y 3 pueden integrarse en uno: Esquema de Objetos, además de utilizar la notación extendida para el diseño]

5.2.2. Documentación Asociada a los Diagramas

1. Documentación de Clases

Nombre: Superclases:

Subclases:

Descripción:

Atributos:

Métodos:

2. Documentación de Métodos

Nombre:

Descripción:

Parámetros:

Algoritmo:

3. Documentación de Objetos

Nombre:

Tipo:

Descripción:

Valor Atributos:

4. Documentación de Atributos

Nombre:

Componente de: (clase)

Tipo:

Restricción:

Referencia: <http://www.rhein-neckar.de/~cetus/software.html>