

Los objetos

Objetivos

- **Identificar objetos y establecer sus características.**
- **Agrupar objetos similares en Clases.**
- **Identificar relaciones de composición de objetos.**
- **Establecer relaciones de Jerarquía de Clases.**
- **Identificar mensajes entre objetos.**
- **Definir y caracterizar la Programación Orientada a Objetos.**
- **Establecer las diferencias con la programación por procesos.**
- **Definir términos básicos del enfoque de Objetos**

Introducción

En esta unidad se introduce al lector en el enfoque de objetos, el cual se constituye en la actualidad en un paradigma para el desarrollo de programas y sistemas en general. A fin de inducir en el lector los conceptos básicos del enfoque de objetos, se establece como escenario su lugar de habitación, su cuarto, y se le invita a identificar los objetos que le son familiares en su mundo cotidiano.

Se presentan, con los objetos de ese escenario, los elementos para la caracterización de los objetos que le son familiares, así como su pertenencia a grupos de características afines, que se denominan clases. Se establece a partir de dicha pertenencia la jerarquía de las diferentes clases presentadas, para a continuación ilustrar los conceptos de herencia, además se estudian las relaciones y la comunicación entre objetos para ilustrar el concepto de mensaje.

Se ilustra también el concepto de composición, mediante la observación de los diferentes componentes (que son también objetos) de algunos de los objetos tratados, como la lámpara o el equipo de sonido.

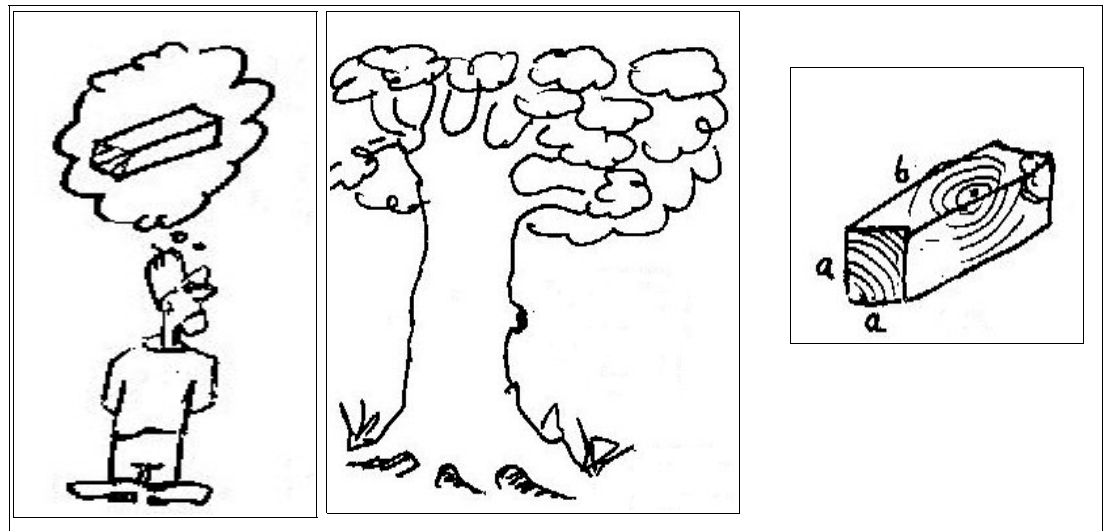
Una vez el lector se ha familiarizado con el mundo de los objetos en su escenario familiar, se presentan formalmente, los conceptos y definiciones más importantes del enfoque de objetos, los cuales servirán de base para el trabajo del desarrollo de programas en Java de las siguientes unidades.

Un problema

Nuestro alrededor esta formado por objetos de diversa índole, unos son naturales animados como los seres vivos, animales y plantas, y otros inanimados como los del mundo mineral. Los hay también artificiales como los contruidos por el ser humano o por máquinas concebidas y elaboradas por él. Vivimos dentro de una casa, en un barrio de una comuna de una ciudad, dentro de una región, que a su vez forma parte de una nación. Conglomerados de naciones se ubican y agrupan en continentes, y toda esta agregación de objetos, conforma nuestro planeta Tierra. Por donde miremos vemos objetos. Nuestro sistema planetario, las galaxias y el universo en su conjunto, son también objetos. En el mundo de lo microscópico encontramos otra dimensión de la diversidad de objetos, por ejemplo, aquellos minúsculos seres que viven en un grano de arena y pelean entre sí, a fin de tomar el dominio de una de sus aristas.

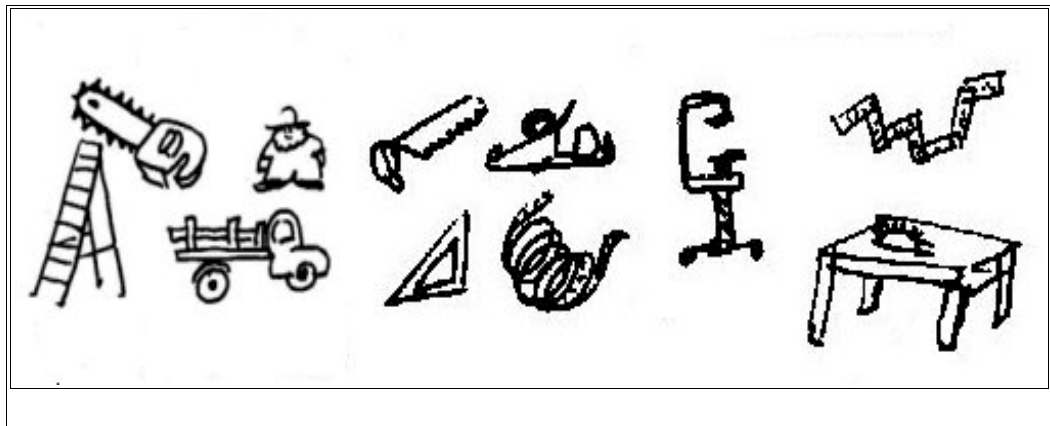
El concepto de sistemas, que no es sino otra manera de ver los objetos interactuando unos con otros en el mundo de una manera más general, nos da una idea de lo diverso y casi difícil que es enumerar los objetos en una simple lista. Estos, interactuando a fin de lograr un objetivo común, nos dan también la idea de la diversidad de sistemas y objetos existentes en el mundo que nos rodea. Estamos sumergidos en un mundo de sistemas, en un mundo de objetos. Nosotros mismos, desde el punto de vista del enfoque de objetos, somos también un objeto. Este texto que se sigue es un objeto. Veamos a continuación, una primer aproximación al enfoque de objetos en la solución de problemas.

Federico requiere de un bloque de madera de dimensiones dadas, a partir de un árbol, cuyo tronco cubre las dimensiones indicadas.



Hablando con Santiago, un experto aserrador, acerca del contrato para disponer del bloque de madera, va tomando nota de las herramientas y objetos en general que utilizarían en el desarrollo del trabajo.

Por ejemplo, algunos de las herramientas y objetos en general, incluido el aserrador, se ilustran en la siguiente figura.



Con lo cual se puede afirmar que, la solución al problema de obtener un bloque de madera a partir de un árbol, se puede obtener mediante el uso de los diferentes objetos o herramientas. Con las funciones, métodos u operaciones

inherentes a cada uno de los objetos se pueden llevar a cabo diferentes labores como por ejemplo, cortar las ramas con la motosierra, medir dimensiones con el metro, cepillar una superficie con el cepillo, etc.

Un escenario

La situación que se presenta a continuación utilizada para identificar, describir y caracterizar los objetos en la vida diaria, en el mundo real, corresponde al escenario brindado por su habitación. Se plantea un cuarto que corresponde eventualmente, al lugar de habitación de cualquiera de los lectores de este libro. Una situación hipotética, que el lector debe interpretar de acuerdo a su situación particular, a fin de poder iniciar nuestro estudio de los objetos.

Usted dispone de una confortable cama doble, dos mesitas de noche una a cada lado de la cama, un tapete persa al lado derecho de su cama, un mueble de madera para televisor sobre el cual tiene un televisor de 19" en donde observa sus programas favoritos. En la mesita de noche que se encuentra en el lado derecho de su cama, usted tiene una grabadora con CD, doble casetera, junto a esta un CD de Los Aterciopelados, uno de Los Fabulosos Cadillacs, un casete de Billy Joel, uno de Phil Collins y otros cassetes que ahora no recuerda de qué artistas son. En la mesita de noche del lado izquierdo de su cama, usted tiene una lámpara, el control remoto de su televisor y un reloj despertador. En la pared del lado izquierdo de su cama, está su armario que ocupa toda la pared, donde aparte de guardar su ropa (un blue jean, un pantalón negro, un pantalón café, una camisa blanca, una camisa negra, correa negra, correa café, entre otras), también guarda sus zapatos (unos mocasines negros, unos mocasines café, unos tenis, entre otros), además sus implementos de aseo personal (desodorante, cepillo, entre otros), también guarda la maleta del colegio que no volvió a usar, pero que conserva porque es un bello recuerdo, la camiseta de la selección deportiva en la que participó cuando tenía 10 años en el colegio, las pantuflas que le regaló su abuelita en su primera comunión, que el perro de su casa volvió añicos y que ya no se sabe de qué color son y otras cuantas cosas más que no utiliza.

Puede ser que en la vida real su habitación difiera un poco de la descripción anterior, para lo cual se le ruega nuevamente imaginar que ésta es su habitación. La figura No. 1 ilustra lo que pudiera ser su habitación para nuestro ejemplo.

Como usted puede observar, su habitación está llena de cosas, que llamaremos objetos: la cama, las dos mesitas de noche, el tapete, el mueble del televisor, el televisor, la grabadora, el CD de los Aterciopelados, el CD de los Fabulosos Cadillacs, el casete de Billy Joel y el casete de Phil Collins, los demás cassetes,

Los objetos: Un escenario 2 - 5

la lámpara, el control remoto, el reloj despertador, el armario, cada una de las prendas que conforman su ropa, los zapatos, los implementos de su uso personal, la maleta del colegio, la camiseta de la selección del colegio, las pantuflas regalo de su abuela, solo por mencionar los nombrados en la descripción de su habitación, descritos aquí. Obviamente faltarán otros objetos que están ocultos y que usted guarda celosamente.

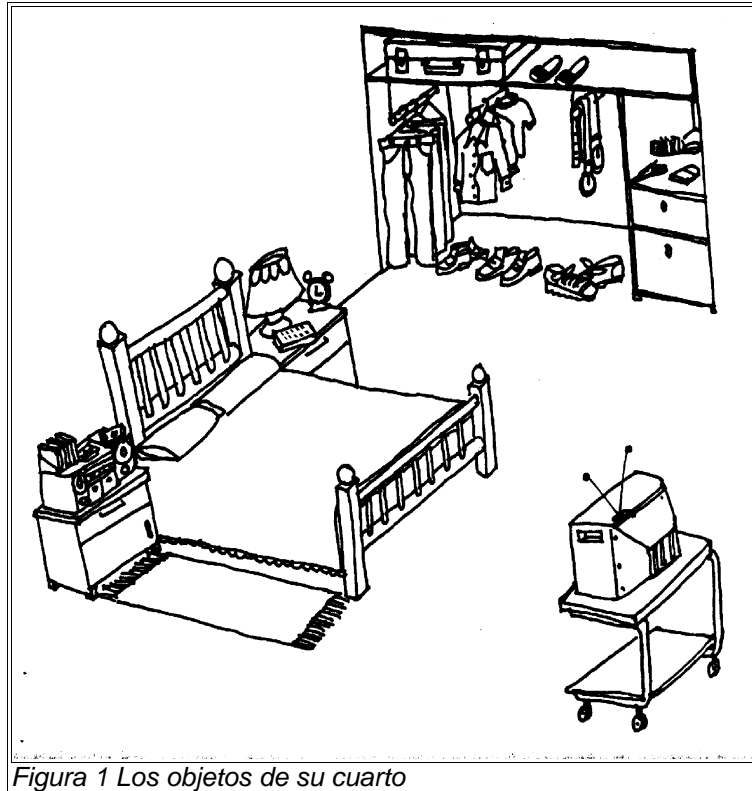


Figura 1 Los objetos de su cuarto

¿Está claro que aquellas cosas que se han tenido en cuenta del mundo de su habitación, son objetos?. Son objetos, en nuestro ejemplo, porque son tangibles e identificables y hacen parte del mundo real.

A continuación se presenta la lista numerada, de los objetos identificados en su habitación.

1. Cama	11. Lámpara	21. Tenis
2. Mesa de noche izquierda	12. Televisor 19"	22. Zapatos Mocasín negros
3. Mesa de noche derecha	13. Control remoto	23. Zapatos Mocasín café
4. Tapete persa	14. Reloj despertador	24. Maleta del colegio
5. Mueble para TV	15. Armario	25. Camiseta selección colegio

Los objetos: Un escenario 2 - 6

6. Grabadora	16. Blue jean	26. Correa negra
7. CD Aterciopelados	17. Pantalón negro	27. Correa café
8. CD Fabulosos Cadillacs	18. Pantalón café	28. Desodorante
9. Casete Billi Joel	19. Camisa blanca	29. Cepillo de cabello
10. Casete Phil Collins	20. Camisa negra	30. Pantuflas regalo de la abuela

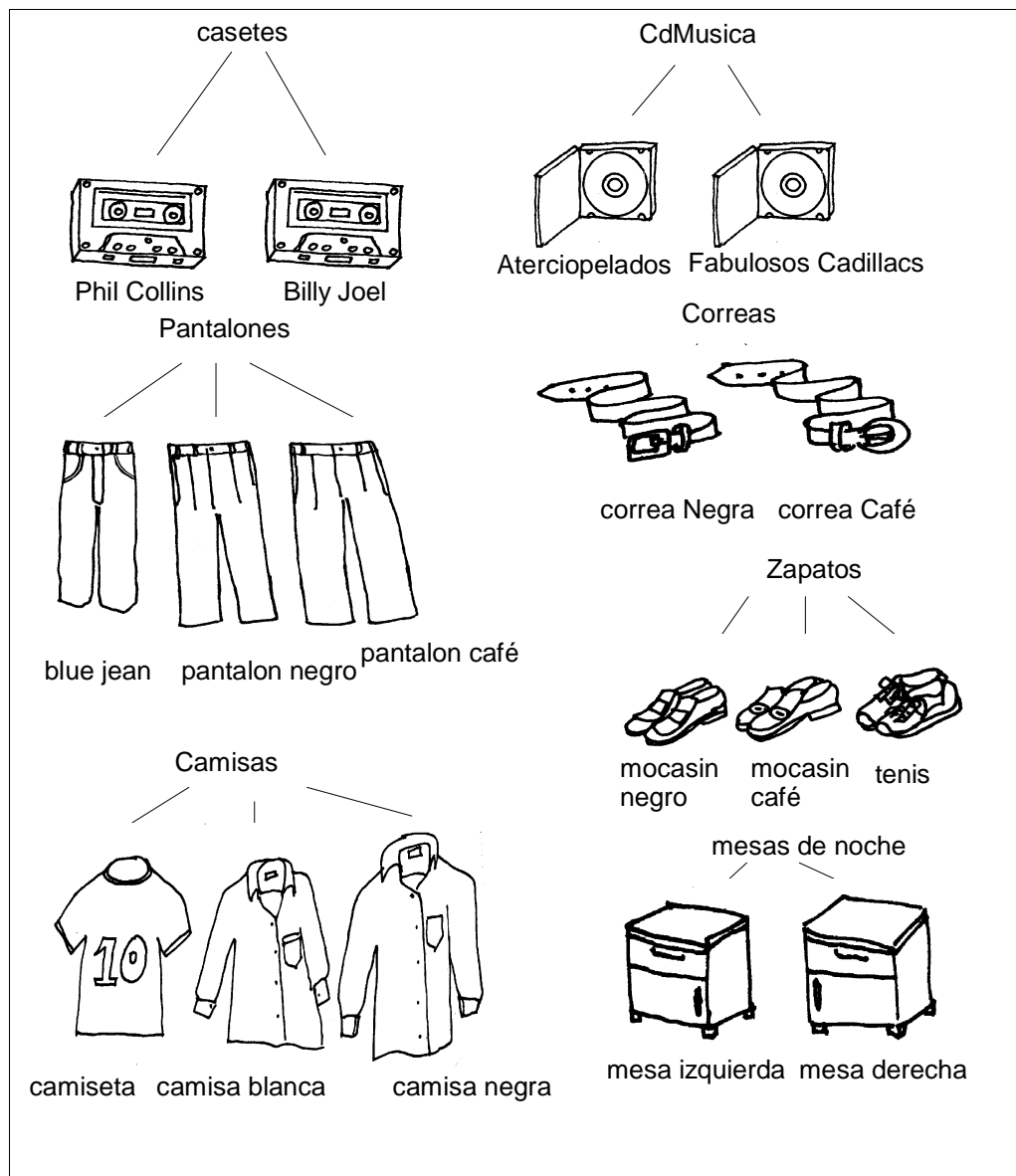
Observe que en el listado de objetos de su habitación hay objetos repetidos o similares. Por ejemplo, hay dos objetos “*mesa de noche*”, la del lado izquierdo y la del lado derecho. Ahora se identificarán en el siguiente cuadro, los objetos similares que se han identificado en su habitación, agrupados según sus características generales, indicando si está o no repetido:

Nombre objeto	Repetido o nó
Cama	Objeto no repetido
mesaDeNocheIzquierda	Estos objetos son Mesas de noche, por lo tanto podemos agruparlas en el conjunto MesaDeNoche
mesaDeNocheDerecha	
Tapete	Objeto no repetido
Mueble para TV	Objeto no repetido
Grabadora	Objeto no repetido
CD de los Aterciopelados	Estos objetos son CD de música, por lo tanto podemos agruparlos en el conjunto CdDeMusica
CD de los Fabulosos Cadillacs	
Casete de Phil Collins	Estos objetos casetes, por lo tanto podemos agruparlos en el conjunto Casete
Casete de Billy Joel	
Lámpara	Objeto no repetido
Televisor 19"	Objeto no repetido
Control remoto	Objeto no repetido
Reloj despertador	Objeto no repetido
Armario	Objeto no repetido
BluJean	Estos objetos son Pantalones, por lo tanto podemos agruparlos en el conjunto Pantalón
Pantalón negro	
Pantalón café	
camisa blanca	Estos objetos son camisas, por lo tanto podemos agruparlas en el conjunto Camisa
camisa negra	
Camiseta selección colegio	
Zapatos Mocasín negros	Estos objetos son Zapatos, por lo tanto podemos agruparlos en el conjunto Zapatos
Zapatos Mocasín café	
Tenis	
Correa negra	Estos objetos son Correas, por lo tanto podemos agruparlas en el conjunto Correa
Correa café	
Desodorante	Objeto no repetido
Cepillo de cabello	Objeto no repetido

Los objetos: Un escenario 2 - 7

Nombre objeto	Repetido o no
Maleta del colegio	Objeto no repetido
Pantuflos regalo abuela	Objeto no repetido

Con lo anterior, se han identificado y agrupado los objetos que presentan características similares en conjuntos que los representan. Los objetos que no se han agrupado se debe a que no están repetidos en su habitación. Las siguientes figuras ilustran los objetos repetidos, agrupados según sus características generales:



En el Enfoque de Objetos, la agrupación de aquellos objetos que tienen características similares, se denomina **Clase**. La **Clase** es entonces, una forma de referirse a agrupaciones de objetos que tienen características similares.

A continuación, se listan las **Clases** identificadas hasta ahora, correspondientes a los objetos repetidos asignándoles un nombre significativo:

Clases de Objetos
MesaDeNoche
CdDeMusica
Casete
Pantalón
Camisa
Zapatos
Correa

Pero, ¿qué pasa con los demás objetos dentro de la habitación? ¿Acaso ellos son únicos en el mundo y no tienen un similar con quién agruparse y formar parte de una clase? El resto de objetos de su habitación que no están repetidos, son únicos, pero en su habitación. Sin embargo, es posible que, en el resto de su casa existan otros similares, por ejemplo otro televisor, otras camas, otras lámparas de noche, otros relojes despertadores, etc. Así que, aunque estos objetos son únicos en su cuarto, fuera de él y en su misma casa pueden existir otros similares.

Tomemos el caso de su cama doble, a fin de identificar si es posible incluirla dentro de una clase, en donde junto con otros objetos con características similares se la pueda agrupar:

El día en que sus padres le regalaron la cama doble, fue justamente en el que usted cumplió 16 años de edad. Días antes, ellos fueron a una mueblería o almacén de muebles y escogieron su cama a través de un catálogo, la encargaron, la pagaron y como por arte de magia la cama llegó a su casa el día de su cumpleaños. Pues bien, cuando sus padres la escogieron del catálogo, pidieron al vendedor de la mueblería una cama similar a la que estaban viendo en el catálogo y cuya referencia era DC-201. El vendedor elaboró una factura de venta, un original lo dió a sus padres y la otra la mandó a la fábrica, en donde carpinteros buscaron la referencia en su libro de diseño de productos, en el cual

existe a manera de guía los pasos, los patrones y las características para la “elaboración de una Cama Doble DC-201”. Tomaron los diseños respectivos, los que les permitirían ir construyendo la cama y empezaron a elaborarla, hasta disponer de una réplica igual a la indicada en el catálogo.

Si cuando los carpinteros terminaron su cama, les hubiese llegado otra copia de factura con un pedido de elaboración igual al que hicieron sus padres, “Cama doble DC-201”, tenga por seguro que, si siguen las instrucciones del diseño al pie de la letra, la cama de éste pedido queda igual a la suya. Luego su cama fue elaborada a partir de un diseño original y como ella deben existir al día de hoy muchas camas similares repartidas en otras casas por toda la ciudad, posiblemente dentro de su misma casa.

Es decir, su objeto Cama Doble referencia DC-201, también tiene objetos similares que se encuentran en otras residencias en la ciudad, o en otros cuartos de su casa y por tanto, su cama puede agruparse también dentro del conjunto o Clase **Camas Dobles Referencia DC-201**. Entonces, aunque en su habitación solo hay un solo objeto de la clase Cama Doble, debe tener en cuenta que su cama hace parte de este conjunto o clase, la que se denominará **Cama Doble**

El ejercicio que se ha hecho con el objeto Cama Doble, tiene como fin ayudarlo a comprender más el término Clase. Si realiza el ejercicio con los demás objetos no repetidos, que aún no se han catalogado en un conjunto, se dará cuenta que cada uno de ellos pertenece a un conjunto del cual ellos son un objeto particular y que ese conjunto es una Clase.

Siguiendo el mismo proceso para los demás objetos presentes en su habitación, se puede elaborar entonces, el listado de objetos con la identificación de su respectiva clase.

Para la denominación de nombres de objetos o identificadores de nombres de objetos, se sigue la regla de utilizar sustantivos que sean significativos, con la condición que la primera palabra es toda con minúsculas y las siguientes palabras sin espacio entre ellas, se inician con mayúscula, las demás letras son minúsculas. Los identificadores o nombres para denominar las clases siguen la misma regla con la condición que la primera palabra se inicia con mayúscula.

La siguiente tabla ilustra ese listado de objetos, con su nombre y su agrupación en la clase respectiva:

Los objetos: Un escenario 2 - 10

Nombre objeto	Nombre de la clase
camaDoble	Este objeto es fiel copia del diseño Camas Dobles DC-201, luego se encuentra dentro del conjunto <i>CamaDoble</i>
mesaDeNocheIzquierda	Estos objetos son Mesas de noche, por lo tanto podemos agruparlas en el conjunto <i>MesaDeNoche</i>
mesaDeNocheDerecha	
tapetePersa	Este objeto es una fiel copia del diseño de Tapetes Persas, luego se encuentra dentro del conjunto <i>Tapete</i>
muebleTv	Este objeto es una fiel copia del diseño de Muebles para TV DC-176, luego se encuentra dentro del conjunto <i>MuebleParaTV</i>
grabadora	Este objeto es una fiel copia del diseño de Grabadoras S-21, luego se encuentra dentro del conjunto <i>Grabadora</i>
cdLosAterciopelados	Estos objetos son CD de música, por lo tanto podemos agruparlos en el conjunto <i>CdDeMusica</i>
cdLosFabulososCadillacs	
caseteDeBillyJoel	Estos objetos son Casetes, por lo tanto podemos agruparlos en el conjunto <i>casete</i>
caseteDePhilCollins	
lámpara	Este objeto es una fiel del diseño de lámparas, luego se encuentra dentro del conjunto <i>Lámpara</i>
televisorDe19Pulgadas	Este objeto es una copia del diseño de Televisores 19", luego se encuentra dentro del conjunto <i>Televisor</i>
controlRemoto	Este objeto es una copia del diseño de Controles Remoto para Televisores de 19", luego se encuentra dentro del conjunto <i>ControlRemoto</i>
relojDespertador	Este objeto es una copia del diseño de Relojes Despertadores, luego se encuentra dentro del conjunto <i>RelojDespertador</i>
armario	Este objeto es una copia del diseño de Armarios, luego se encuentra dentro del conjunto <i>Armario</i>
blueJean	Estos objetos son Pantalones, por lo tanto podemos agruparlos en el conjunto <i>Pantalón</i>
pantalónNegro	
pantalónCafé	
camisaBlanca	Estos objetos son Camisas, por lo tanto podemos agruparlas en el conjunto <i>Camisa</i>
camisaNegra	
tenis	Estos objetos son Zapatos, por lo tanto podemos agruparlos en el conjunto <i>Zapato</i>
zapatosMocasínNegros	
zapatosMocasínCafé	
correaNegra	Estos objetos son Correas, por lo tanto podemos agruparlas en el conjunto <i>Correa</i>
correaCafé	
desodorante	Este objeto es una copia de la receta de Desodorantes, luego se encuentra dentro del conjunto <i>Desodorante</i>
cepilloDeCabello	Este objeto es una copia del diseño de Cepillos para Cabello, luego se encuentra dentro del conjunto <i>CepilloDeCabello</i>
maletaDelColegio	Este objeto es una copia del diseño de Maletas de Colegio, luego se encuentra dentro del conjunto <i>MaletaDeColegio</i>
camisetaSelecciónColegio	Este objeto es una copia del diseño de Camisetas de Selección Colegio, luego se encuentra dentro del conjunto <i>Camiseta</i>
pantuflasRegaloAbuela	Este objeto es una copia del diseño de Pantuflas, luego se encuentra dentro del conjunto <i>Pantufla</i>

A esta altura de nuestro estudio sobre objetos, ya se debe tener claro que los objetos dentro de una misma clase tienen características similares. Dichas

características se deben encontrar definidas en la clase que las agrupa, ya que la clase es la definición en sí de los objetos que se agrupan en un mismo conjunto.

La clase define las características comunes a todos los objetos que la conforman. Por ejemplo, aunque la clase Zapato agrupa a todos los objetos con características similares, cada objeto en particular, por ejemplo un par de tenis, tiene valores específicos o propios para cada característica. Todos los zapatos tienen la característica color, en los suyos el color de los tenis es blanco por ejemplo. La clase posee la característica color, el objeto en particular, tiene un valor específico para dicha característica.

Ahora se tomará una clase cualquiera de las enumeradas antes y se identificarán las características comunes a todos los objetos que la conforman. Sea el caso de la clase CdDeMusica, a la cual pertenecen los CD de música que usted tiene junto a su grabadora. Estos CD de música tienen, entre otras, las siguientes características comunes:

Característica
Título del CD
Nombre del artista
Año de grabación ó producción
Nombre del productor fonográfico
Lista de canciones
Número de serie

Aunque un CD puede tener mas características, como tamaño, duración y otras, se tomarán las indicadas como las características a tener en cuenta para la clase CdDeMusica. Estas características como son comunes a cada CD de música, entonces deben estar contenidas en la definición del conjunto que las agrupa, o sea la Clase CdDeMusica.

Cuando se define la clase, se definen las **características** que van a tener todos los objetos comunes a esa clase. Por lo tanto, la clase CdDeMusica, tendrá como características las mencionadas anteriormente y que señalamos a continuación con la notación que se seguirá de ahora en adelante cuando se haga referencia a las características como tal. A cada una de estas características se les denomina **atributo**.

Clase: CdDeMusica	
Nombre Atributo	Notación de Atributo
Título CD	tituloCD
Nombre Artista	nombreArtista
Año Grabación	añoGrabacion
Nombre Productor Fonográfico	nombreProductorFonográfico
Lista de Canciones	listaDeCanciones
Numero de Serie	numeroSerie

Esta notación para los atributos de las clases, similar a la seguida en lenguajes orientados a objetos, como es el caso del Java, recomienda que se utilicen sustantivos compuestos que claramente definan el atributo. Por eso la notación propone construir el nombre del atributo a partir de las palabras que lo definen, de las cuales la primera comienza con minúscula y la primera letra de cada una de las palabras siguientes en mayúscula. Esta notación se usará de ahora en adelante para los atributos.

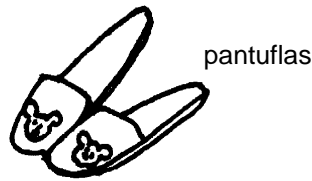
Como ya se ha mencionado antes, los atributos que se definen en la clase, toman valores en cada uno de los objetos que hacen parte de ella. En el siguiente cuadro se han tomado solamente dos objetos de la Clase CdDeMusica, a fin de identificar sus atributos:

Atributos Clase	Atributos cdLosAterciopelados	Atributos cdLosFabulosos Cadillacs
tituloCD	La Pipa de la Faz	En la Vida
nombreArtista	Los Aterciopelados	Fabulosos Cadillacs
añoGrabación	1994	1992
nombreProductor	EMI Sora	Fonolux
listaDeCanciones	Paracutanara, El Cofre, Florecita Salsera, Mujer Bala, La Pipa de la Faz, Camino al Dorado, Me siento Colombiano, Solsticio, Bolero Sagaz	Mal Dicho, En la Vida, Cazador, Vasos Llenos, Matasor, Canción 2, Canción 3, Canción 4, Canción 5, Canción 6
numeroSerie	24352-46552	28475-69875

Cada objeto de la clase CdDeMusica tiene su definición individual de sus atributos, los cuales toman valores particulares que lo hacen diferente y distinguible de otros. Esto se observa claramente en la tabla anterior. En la columna 1 tenemos definidos los atributos de la clase CdDeMusica. En la columna 2, estos atributos están presentes en el objeto cdLosAterciopelados, con los valores que hacen único a este objeto. Lo mismo sucede con la columna 3, representada por los atributos del objeto cdLosFabulososCadillacs, el cual tiene valores propios para estos atributos. En este sentido, en los objetos, a diferencia de las clases, los atributos dejan de ser una definición de características para convertirse en el valor propio que representa la característica dentro del objeto como tal.

Ahora bien, si se tuviera dos CD exactamente iguales, es decir que fueran copias iguales, sería necesario adicionar un nuevo atributo que diferenciara las dos copias, por ejemplo un número consecutivo para indicar el número de la copia. En cuyo caso en la clase CdDeMusica se debería tener en cuenta esa característica.

Pasemos ahora al objeto pantuflasRegaloDeAbuela, el cual pertenece a la clase Pantuflas. Los atributos que se podrían encontrar para la clase Pantuflas y con los cuales se trabajará aquí son: Tipo Pantuflas, material de Fabricación, Talla, nombre Dueño,color.



De esta manera, su objeto pantuflasRegaloAbuela, toma los siguientes valores para sus atributos:

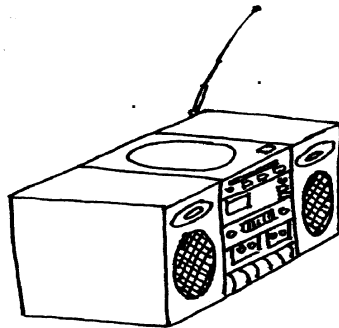
<i>Atributos Clase Pantufla</i>	<i>Nomenclatura para los atributos de la Clase Pantufla</i>	<i>Valores de los atributos del objeto pantuflasRegaloAbuela</i>
Tipo Pantuflas	tipoPantufla	De Batalla
Material Fabricación	materialFabricacion	Peluche
Talla	talla	39 ½
Nombre Dueño	nombreDueño	Usted
Color	color	blanco

Entonces, se podría decir que las clases son la representación general de los objetos que se encuentran agrupados en ella. En la terminología orientada a objetos, a éstos se les denomina también **instancias** de la clase a la que pertenecen. Así pantuflasRegaloAbuela es una instancia de la clase Pantuflas. Los términos objeto e instancia son sinónimos y se utilizarán indistintamente.

Se ha dicho que en las clases se definen las características comunes de los objetos que pertenecen a ella. Que estas características se denominan **Atributos** y que éstos atributos toman valores únicos en un objeto particular. La forma como se organizan los atributos en la definición de una clase, determinan lo que se denomina la **Estructura** de la clase.

Tomemos ahora por ejemplo, el objeto grabadora. Es posible tener como

nombre de una instancia el mismo nombre de la clase, como en el caso del objeto grabadora y el de la clase Grabadora. La notación del identificador para objetos y clases indica claramente cual es el objeto y cual es la clase, por tanto los nombres realmente son diferentes; el de la clase inicia con mayúscula y el del objeto con minúscula.



Algunos de los atributos que se podrían encontrar en la clase Grabadora y los valores que esos atributos toman en el objeto grabadora, se muestra a continuación.

Atributos Clase Grabadora	Notación atributos de la clase Grabadora	Valores para los atributos del objeto Grabadora
Marca	marca	Gato
Referencia	referencia	S-21
Dueño	dueño	Usted

Puede observarse que, a diferencia de las pantuflas, de la mesa de noche, de los tenis y los bluejeans por ejemplo, el objeto grabadora realiza operaciones propias que es posible identificar, como reproducir Cd, reproducir casete, aumentar volumen, apagarse y otras, según indicaciones dadas por usted, presionando un botón de la grabadora, o accionando el control remoto. Estas funciones u operaciones que realizan los objetos se denominan **comportamiento**. Así como todos los objetos tienen nombre y atributos, todo objeto tiene comportamiento, aunque no parezca evidente como en el caso de los tenis, las pantuflas o una camisa. Obviamente la grabadora realiza funciones u operaciones, por eso se toma como ejemplo.

En estos términos, las clases como agrupaciones de objetos tienen estructura y comportamiento: la **estructura** define qué características van a tener los objetos agrupados en la clase y el **comportamiento** define las funciones u operaciones

que van a realizar esos objetos. Dichas funciones se denominan **métodos**.

Ya definida una **estructura** para la clase Grabadora, basada en algunos de sus atributos, centremos la atención ahora en enumerar cuáles son algunas de las funciones ó métodos que realiza su grabadora y que determinan su **comportamiento**.

Su grabadora reproduce el sonido de un CD, reproduce el sonido de un casete, graba de CD a casete, graba de casete a casete, sintoniza emisoras, amplifica ondas de radio, se enciende y apaga, aumenta o reduce el volumen del sonido, entre otras funciones.

Estas funciones del objeto grabadora se ofrecen gracias a las operaciones y dispositivos que se han puesto en la grabadora cuando fue ensamblada. A estas operaciones que implementan las funciones que realiza un objeto y que constituyen el comportamiento de un objeto se les denominan **métodos**. Estas funciones las ofrecen todas las grabadoras que como la suya son de la referencia S-21. Luego estas funciones las realizan todos los objetos que están dentro de la clase Grabadora que se ha definido y por tanto, se incluirán éstas funciones dentro de la definición de la clase Grabadora, como aparece en el siguiente cuadro:

Métodos de la Clase Grabadora	Notación métodos de la clase Grabadora
Reproducir sonido CD	reproducirSonidoCD
Reproducir sonido casete	reproducirSonidoCasete
Grabar de CD a casete	grabarDeCDaCasete
Grabar de casete a casete	grabarDeCaseteACasete
Sintonizar emisora	sintonizarEmisora
Reproducir sonido Emisora	reproducirSonidoEmisora

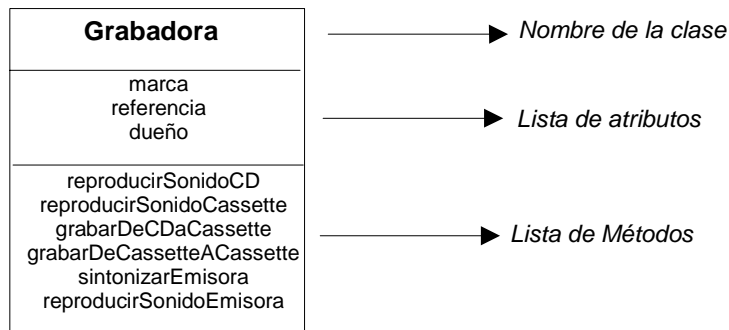
La notación que se sigue en la columna 2 de la tabla anterior para referirse a los métodos de una clase, recomienda que se utilicen nombres compuestos que comiencen con el verbo que represente la función que el método va utilizar en minúscula, seguido del ó los sustantivos que ayuden claramente a definir la acción del método. Cada uno de estos sustantivos que siguen al verbo, deben empezar por mayúscula. Esta notación la usaremos en lo sucesivo cuando nos refiramos a métodos de una clase.

Para la clase Grabadora en términos de **Estructura** (atributos) se tendrán en cuenta entre otros, los siguientes: marca, referencia y dueño.

Para su **Comportamiento** (métodos) se tendrán en cuenta los siguientes considerados como básicos: reproducir sonido de CD, reproducir sonido de casete, grabar de CD a casete, grabar de casete a casete, sintonizar emisora, reproducir sonido emisora.



De ahora en adelante cuando se vaya a referir a la estructura y el comportamiento de una clase, de manera gráfica, en lugar de utilizar la representación del cuadro anterior, se utilizará notación estándar siguiente:



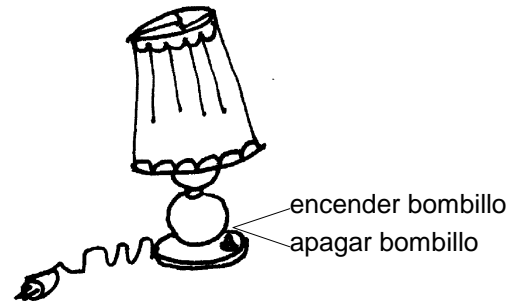
La notación para ilustrar la definición de una clase, comprende entonces tres áreas o compartimientos: en la parte superior va el nombre de la clase, en la intermedia van los atributos y en la parte inferior los métodos.

Siguiendo un recorrido por los objetos de su habitación, tomemos ahora el objeto lámpara que tiene sobre la mesita del lado izquierdo de su cama. Definamos los atributos y los métodos de la clase LámparaDeMesaDeNoche.

La lámpara puede tener como atributos, la marca, el nombre, la referencia, el

tipo de bombillo que utiliza, el voltaje, entre otros. Si se piensa en cuáles son las funciones que la lámpara realiza, se podría afirmar que ella permite encender y apagar el bombillo. Luego se puede afirmar que, los métodos de la lámpara son: `encenderBombillo` y `apagarBombillo`. Siguiendo la notación gráfica indicada antes, la clase `LámparaDeMesaDeNoche` quedaría representada así:

LamparaDeMesaNoche
<div>marca</div> <div>nombreLampara</div> <div>referencia</div> <div>tipoDeBombilloRequerido</div>
<div>encenderBombillo</div> <div>apagarBombillo</div>



Bueno, entonces ya se tiene claro que las clases poseen estructura y comportamiento, los cuales son extensibles a los objetos representados. La estructura la determina los atributos y el comportamiento está definido por los métodos que implementan las funciones que realizan los objetos agrupados en esta clase.

Organización y relación entre Clases

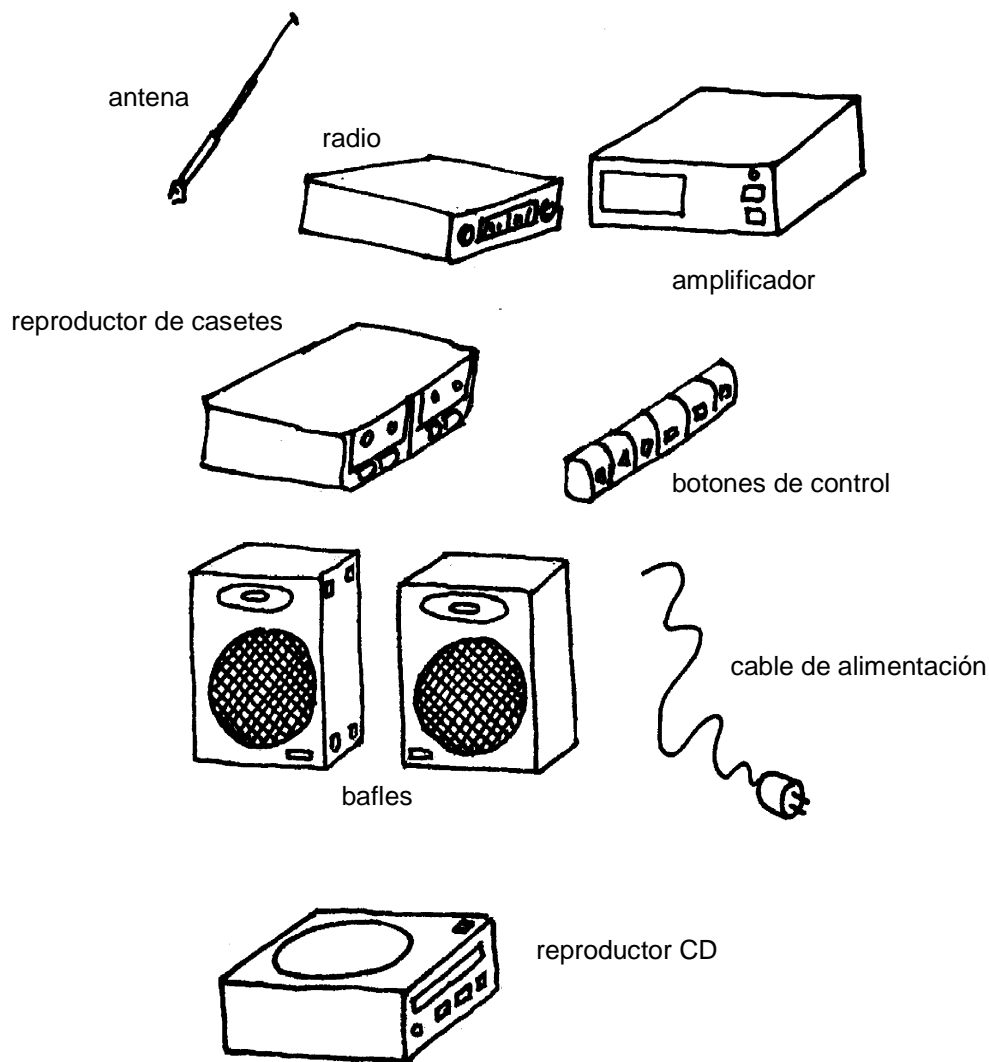
En la sección anterior, se identificó y definió la estructura y el comportamiento de la clases `Grabadora` y `LamparaDeMesaDeNoche`. Observando en detalle el objeto grabadora, se puede reconocer que la grabadora tiene un lector de CD, dos baffles, radio, amplificador, dos caseteras, una antena, un cable para conectar al toma-corriente, 11 botones para controlar las caseteras, 5 botones para controlar el reproductor de CD, un botón controlador del volumen y un botón controlador de cambio de función (el que permite, prender, pasar de casete a CD ó a radio), entre otros.

Observe detenidamente los elementos de la siguiente figura, que corresponden a una desagregación de los principales componentes de su grabadora. ¿No le parece que esos componentes son también objetos? De cada uno de ellos existen otros en otras grabadoras, tienen nombre, atributos y realizan funciones.

Se puede afirmar entonces que, en el mundo real, existen objetos que se

componen de otros objetos, o en otras palabras, que son agregado de muchas partes independientes que son indispensables dentro del objeto compuesto. Pues esas partes independientes que forman el objeto, en este caso su grabadora, se pueden distinguir como objetos también, pues dentro del mundo real, también lo son.

Los diferentes objetos, componentes de una grabadora, se pueden conseguir por separado y ensamblar a su gusto. Por ejemplo, cuando se daña un parlante de ésta, se retira y se cambia por uno nuevo que se adquiere en un almacén de artículos electrónicos.

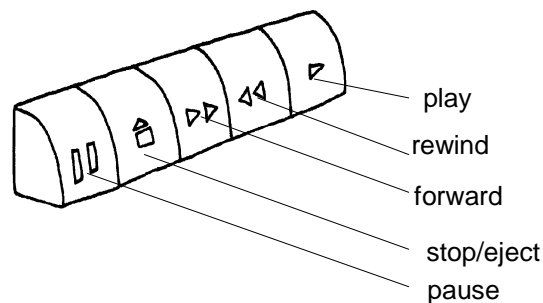


Cada uno de estos elementos que hacen parte del objeto su grabadora y que caracterizan a las grabadoras agrupadas en la Clase Grabadora referencia S-21,

Los objetos: Organización y relación entre Clases 2 - 19

son elementos de alguna manera independientes que, juntos constituyen la grabadora. Luego se podría decir que el objeto su grabadora se constituye de los anteriores elementos indicados, los cuales a su vez, son objetos que existen en el mundo real. Es decir, se tiene un objeto: su grabadora, que se compone de otros objetos. Cada uno de esos objetos existe en el mundo real y como tales, deben estar agrupados en una clase, pues no son los únicos objetos con esas características que existen en el mundo real.

Por ejemplo, los 5 **botones** que permiten operar el reproductor de CD, están agrupados dentro de un conjunto o clase **BotonesCD** y por lo tanto cada uno de ellos debería tener la misma estructura y similar comportamiento.



Miremos cómo podría ser una estructura para la clase Botón y qué valores podrían tomar los 5 botones del reproductor de CD, del objeto grabadora, en el cuadro siguiente:

Atributos Clase Botón esCD del Reproductor de CD	Valores para el botón Parar/Abrir (stop/eject)	Valores para el botón Pausa (pause)	Valores para el botón Adelantar (Ffwd)	Valores para el botón Devolver (rew)	Valores para el botón Reproducir (play)
nombre	Parar/Abrir	Pausa	Adelantar	Devolver	Reproducir
etiqueta	Stop/Eject	Pause	Ffwd	Rew	Play
icono					
tamaño	10x10	10x10	10x10	10x10	10x10
posición	20,20	40,20	60,20	80,20	100,20

El comportamiento de cada uno de los objetos botón, corresponde a la función que realiza y que se activa cuando se presiona o se libera sobre el icono o símbolo respectivo.

Los objetos: Organización y relación entre Clases 2 - 20

Con el ejemplo de la grabadora y su conformación por medio de diferentes objetos, como los indicados en la figura, y especialmente para el caso de los botones del reproductor de CD, se ilustra que existen objetos que están compuestos de otros objetos. En el enfoque de objetos, cuando un objeto se forma a partir de otros objetos, como en el caso de la grabadora, la lámpara, el televisor, y otros en su cuarto, se realiza una **composición**.

Cuando existe composición entre objetos, es decir, cuando un objeto hace parte de otro, entonces el objeto compuesto se caracteriza también por ser la agrupación de uno o varios objetos. Esta caracterización que lo hace ser único como objeto, hace parte de su definición como objeto y por tanto hace parte de la definición de la clase. En este sentido, la **composición** es referenciada dentro del cuerpo de la estructura de la clase que representa al objeto compuesto.

Cada uno de los objetos que es una parte del objeto compuesto, se convierte en un atributo de la clase correspondiente del objeto compuesto. Tal como en el mundo real, la **composición** caracteriza a los objetos en el enfoque de objetos. Por lo tanto, estos atributos que representan objetos, se ponen como atributos del tipo objetos, pertenecientes a la estructura de una clase particular.

De esta manera, la estructura de la clase Grabadora incluyendo la composición basada en algunos de los objetos que la conforman, queda de acuerdo a lo ilustrado en el siguiente cuadro:

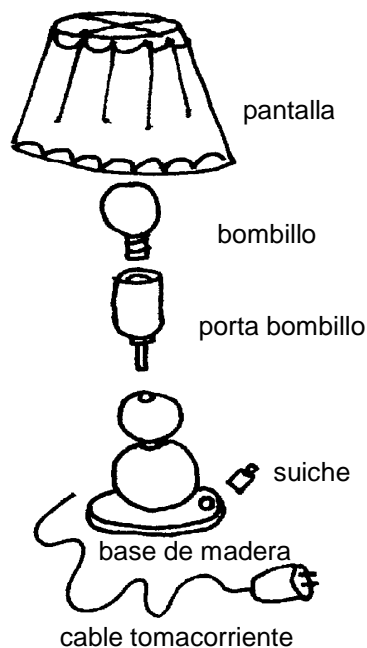
Atributos Clase Grabadora	Notación atributos de la clase Grabadora	Valores para los atributos del objeto Grabadora
marca	marca	GatoSilvestre
referencia	referencia	S-21
dueño	dueño	Usted
Contiene objeto Lector CD	objetoLectorCD	LectorCD para grabadora S21, serie 00012
Contiene 2 objetos caseteras	parObjetosCaseteras	Casetera 1- serie 09373 y Casetera 2-serie 09374
Cable al toma-corriente	objetoCableTomaCorriente	Cable Toma corriente serie 388734
antena	objetoAntena	Antena serie 244
sintonizador de radio	objetoRadio	Radio Zenith 123
amplificador	objetoAmplificador	Amplificador Aiwa 345
botones para controlar las caseteras	listaBotonesCaseteras	Boton Stop serie23, Boton Pause serie 23, boton Ffwd serie 23, boton Rew serie 23, botón Play serie 23, botón segunda casetera Stop serie 24, botón segunda casetera Pause serie 24, botón segunda casetera Ffwd serie 24, botón segunda casetera Rew serie 24, botón segunda casetera Play serie 24, botón segunda casetera Rec serie 34
botones para controlar el reproductor de CD	listaDeBotonesCD	Botón Stop-Cd serie 45, Botón pause-Cd serie 45, Botón Ffwd-Cd serie 45, botón Rew-Cd serie 45, Botón Play-Cd serie 45

Los objetos: Organización y relación entre Clases 2 - 21

Atributos Clase Grabadora	Notación atributos de la clase Grabadora	Valores para los atributos del objeto Grabadora
botón controlador del volumen	botonControladorVolumen	Botón Controlador Volúmen serie 23
botón controlador de cambio de función	botonCambioFuncion	Botón Cambio Función serie 23

Ahora seleccionemos otro de los objetos de su habitación, por ejemplo, la lámpara que tiene sobre la mesita de noche del lado izquierdo.

Esta lámpara está compuesta de una base de madera formada por dos esferas y una tabla circular de madera, por el centro de ellas pasa un tubo de aluminio de 0,5 centímetros de diámetro, tiene además arandelas y tuercas de fijación, un bombillo, un cable toma corriente, el portabombillo, el suiche y la pantalla. Cada una de estas partes son objetos también y éstos objetos pueden estar agrupados en clases y por lo tanto tienen estructura y comportamiento definidos. Luego estamos hablando también de **composición** para la clase `LamparaDeMesaDeNoche`.



La siguiente tabla ilustra los componentes de la clase `LamparaDeMesaDeNoche`:

Los objetos: Organización y relación entre Clases 2 - 22

Atributos Clase LamparaDeMesaDeNoche	Notación atributos clase LamparaDeMesaDeNoche	Valores para los atributos del objeto lampara
marca	marca	Particular Electric
referencia	referencia	PE-56A
dueño	dueño	Usted
voltaje	voltajeAlimentación	120 V
Contiene objeto pantalla	objetoPantalla	Pantalla, aluminio-tela, S453
Contiene objeto bombillo	bombilloLampara	SLR-45 70W
Contiene objeto portalampara	objetoPortalampara	Buho Fg-987
Contiene objeto base de madera	baseDeMadera	Base DF-45
Contiene objeto suiche	suicheLampara	Sw-843
Contiene cable-tomacorriente	cableTomacorriente	cable awt-15

Centremos ahora nuestra atención en los objetos que tiene dentro de su armario, pero no se preocupe no le vamos a preguntar nada por aquellos objetos ocultos y privados.

Vamos a tomar los objetos con los cuales usted calza sus pies. Para este ejemplo usted tiene un par de tenis, un par de zapatos mocasines de color café y otro par de mocasines de color negro. Los objetos zapatos mocasines de color café y zapatos mocasines de color negro, están agrupados en la clase **ParDeZapatos**. Como atributos de la clase ParDeZapatos se pueden distinguir: numero, marca, tipo, color, tipo cuero, entre otros.



La tabla correspondiente a los atributos de la clase ParDeZapatos y los valores que toma en los objetos zapatos mocasines de color café y zapatos mocasines de color negro, se muestra a continuación:

Atributos Clase ParDeZapatos en notación	Valores de atributos para el objeto ParZapatos MocasinCafé	Valores de atributos para el objeto ParZapatosMocasinNegro
numero	39	39
marca	CopaShoes	TresPatras
tipo	clásico	Normal
color	café	negro
tipoMaterial	normal	Gamo
tipoHebilla	ninguna	Pequeña-plata

El objeto Par de Zapatos Tenis, está agrupado dentro de la clase ParDeZapatosTenis, la cual se puede definir con los siguientes atributos y con los siguientes valores de atributos correspondientes al objeto su par de zapatos tenis, tal como muestra el cuadro siguiente:



Atributos Clase ParDeZapatosTenis	Valores de atributos para el objeto zapatos Tenis
numero	39
marca	Patitas
tipo	para Trotar
color	blanco y azul
tipoMaterial	cuero y tela
tipoCordon	Esférico-largo

Tanto el objeto par de Zapatos Tenis como los objetos Par de Zapatos Mocasín, son zapatos porque le permiten calzar sus pies, sin embargo ellos se diferencian por ejemplo en que los zapatos tenis le sirven para hacer deporte en cambio sus mocasines solo le sirven para vestir. Como obviamente ambos tipos de objetos son Pares de Zapatos, entonces ellos deberían estar agrupados en una clase común que los incluye a ambos, la clase Par de Zapatos. Es decir, al ser zapatos las clases Par de Zapatos Tenis y Par de Zapatos Mocasín, tienen atributos similares, pero en cambio difieren en que, la clase Zapato Tenis tiene un atributo tipoCordon que la clase Mocasín no tiene y la clase Mocasín tiene un atributo tipoHebilla que la clase Tenis no tiene. Por supuesto los Mocasines no tienen cordones, pero en lo demás ambas clases son similares. Entonces se debe crear una clase más general que se llame Clase ParDeZapatos, que contenga los atributos que tanto la clase Par de Zapato Tenis como la clase Par de Zapato Mocasín poseen. A continuación se relacionan en la siguiente tabla los atributos que va a tener la clase Par de Zapatos, que es prácticamente común a cualquier par de zapatos:

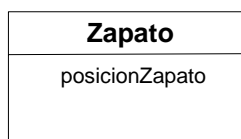
ParDeZapatos
numero marca tipo color tipoMaterial

Como se está hablando de atributos y no se está tratando el tema de los métodos de esta clase, entonces el compartimiento respectivo a los métodos no aparece en el gráfico.

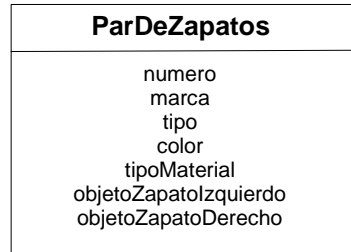
Como se sabe, un objeto parDeZapatos cualquiera que sea, se compone de un objeto zapatoDerecho y de un objeto zapatoIzquierdo. Debe observarse que, a la definición de la clase ParDeZapatos ilustrada en la figura anterior, le faltan los atributos correspondientes a los objetos que componen la clase ParDeZapatos. En este caso, su objeto zapatoDerecho y su objeto zapatoIzquierdo.

Al ser tanto zapatoIzquierdo como zapatoDerecho objetos, entonces deben pertenecer a una clase, justamente a la clase Zapato. De esta manera, la clase Par de Zapatos está compuesta de dos objetos de la clase Zapato y por lo tanto hay una relación de composición entre ambas clases. Esta relación de composición se traduce dentro de la clase ParDeZapatos como dos atributos dentro de ella que son instancias de la clase Zapato o que en otras palabras son de tipo Zapato. La clase Zapato entonces tendrá un atributo posicionZapato que define si un objeto particular de la clase es un zapato izquierdo o derecho} que definen el objeto zapatoIzquierdo y zapatoDerecho, que es más general que la clase ParDeZapatos, y que agrupa las características que los representan a los dos. Dichas características en este caso son solo una: posiciónZapato, es decir si es el izquierdo o el derecho.

Usted se podría preguntar, porqué se definieron las clases Zapato y ParDeZapatos por separado y no se buscó la forma de crear una sola clase que pudiera contener los atributos de ambas y por lo tanto la misma funcionalidad, pero en este caso, esta distribución se ha escogido de esta manera debido al siguiente argumento: un objeto zapatoIzquierdo o zapatoDerecho, no hace nada por si solo, es decir, un objeto zapatoSolo, ya sea izquierdo o derecho, no hace nada si la funcionalidad que hemos definido para éste es la de calzarlo, como se vio en la unidad anterior. Por lo tanto, la clase ParDeZapatos, que es la que realmente agrupa el conjunto del zapato izquierdo y el derecho, es la que debe controlar tanto la estructura y el comportamiento, pues es la que representa la funcionalidad de permitirle a usted calzarse. Entonces la clase que representa al objeto su zapatoIzquierdo y al objeto su zapatoDerecho, se denominará Zapato y tendrá un solo atributo que representará si es el zapato izquierdo o el derecho y que se denominará posicionZapato, como se muestra en la siguiente figura:



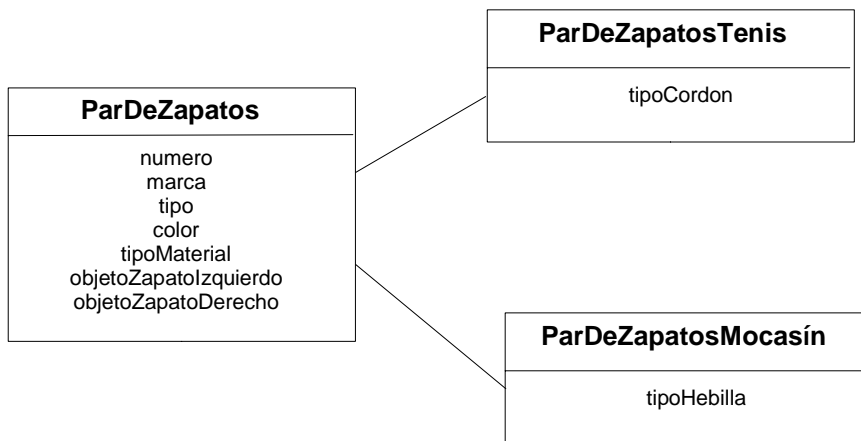
Luego la definición de los atributos de la clase ParDeZapatos queda de la siguiente manera:



En esta definición de la clase ParDeZapatos se está definiendo una clase compuesta, que agrupa las generalidades de las clases ParDeZapatosTenis y ParDeZapatosMocasín, es decir los atributos que son comunes a ambas clases. Ahora bien, se debe tener claro que, tanto a la clase ParDeZapatosTenis como a la clase ParDeZapatosMocasín les falta un atributo particular que no está contenido en la definición de la clase Par de Zapatos.

En la clase ParDeZapatos reuniremos los atributos que comparten tanto la Clase ParDeZapatosMocasín como la clase ParDeZapatosTenis y en éstas clases vamos a dejar los atributos que solo las caracterizan a ellas, o sea, los atributos tipoHebilla y tipoCordon respectivamente.

La representación gráfica de estas clases seria como ilustra la siguiente figura.

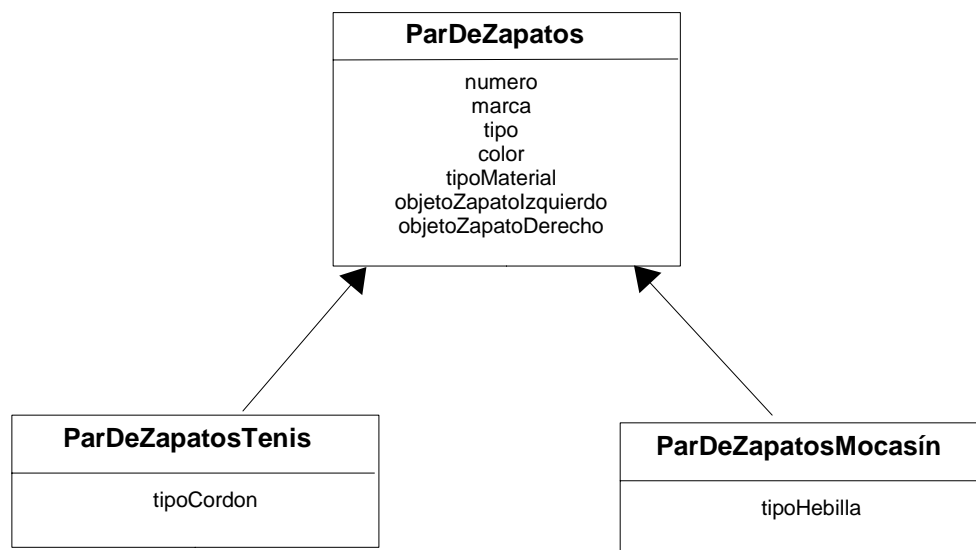


La razón por la cual se hace esto es porque, agrupando en la Clase Par de

Zapatos las características comunes, se está evitando tener dos clases que siendo similares tengan definiciones duplicadas de características. De alguna manera se debe definir que tanto la Clase ParDeZapatosMocasín y la Clase ParDeZapatosTenis están contenidas, agrupadas, pertenecen o son de la Clase ParDeZapatos. Este tipo de agrupación, en donde una clase general agrupa características que otras clases poseen, independiente de las características particulares que las clases a ser agrupadas posean además, se denomina **jerarquía de clases**.

El término ilustra claramente la idea, puesto que en el proceso de encontrar una clase con estructura y comportamiento similar que represente elementos que pueden ser agrupados en una clase más general que las contenga, justamente lo que se está haciendo es ir de lo particular a lo general, estableciendo así como un tipo de clasificación, en la cual se van jerarquizando las clases y esta organización puede ir de las más generales a las más particulares o viceversa.

La jerarquía se representa gráficamente de acuerdo a la siguiente figura. La flecha significa que las clases ParDeZapatosMocasín y ParDeZapatosTenis, toman o **heredan** de la clase ParDeZapato, todas las características de ésta, además tienen sus propias características tipoHebilla y tipoCordon respectivamente, las cuales los hacen diferentes de otro par de zapatos.



De esta manera, se está diciendo que la clase ParDeZapatosMocasín **es un** ParDeZapatos y que la clase ParDeZapatosTenis **es un** ParDeZapatos también,

Los objetos: Organización y relación entre Clases 2 - 27

que además de las características que heredan de la clase Zapato, tienen las suyas propias.

A la jerarquía que representa este ejemplo de la Clase ParDeZapatos, también se le denomina estructura de clases o jerarquía de generalización/especialización. Esta relación se da mediante la identificación <<es un>> entre clases.

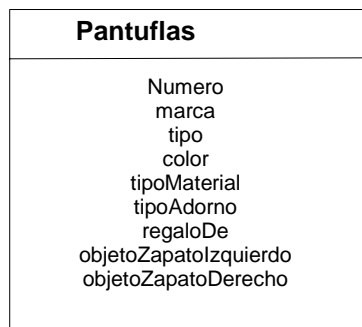
A la clase ParDeZapatos, que es la clase general en la cual están agrupadas la clase ParDeZapatosMocasín y la clase ParDeZapatosTenis, se le denomina **Superclase**, o clase Padre. A las clases particulares que están agrupadas dentro de una clase particular, en este caso la Clase ParDeZapatosTenis y la clase ParDeZapatosMocasín se les denomina **Subclases**, clase hijas o clases derivadas.

Pero, si se ha definido una clase ParDeZapatos que agrupa las características de los objetos que permiten calzarlo a usted.... ¿no cree que hemos olvidado a un objeto? ¿Dónde queda el objeto pantuflas regalo de su abuela? ¡El que ya no se sepa de que color son y que el perro de su casa las haya mordido, no nos da derecho a omitirlas en tal grado!. Independiente del estado actual en que el objeto sus pantuflasRegaloAbuela estén, ellas le han servido a usted para calzarse, luego son un ParDeZapatos, lo que pasa es que pertenecen a la clase Pantuflas, es decir a la clase que agrupa aquellos objetos que sirven para calzarlo a usted, pero solo en su casa.

Son un ParDeZapatos cómodos, que le sirven para andar en la casa, incluso para ir a la panadería de la esquina a comprar el pan para el desayuno, pero ¡para nada más!. Se definirá entonces la clase Pantuflas que agrupe a los objetos par de pantuflas en general. Los valores para los atributos del objeto **pantuflasRegaloAbuela**, quedarán como se ilustra a continuación:

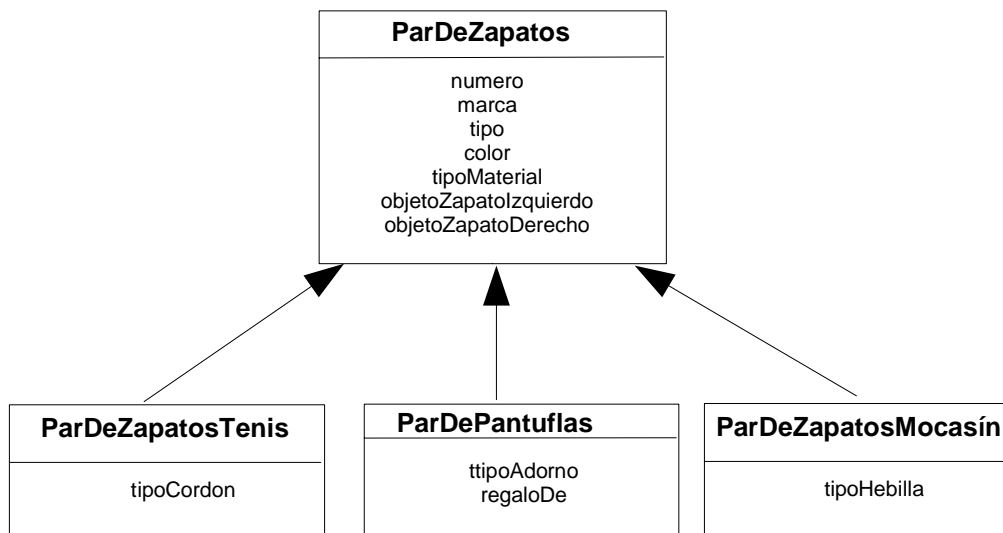
Atributos Clase Pantuflas en notación	Valores de atributos para el objeto pantuflasRegaloAbuela
numero	39
marca	Petite
tipo	Para la casa y no más!
color	Blanco y amarillo mostaza
tipoMaterial	peluche
tipoAdorno	Cabeza de Winnie de Pooh al frente
regaloDe	abuela
objetoZapatoIzquierdo	pantuflaIzquierda
objetoZapatoDerecho	pantuflaDerecha

En notación gráfica, la clase Pantuflas quedaría:



Como la clase Pantuflas tiene características similares que las clases ParDeZapatosMocasín y ParDeZapatosTenis, entonces la clase Pantuflas está agrupada también dentro de la clase ParDeZapatos, pues el objeto par de pantuflasRegaloAbuela **es un** tipo de ParDeZapatos.

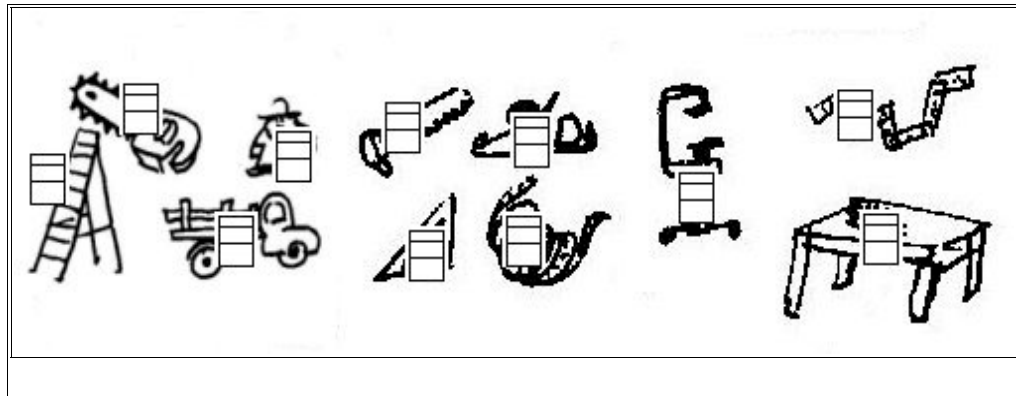
Luego, el gráfico con las jerarquías de clases alrededor de la clase ParDeZapatos, queda como se ilustra en la siguiente figura.



Ya usted sabe que existen objetos que están compuestos de otros objetos, o sea, objetos que hacen parte de otro más complejo u objetos que están agregados dentro de otros que lo componen y que a esto se le llama **Composición**. Usted ya sabe también que existen clases que pueden recopilar características de clases similares, en otras palabras, una clase general puede agrupar la estructura

y el comportamiento de clases similares, que esa clase general se llama **superclase** y que las clases agrupadas en ella se llaman **subclases** y que ésto se llama jerarquía de clases.

Retomando el problema planteado al principio de la unidad, acerca del bloque de madera a obtener a partir de un tronco de árbol, se puede representar cada uno de los objetos que se ha indicado se utilizan en la solución, por una clase, por ejemplo como se ilustra en la siguiente figura:



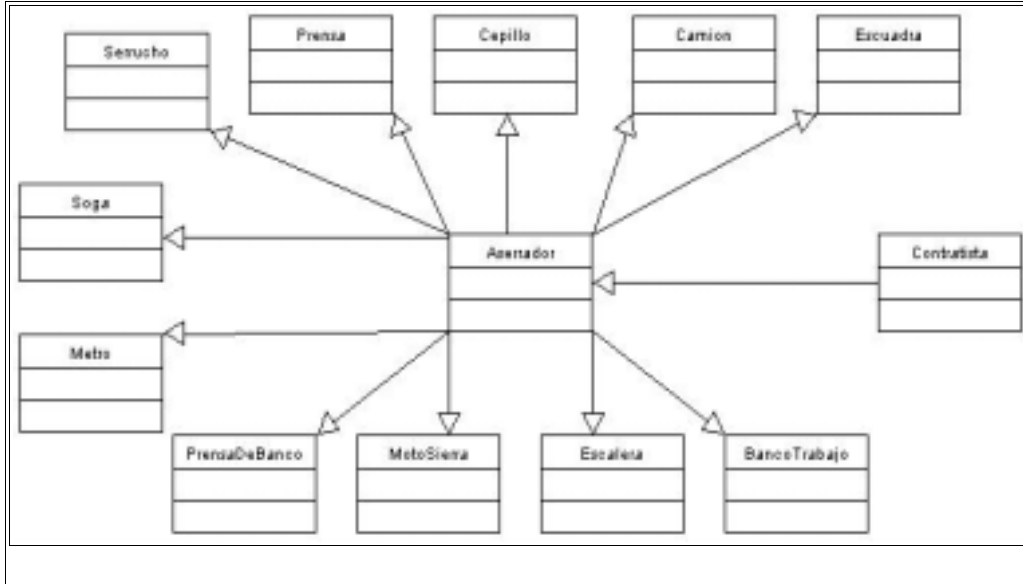
Lo cual organizado en conjunto se ilustra en la siguiente figura, donde solo se han tenido en cuenta las clases a las cuales pertenecen los objetos.

Serrucho	Aserrador	Cepillo	Camion
Soga	Prensa	Escuadra	Banco Trabajo
Metro	Prensa	MotoSierra	Escalera

Como se puede observar, en esta figura están indicadas algunas de las clases que se utilizan en la solución del problema, sin embargo ni sus atributos ni sus métodos se han indicado. El lector debe llenar estos campos con datos correspondientes a objetos de la vida real.

En la siguiente figura se ilustran las relaciones entre las diferentes clases, respecto del aserrador, puesto que este, o sea la clase Aserrador, es la clase

encargada de usar los diferentes objetos. Las relaciones entre las clases y la clase Aserrador es de uso.



Mensajes entre clases

Ahora por primera vez usted va a aparecer en escena, como objeto, claro está. Imagínese a usted dentro de su habitación, recostado en su cama viendo en la televisión su programa favorito. Es de noche y mañana usted tiene que madrugar antes de lo acostumbrado, luego tiene que reprogramar el reloj despertador. Usted no interrumpe su programa favorito solo espera que, estén en propaganda para tomar el despertador.

Por favor, para efectos prácticos de este ejemplo se le sugiere que las siguientes secuencias de imágenes asociadas con los pasos a seguir las pase en cámara lenta. Usted toma el reloj despertador, activa el botón Alarma, luego ajusta el control para programar la alarma en la hora deseada, en este caso las 5 a.m., así el despertador le queda cuadrado en las 5:00 a.m. Luego se asegura nuevamente que el despertador quede con “alarma programada”. Para asegurarse que su reloj no se pare antes de la hora indicada para despertarlo, ya que es un despertador de cuerda, usted le da cuerda girando la manecilla que se encuentra en la parte posterior.

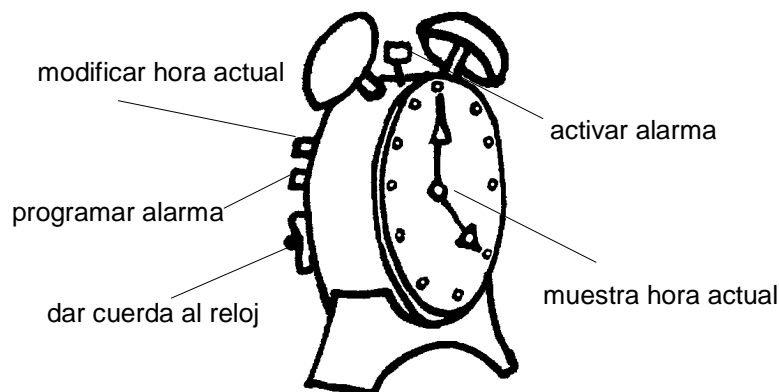
Como se ha traído a la escena al objeto usted y al objeto relojDespertador entonces se definirá las clases para ambos objetos como se ha hecho en las secciones anteriores.

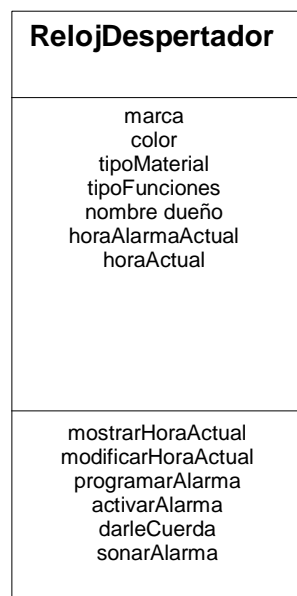
Atributos para la clase RelojDespertador y los valores de dichos atributos para su objeto relojDespertador se pueden establecer de acuerdo a la siguiente figura:

Atributos de la Clase RelojDespertador	Valores de atributos para el objeto relojDespertador
marca	Pony
color	Azul Cielo
tipoMaterial	Metal-Plástico
tipoFunciones	Alarma
nombreDueño	Usted
horaActual	La hora actual
horaAlarmaActual	5:00 a.m.
operacion	De cuerda

Con el atributo tipoFunciones lo que se quiere indicar es que hay algunos relojes despertadores que no solo tienen alarma para despertar sino que también son a la vez, radios, por ejemplo. En este caso particular su despertador de cuerda, es solo eso: un reloj despertador, por eso en el valor del atributo tipoFunciones se indicó únicamente la “alarma”. Si hubiese sido también radio, hubiésemos puesto alarma y radio, por ejemplo.

Con lo anterior se ha definido la estructura de la clase RelojDespertador, definamos entonces el comportamiento de dicha clase. Primero se identificarán las funciones que realiza un objeto relojDespertador. Un objeto Reloj Despertador da la hora, modifica su hora actual, programa la alarma, activa la alarma, hace sonar la alarma, detiene el sonido alarma, recibir cuerda, entre otras. Centrémonos en estas funciones, que son las más relevantes, las cuales se muestran en las siguientes figuras:





Ahora le toca su turno. ¿Qué creyó, que lo habíamos olvidado? Pues ni más faltaba. Usted, el dueño de los objetos que hay dentro de la habitación (o por lo menos eso es lo que aparenta), el propietario de la habitación y es también un objeto!. Tranquilo, no lo estamos ofendiendo, está bien que de vez en cuando se sienta el “objeto” de su novia, de sus amigos o amigas, pero no nos estamos refiriendo a lo que usted en determinado momento pueda sentir, sino a lo que usted es desde el punto de vista del enfoque de objetos: también es un objeto.

Observemos, ¿usted tiene estructura? ¿Usted tiene comportamiento? ¡Claro que sí! ¿Usted tiene similares y por tanto puede estar agrupado en una clase? ¡Claro que también!

Entonces se definirá una Clase Persona dentro de la cual está agrupado usted y

todas las demás personas. Identifiquemos cuáles podrían ser algunos de los atributos que podemos mencionar de la clase Persona, a título de ilustración para nuestro ejemplo: nombres, apellidos, numero de identificación, ocupación, programas de televisión favoritos, hobbies, edad, fecha de nacimiento, estado civil, hora promedio de levantarse, tipo de sangre, estatura, sexo, hora promedio de acostarse, color de cabello, color de ojos, y podríamos seguir con una lista mayor, pero por ahora solo nos ocuparemos de algunos de los atributos anteriores. Debe anotarse que, todos los anteriores atributos, tomando valores en usted, lo caracterizan como un objeto único, ¿estamos de acuerdo?.

Ahora examinemos o tratemos de identificar, a título de ilustración para nuestro ejemplo, algunas funciones que realiza usted como objeto de la clase Persona.

Usted se levanta temprano, se baña, desayuna, se viste, se sube a un bus o a un carro, asiste a clase, presenta exámenes, almuerza, va al baño a hacer sus necesidades diarias, conversa con sus compañeros, se deleita viendo chicas en la universidad, de nuevo se sube a un bus o a un carro, grita al conductor del bus por frenar despiadadamente o grita a la persona que conduce el carro de al lado por no saber manejar, entra a su casa, saluda a su familia, estudia o repasa lo que vio en el día de hoy en sus clases, cena con su familia, se acuesta en su cama, prende el televisor, cambia mil veces de canal hasta que encuentra qué ver mientras empieza su programa favorito, sintoniza su programa favorito, se ríe lo más duro que puede, patatea encima de la cama, se pone la almohada encima, habla solo y en voz alta, vuelve a cambiar mil veces de canal una vez su programa favorito se ha acabado, re-programa el reloj despertador porque mañana tiene un parcial a las 7:00 a.m., apaga el televisor, apaga la luz, se cepilla los dientes, le da las buenas noches a su familia, se acuesta, se duerme, ronca y hasta habla dormido.

Persona
NumeroDeIdentificación fechaDeNacimiento nombre
programasDeTVFavorito s hobbies horaDeLevantarse horaDeAcostarse colorCabello
colorOjos levantarse bañarse asistirAClase almorzar conversar gritar cenar encenderTV cambiarCanalTelevisor sintonizarCanalTV apagarTV encenderGrabadora apagarGrabadora programarRelojDespertador apagarTV apagarLuz cepillarDientes darBuenasNoches acostarse dormir



Persona
programandoRelojDespertador

Este es un corto resumen de cuáles pueden ser las funciones que usted realiza en un día cualquiera y en general, así que para el ejercicio que hemos venido haciendo se tomarán las más significativas dentro del ejemplo y por favor, no reniegue si omitimos alguna.

La Clase Persona, en la cual está agrupado el objeto Usted, se define de la manera indicada en la siguiente figura. Para efectos prácticos, los atributos y métodos de la clase Persona, presentes en esta figura, son los que se tendrán en cuenta en esta unidad.

Hasta ahora no se ha mencionado que los métodos que conforman el comportamiento de una Clase, pueden clasificarse en dos grandes grupos:

1. Métodos que implementan las funciones propias de la clase
2. Métodos que implementan funciones que dependen de sucesos o eventos generados por otros objetos.

Por ejemplo, en el caso de la clase Persona, métodos que implementen funciones propias de la clase son levantarse, bañarse, almorzar, cenar, acostarse, dormir, encenderGrabadora, sintonizarCanalTV, entre otros, puesto que son actividades propias de la Persona y que se dan como una rutina en su vida, es decir, las realiza independiente del día y del lugar.

Los métodos que implementan funciones que dependen de sucesos o eventos generados por otros objetos, como conversar, gritar, reír, etc. Por ejemplo conversar, depende de sucesos o eventos generados por otros objetos, ya que si usted va a la cafetería después de salir de clase y no se encuentra con ninguno de sus amigos en ella, entonces no puede conversar. Para ello necesita encontrarse con alguien de sus conocidos o simplemente contestar gustoso el coqueteo de alguna chica que está esperando que le pregunte algo para iniciar conversación con usted.

En el caso de gritar, si el conductor del bus al que se montó no es tan salvaje y es más recatado para manejar, o el conductor que va en el carro de enseguida fuera más consciente al manejar, entonces usted no tendría que gritar. Gritó por un estímulo externo, porque algo le molestó y quiso que quién produjo su incomodidad se diera cuenta de su disgusto, lo cual fue una reacción a un suceso o evento generado por el conductor del bus o por el conductor del carro de enseguida.

En el caso de reír, si la escena que están pasando en su programa favorito no es de su total agrado o no le causa chiste en el momento, usted no se reiría, pero si la escena le recuerda un hecho gracioso en su vida (porque hay un dicho que afirma: “el que se ríe solo de sus picardías se acuerda”) entonces así sea solo, usted se carcajea hasta que no pueda más. Luego tuvo que observar una escena que le pareció graciosa en su programa favorito y que de inmediato hizo que su cuerpo reaccionara para producir risa.

Entonces hay métodos que son propios de la clase y se realizan porque hacen parte de su rutina y otros métodos que solo se realizan dependiendo de lo que pase con otros objetos y cómo afecten éstos a la clase.

Usted como objeto que pertenece a la clase Persona y como dueño de la habitación y de los objetos que hay dentro de ella, es quien de alguna manera los utiliza y los organiza, es decir, es quien en primer lugar, se relaciona con ellos, entendiendo por “relaciona”, en este caso, **utilizar**.

Usted como objeto se relaciona con los objetos parDeZapatosMocasínNegros, ya que con ellos usted se calza, usted se relaciona con su objeto lámparaDemesaDeNoche del lado izquierdo, porque usted es quién la enciende y la apaga. Lo mismo pasa con su grabadora y con su televisor. Usted se relaciona con el objeto relojDespertador porque usted es quien lo programa para activar la alarma. Con su cama usted se relaciona porque en ella es donde usted duerme y así podemos seguir mencionando las relaciones que usted tiene con cada uno de los objetos de la habitación. Por eso, teniendo en cuenta su relación en el caso de la grabadora y el televisor, incluimos los métodos encenderTV y encenderGrabadora y apagarTV y apagarGrabadora, por ejemplo, en la definición del comportamiento de la clase Persona. Ya que Usted es el encargado de hacerlo.

Cuando usted enciende el objeto televisor, usted debe cojer su objeto controlRemotoDeTv y presionar el botón de encendido (Power). Como se va a trabajar ahora con el objeto controlRemotoDeTv entonces se definirá en seguida tanto los atributos, como los métodos de la clase ControlRemoto.

Atributos Clase ControlRemoto	Valores de atributos para el objeto Control RemotoDeTv
marca	Pony
conjuntoPilas	Pila 1, Pila 2

Los objetos: Mensajes entre clases 2 - 36

Atributos Clase ControlRemoto	Valores de atributos para el objeto Control RemotoDeTv
conjuntoBotones	Power, Muting, Sep, Display, TV/Video, 1, 2, 3, 4, 5, 6, 7, 8, 9, SubirCanal, BajarCanal, subirVolumen, BajarVolumen
...entre otros...	

ControlRemoto
marca conjuntoPilas conjuntoBotones
EnviarSeñalEncenderTelevisor enviarSeñalApagarTV enviarSeñalCambiarCanalTV subirVolumen bajarVolumen activarMuting activarVideo activarModoSleep

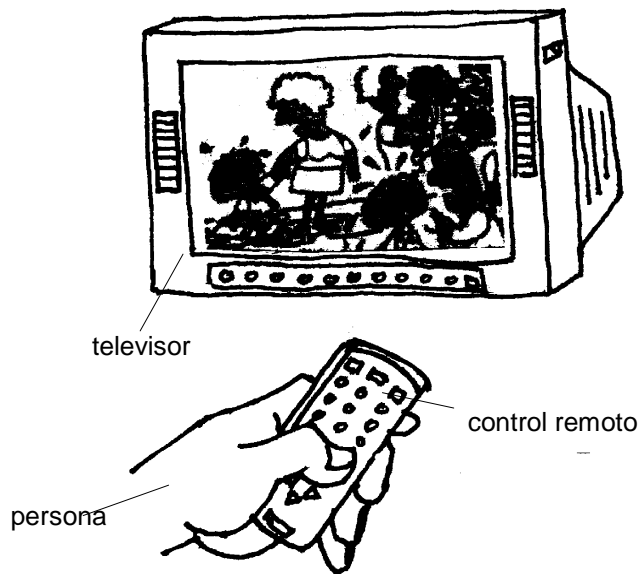
Por ahora solo nos van a interesar los métodos que hemos identificado en el gráfico anterior.

Para nuestro ejemplo, suponemos que usted iba a presionar el botón Power de su controlRemoto. En ese instante usted se está relacionando con el objeto controlRemoto, el que a su vez se está relacionando con el objeto televisor. En otras palabras, usted está dando una orden al objeto controlRemoto para que encienda el objeto televisor. El objeto controlRemoto a su vez, está enviando una orden al televisor para que se encienda y aparezca la imagen en la pantalla.

Hay una relación entre más de dos objetos que en este caso podríamos decir que es una orden. Esta orden o esta relación, lo que hace desde el momento en que usted presiona el botón Power es, activar el método encenderTV del objeto controlRemoto, método que entonces activa el método desplegarImagen o encenderTV del objeto Televisor.

Cuando usted va a cambiar el canal actual que se está visualizando en el televisor, entonces oprime el botón correspondiente al canal, por lo tanto usted está activando su televisor mediante el objeto controlRemoto. Vemos que este

método requiere de un suceso o evento de otro objeto para activarse, luego podemos clasificarlo también como del tipo 2 o de los que requieren de un suceso o evento generado por otro objeto, tal como se explicó antes.



Con esto se puede entender que, existen objetos que se relacionan con otros por medio de órdenes y que esas órdenes se dan a través de mensajes. Esos mensajes entonces activan métodos que requieren de ellos para comenzar a realizar la operación que representan y que esos mensajes activan métodos que se pueden clasificar dentro del tipo de aquellos métodos que requieren de un suceso o evento producido por otro objeto para poder activarse.

Ahora imagine que usted estaba viendo su programa favorito y que se ha terminado. Usted mira su objeto reloj despertador y son las 11:00 p.m., y recuerda angustiosamente que mañana tiene un parcial a las 7:00 a.m. Entonces se pone su pijama, apaga el objeto televisor a través del botón Power de su objeto controlRemoto, apaga su objeto lámparaDeMesaDeNoche, se acuesta en su objeto cama, se arroja con su objeto cobija porque está haciendo mucho frío y activa su método dormir y muy posiblemente active su método soñar. Que tenga un feliz descanso, que amanezca como nuevo y que le vaya muy bien en su parcial. Mañana será un nuevo día, ¡mucho suerte!

La Programación Orientada a Objetos

La Programación Orientada a Objetos (POO) es un paradigma de la programación de computadores. Cuando se habla de paradigma de programación se hace referencia al conjunto de teorías, estándares, modelos y métodos que permiten organizar el conocimiento, proporcionando un medio bien definido para visualizar el dominio e implementar en un lenguaje de programación la solución de un tipo de problema.

El paradigma orientado a objetos se basa en el **modelo objeto**, que tiene como objeto de conocimiento o elemento principal el objeto, como se vió de manera informal en la identificación, descripción y caracterización de algunos de los objetos de su habitación.

Como ya se ha establecido a partir de los ejemplos dados, un objeto es una entidad que contiene en sí misma los datos (valores de sus atributos) que la identifican y los algoritmos (métodos) que permiten manipular dichos datos.

Los diferentes tipos de paradigmas de programación, requieren de una manera propia de pensar y enfocar el problema a resolver, pues cada uno de ellos tiene un marco de referencia conceptual que permite modelar el dominio del problema.

Anterior al paradigma de objetos, es el paradigma algoritmico o de procesos, el cual se fundamenta en los procesos o funciones que se llevan a cabo en el mundo real dentro del dominio de un problema dado. Concretamente se refiere a lo que le entra, qué y cómo maneja dichas entradas y qué sale o produce un proceso determinado.

Para la programación tradicional se puede decir que el marco de referencia conceptual lo sustentan los procesos, algoritmos o bloques de construcción modular, cuya abstracción va de lo general a lo particular, mientras que la POO tiene como marco de referencia conceptual el objeto, el cual pertenece a una clase que agrupa a todos aquellos compañeros suyos que presentan las mismas características y comportamiento similar.

La manera tradicional de pensar un programa es verlo como una lista de instrucciones que le indican a un computador qué hacer. Bajo el enfoque de objetos un programa es un objeto, que a su vez es un conjunto de objetos que operan juntos en formas predefinidas para realizar las tareas que les han sido asignadas. Mediante el enfoque de objetos, su programa está compuesto de partes diferentes llamados objetos, y en el enfoque algoritmico el programa esta compuesto de funciones, procedimientos, subprogramas.

En contraste con la programación tradicional, la programación orientada a objetos utiliza objetos como sus bloques lógicos de construcción fundamentales, y no procesos. Cada objeto es una instancia de una clase y las clases están relacionadas con otras clases por medio de relaciones de herencia.

En la programación orientada a objetos, la solución a un problema se construye sobre la base de clases del dominio de este, sus atributos, métodos y las relaciones entre estas.

Para un lenguaje el soportar herencia significa que es posible expresar relaciones <<es un>> entre clases. Si un lenguaje no ofrece soporte directo para la herencia entonces no es un lenguaje completamente orientado a objetos.

Una de las ventajas que presenta el paradigma de objetos con respecto al algorítmico, es la facilidad que brinda, a través de sus herramientas, de concebir, analizar, modelar, diseñar e implementar el mundo real de manera fiel a como se presenta en la realidad. En otras palabras, el paso que hay desde la concepción y asimilación del problema hasta la implementación del mismo, es un proceso que se hace de manera casi natural.

Al facilitar la representación del dominio del problema de una manera fiel a la realidad, uno de los mayores logros de este paradigma lo constituye la facilidad con que se puede traducir el modelo de objetos, plasmado en papel, al código en un lenguaje de programación orientado a objetos. Es decir se simplifica la etapa final del desarrollo de programas, o sea la programación en sí.

En ese sentido, las técnicas orientadas a objetos, por su naturaleza misma proporcionan mejoras y metodologías adecuadas para construir sistemas de software complejos, a partir de partes de código llamadas **componentes** lo cual hace que el software sea modularizado y reutilizable.

El concepto de **reutilización**, desarrollado bajo el enfoque de objetos, le permite al programación utilizar código que ya se ha implementado y probado, agregándole en caso de requerirse algunas características, o modificándole o sobrescribiendo métodos sin necesidad de desde cero. Se facilita y se mejora considerablemente la labor de programación, haciéndola más eficiente y productiva.

Propiedades de la Programación Orientada a Objetos

El paradigma orientado a objetos, posee unas características que lo definen y lo diferencian de los demás paradigmas. Esas características están definidas por las propiedades más importantes que posee, las cuales son:

Abstracción

Es la propiedad que permite distinguir a un objeto de los demás, observando en el objeto sus características y su comportamiento, pensando más bien en qué es el objeto y no, cómo es traducido ese objeto en código a la hora de programarlo.

El ejercicio hecho en la sección anterior, en cuanto a encontrar objetos en su habitación y distinguir de los objetos que hay en la habitación, cuales eran relevantes para ese ejemplo y cómo estarían esos objetos agrupados en clases según sus características similares, es lo que se denomina Abstracción.

La abstracción consiste en la identificación de objetos dentro del dominio del problema que se está estudiando, identificar cómo esos objetos están agrupados en clases según las similitudes que ellos tengan, en cuanto a la estructura y el comportamiento que posean, así como de las relaciones entre ellos.

Encapsulamiento

Es la propiedad que permite incluir en un mismo espacio los elementos identificables en un objeto, los que constituyen su **información** (los datos o atributos) y las **operaciones** (los métodos o funciones) que operan sobre esa información.

El encapsulamiento es la propiedad que permite agrupar en una clase la estructura y el comportamiento de los objetos que están representados por ella. En una clase se hallan inmersos los datos y los algoritmos que manipulan los datos de los objetos agrupados en ella, es decir, dentro de la clase está todo lo que requieren los objetos para existir como tales.

En el caso de su habitación, fue justamente definirles a las clases que se habían identificado, los atributos que las caracterizan, así como los métodos que contienen y que van a manipular esos atributos. En el caso de la programación, el encapsulamiento es lo que permite que tanto la estructura como el comportamiento se encuentren dentro del mismo cuerpo de código de una clase determinada.

Modularidad

Es la propiedad que caracteriza a las aplicaciones de software, que se pueden definir como la agregación de partes más pequeñas (componentes o módulos), donde cada una de esas partes es lo más independiente posible de las otras y de la aplicación en su conjunto.

La modularidad en el código de una aplicación, consiste en que ésta tenga módulos de código independientes entre sí, que todos en conjunto hacen que la

aplicación funcione correctamente. En el caso en que se requiera actualizar un módulo o un componente, es decir una parte del programa, entonces solo se requerirá modificar el módulo respectivo y no hay necesidad de hacer cambios en otras partes del código de la aplicación. Esto facilita en alto grado, el proceso de mantenimiento de una aplicación o programa.

Con respecto a los objetos identificados en su habitación, cada clase en sí, con su estructura y comportamiento correspondiente representa un módulo, dentro del problema y un módulo también para el caso del software, una vez esté implementado.

Jerarquización

Es la propiedad que permite organizar en un problema determinado, las clases que se están identificando dentro de una jerarquía existente, ó dentro de una nueva. Cada clase una vez implementada llega a ser parte de una jerarquía determinada.

Se presenta la jerarquización cuando se define una superclase y a partir de ésta se definen clases que vienen siendo sus subclases y a éstas se le definen subclases también y así sucesivamente, hasta llegar a clases particulares o específicas, que contienen tanto atributos como metodos propios para una generalidad de objetos.

En el caso de los objetos de su habitación, se presenta la jerarquización cuando se creó una superclase ParDeZapatos, de donde ParDeZapatosMocasín, ParDeZapatosTenis y ParDePantuflasRegaloDeAbuela son subclases, lo que se está haciendo es ir organizando las clases dentro de una jerarquía que encabeza la clase ParDeZapatos. En estos casos, las subclases ParDeZapatosMocasín, ParDeZapatosTenis y ParDePantuflasRegaloDeAbuela se derivan de la clase ParDeZapatos y al ser derivadas, ellas heredan la estructura y el comportamiento de su superclase ParDeZapatos.

Para el objeto su Grabadora, se observa que se compone de objetos como los diferentes botones, la antena, el cable, las caseteras, etc., y se dijo que se establecía una relación de composición, allí también se está aplicando la propiedad de jerarquización, puesto que la composición permite organizar y determinar una jerarquía de objetos que la encabeza el objeto compuesto.

Teniendo en cuenta el caso de la jerarquía definida por la clase ParDeZapatos por un lado, y por el otro la Grabadora, se pueden distinguir dos clases de jerarquías:

- Jerarquía de clases o de generalización/especialización.

Determinada por una relación **es-un (is-a)**, entre clases. Se le denomina también generalización/especialización, porque cuando se crea una subclase de una clase, se está especializando la superclase por medio de su clase derivada, es decir, se está agregando más estructura o más comportamiento que el que posee la superclase. En otras palabras, se está realizando un proceso de especialización de características. De la misma manera, si se observa el sentido contrario, es decir, partir de las subclases para encontrar la superclase, lo que se está haciendo es encontrar una clase más general que las cubra a todas. Se está llevando a cabo un proceso de generalización de características. Tal es el caso de la clase ParDeZapatos que fue generalizada a partir de las clase ParDeZapatosTenis y ParDeZapatosMocasín.

Por ejemplo, la clase Avioneta **es un** tipo de la clase Aeronaves, una clase Manzana **es un** tipo de la clase Frutas, la clase Par de Zapatos Tenis **es un** tipo de la clase Par de Zapatos, un BlueJean **es un** Pantalón, un Gato **es un** Mamífero, una Rosa **es una** Flor, etc. Donde Aeronaves, Frutas, Par de Zapatos, Pantalón, Mamífero, Flor son las superclases a partir de las cuales se pueden derivar las subclases indicadas.

El perrito de nombre Lucas, que tiene su vecina, es de la raza Pincher Miniatura, que a su vez pertenece a la clase de Canes, que son Carnívoros, que son Mamíferos, que son Vertebrados, que son SeresVivos. Es decir que el perrito Lucas, hereda todas las características de los SeresVivos, Vertebrados, Mamíferos, Carnívoros, Canes y PincherMiniaturas.

La definición del objeto perritoLucas, permitira que se “hereden” todas las características y metodos de todos sus ancestros o superclases en la jerarquía de clases anotada. Sin embargo se le añadirán aquellos atributos o métodos propios del objeto perritoLucas. Por ejemplo el perritoLucas tiene un collar de dientes de tiburón, lo cual es un atributo propio de ese perritoLucas y de ninguno otro. Así mismo tiene un método propio, que solo le pertenece a el y es el de esconderse dentro del microondas. Tambien es un método propio del peritoLucas y posiblemente de ningún otro perrito.

A este tipo de jararquía también se le denomina **Herencia**, ya que al pertenecer las clases a una jerarquía determinada, en la cual una clase posee una superclase o clase padre, de la cual toma estructura y comportamiento, es como si de alguna manera estuviera “heredando” dichas características.

- La jerarquía de objetos o de agregación:

Determinada por una relación **parte de (part of)** entre objetos y también llamada relación de agregación o relación de composición. Se denomina también jerarquía de agregación porque cuando se define que un objeto está compuesto de otros, es como si se le agregaran objetos al objeto compuesto. Cada una de las partes que se agregan se denominan agregados.

Este tipo de jerarquía se define en la declaración de atributos de la clase a la cual pertenece el objeto compuesto. Cada uno de los objetos agregados del objeto compuesto deben aparecer como atributos en la definición de la clase de éste. Por eso se denomina también jerarquía de objetos, porque esta relación solo se hace clara cuando se tiene un objeto y se pueden ver sus componentes, mientras que en la relación de generalización/especialización no hay necesidad de tener un objeto para ver claramente la relación.

Por ejemplo, el objeto antena es **parte de** su objeto grabadora, el objeto botónPlayCasete es **parte de** su objeto 'Grabadora, el lectorCD es **parte de** su objeto grabadora, el objeto boton CambioFuncion es **parte de** su objeto grabadora y así sucesivamente con todos los objetos agregados del objeto su grabadora. El objeto collarDeDientesDeTiburon, es parte de su objeto perritoLucas.

Para el caso del objeto computador personal, el objeto teclado es **parte de** su objeto computador, el objeto monitor es **parte de** su objeto computador, el objeto CPU es **parte de** su objeto computador y así sucesivamente con todos los objetos agregados a su objeto computador.

Polimorfismo

Es la propiedad que indica que un elemento puede tomar diferentes formas. Para el caso de la programación orientada a objetos, el elemento puede ser un atributo, un método o un operador. Esta propiedad permite hacer referencia bajo la misma forma a elementos de diferentes clases y a operaciones que, teniendo el mismo nombre pueden realizarse de diferentes formas dependiendo de la clase que esté realizando la operación.

Desde el punto de vista de los métodos, permite que objetos diferentes puedan responder de modo diferente al mismo mensaje que activa un método, produciendo acciones o resultados totalmente diferentes. Desde el punto de vista de los **operadores**, se puede observar cómo un operador representa o significa más de una acción.

Cuando se habla del polimorfismo de métodos y operadores, se está hablando de una clase de polimorfismo que se llama **sobrecarga**, específicamente se habla de **sobrecarga de métodos**, que se define como el uso de varios métodos con el mismo nombre y cuyo contenido es diferente, así como al uso de un operador que dependiendo del tipo de datos sobre el que actúa realiza una operación diferente.

Cuando se habla de un operador o un método que con el mismo nombre puede utilizarse para que actúe de manera diferente dependiendo del tipo de datos que está utilizando, se habla de un tipo de polimorfismo que se denomina

Sobrecarga. Por ejemplo, en Java un caso muy común es el del operador ``+``. Usted puede utilizar indistintamente este operador para sumar números y para concatenar cadenas, ya que el operador está sobrecargado de tal manera que cuando usted lo utiliza Java sabe qué operación (suma o concatenación respectivamente) debe realizar dependiendo del tipo de los argumentos a partir de los cuales se está haciendo la invocación. Se pueden declarar métodos con el mismo nombre pero con invocaciones distintas que difieren por sus firmas, las que consisten en número y tipos de sus argumentos.

Las siguientes son las diferencias que pueden presentar métodos con el mismo nombre:

- La cantidad de argumentos que toman
- El tipo de datos u objetos de cada argumento

Estos dos aspectos conforman lo que se llama firma o signature del método. El uso de varios métodos diferentes con el mismo nombre se le denomina **sobrecarga de métodos**.

La sobrecarga de métodos elimina la necesidad de métodos totalmente distintos que en esencia hacen lo mismo. También hace posible que los métodos se comporten de manera distinta basados en los argumentos que reciben.

Para crear un método sobrecargado se crean distintas definiciones del método en una clase, cada una con el mismo nombre pero con diferentes listas de argumentos. La diferencia puede ser el número, el tipo de argumentos o ambos.

Persistencia

Es la propiedad que permite a un objeto sobrevivir a la ejecución del programa que lo crea y lo manipula. Tiene que ver con el almacenamiento y captura de los objetos que manipula un programa determinado, es decir, el objeto sigue existiendo así el programa haya terminado su ejecución.

Todas las anteriores propiedades de la Programación Orientada a Objetos, se tratarán en detalle en las siguientes unidades, dentro del contexto del lenguaje de Programación Java.

Algunas definiciones del enfoque de objetos

Clase

Es la descripción de un conjunto de objetos, que poseen las mismas características. Cuando se habla de características se habla de estructura (datos) y comportamiento (métodos u operaciones).

Como ya se indicó antes para el caso de las clases que agrupan objetos de su habitación, particularmente aquellas que tienen métodos, como en el caso de la grabadora, o el control remoto de su televisor, una clase posee nombre, estructura y comportamiento.

En una clase están agrupados los objetos que poseen los mismos datos y se comportan de manera igual. Dicho de otro modo, una clase es una declaración de un tipo de datos, en este caso, un tipo de objeto. Al ser tratada una clase como un tipo de datos, una clase viene a ser como una plantilla a través de la cual se pueden construir objetos de cierto tipo o de ciertas características. Es como si una clase se utilizara como un modelo por medio del cual se crean objetos cada vez que se requieran dentro del programa, no se necesitaría entonces estar repitiendo las mismas partes de código cada vez que se van a crear objetos similares, simplemente se define una clase con las características generales de los objetos que se van a crear y se crean objetos a partir de ella. Es más, si un objeto posee las mismas características pero además unas adicionales, se puede extender la clase, es decir, se puede crear una clase nueva que sea subclase de dicha clase y añadir la estructura y/o el comportamiento deseado, ahorrando también así, el tener que repetir código que ya está predefinido en una clase existente. Esta característica de heredar de una superclase es lo que se ha definido anteriormente como herencia.

Cada vez que se utiliza la clase como plantilla o modelo para crear un objeto, se está creando una instancia. En ese sentido los términos instancia y objeto se refieren a lo mismo.

Si se dice que una clase es como la declaración de un tipo de datos, un objeto vendría a ser una “variable” de ese tipo. En este punto radica la diferencia entre la definición de clase y objeto, la clase es el tipo de datos, mientras que el objeto es la variable real, la instancia de la clase. Esta diferencia es similar a la que existe en otros paradigmas de programación con respecto a tipo de dato y variable.

De esta manera se puede entender que, una clase, es una especie de plantilla o patrón que se utiliza para crear múltiples objetos con características similares.

Atributo

Es una característica de un objeto, que ayuda a definir la estructura de la clase que representa al objeto y que lo hace diferenciar de otros objetos. Un atributo se representa en la notación de la programación orientada a objetos de la siguiente manera:

tipo de dato nombreAtributo;

tipo de dato hace referencia los tipos posibles que puede tomar un atributo, por ejemplo entero, carácter, lógico, de punto flotante, etc. Estos tipos se definen y describen en la siguiente unidad.

nombreAtributo es el identificador del atributo, el cual se expresa con la palabra o las palabras que lo identifican. La primera letra de la primera palabra en minúscula y las primeras letras de las palabras siguientes en mayúscula, las demas en minúscula.

Un atributo toma en un objeto, un valor del rango que le ofrece el tipo de dato de su definición. En la clase se definen el conjunto de atributos que constituyen la estructura de la clase y esos mismos atributos toman valores cuando se crea un objeto a partir de esa clase, por ejemplo para los objetos CD Fabulosos Cadillacs y CD Aterciopelados, estos objetos toman los valores de los atributos que los diferencian el uno del otro.

Las instancias botónImprimir y botónGuardar de la clase boton y que pueden estar siendo usadas en un paquete de software cualquiera, toman valores para los atributos de dichas clases que los identifican de manera única: la etiqueta del primero es “Imprimir”, mientras la etiqueta del segundo es “Guardar”. La etiqueta hace referencia al letrero que acompaña el respectivo boton y que es un atributo de esta clase.

El conjunto de valores que toman los atributos para un objeto en un momento dado es lo que se denomina estado del objeto.

Método

Es la implementación de un algoritmo o proceso que representa una operación o función que una clase realiza. El conjunto de métodos que representan las funciones que una clase realiza determinan el comportamiento de dicha clase. Al definir cuáles son los métodos que una clase realiza, es necesario tener en cuenta cuáles son las **responsabilidades** que los objetos de dicha clase deben tener, o sea qué funciones deben realizar.

Un método ofrece entonces, el comportamiento que un objeto presenta cuando le es enviada una señal o un mensaje, al cual debe reaccionar. Los métodos se pueden clasificar en dos categorías:

- Métodos que la clase realiza por sí misma y que son inmanentes a su definición. Dentro de este grupo se pueden tener a su vez dos tipos de operaciones:

Operaciones que manipulan los datos del objeto de alguna forma específica, por

ejemplo asignan un valor, cambian un valor, etc.

Operaciones que realizan un cálculo o proceso específico.

- Métodos para que los objetos puedan reaccionar ante eventos del medio externo. Específicamente activados por medio de mensajes originados por otros objetos.

Un método se representa en la notación de la programación orientada a objetos de la siguiente manera:

tipo nombreMetodo (lista argumentos) ;

tipo hace referencia al tipo de datos del valor que retorna, en caso de que retorne algún valor. Si no retorna valor alguno se escribe **void**.

nombreMetodo es el identificador que representa el nombre dado al método. La primera letra de la primera palabra en minúscula, y las primera letras de las palabras siguientes en mayúscula, las demás en minúscula.

lista argumentos, representa los identificadores de los elementos que entran al método en la definición de este. Los argumentos se separan con comas, y para cada argumento se debe especificar su tipo y su identificador. Se denominan parámetros, cuando el método se invoca, correspondiendo a los valores verdaderos o actuales a tener en cuenta.

Mensajes

Estímulo que recibe un objeto, a fin de activarlo y generar una reacción. El objeto que envía el mensaje se llama **objeto fuente** u objeto emisor y el objeto que recibe el mensaje se denomina receptor u **objeto destino**. Un mensaje es un tipo de operación en la cual un objeto actúa sobre otro, por lo que entonces un mensaje es el resultado en un objeto, de solicitar la ejecución de uno de sus métodos, de aplicar sobre el de un método del objeto emisor. Es recibir de otro objeto una señal a la cual deba responder con una acción determinada.

Cuando Usted presiona el botón Power de su control remoto, el objeto Usted concretamente esta enviando un mensaje que consiste en la petición de invocación del método EnviarSeñalPrender/Apagar del objeto su Control Remoto. Así mismo el objeto su Control Remoto apenas recibe la señal del objeto Usted, enviará un mensaje al objeto Televisor que consiste en la petición de invocación del método Prender/Apagar del objeto Televisor. De la misma manera se envían diferentes mensajes presionando otros controles (invocación de métodos) del objeto controlRemoto, y a su vez se envían otros mensajes que invocan los métodos respectivos en el objeto Televisor.

Cuando usted está escribiendo un documento en un editor de texto cualquiera y presiona el botón Guardar Cambios, usted le está enviando un mensaje a la aplicación que le indica a ésta, que debe activar un método en ella, que permita guardar los últimos cambios en el documento.

Objeto

Un objeto es una entidad que representa dentro de un programa a un objeto del mundo real (que existe en el tiempo y espacio), el cual tiene características propias que describen su estado y acciones que determinan su comportamiento.

Una definición muy tradicional de lo que es un objeto, es aquella que lo considera como una cosa tangible o una cosa que se puede reconocer visual o conceptualmente. Cuando se dice que representa un objeto del mundo real, esto no quiere decir que el único tipo de objetos necesarios para la programación son los objetos tangibles. Para este proceso también se requiere identificar objetos que son intangibles y que se requieren para el funcionamiento de un sistema, aplicación o programa.

Se puede entender por objeto, cualquier unidad o entidad individual e identificable, ya sea real o abstracta, con un papel bien definido dentro del dominio de un problema dado.

Un objeto tiene estado, comportamiento e identidad. La estructura y el comportamiento de los objetos similares están definidos en la clase que los representa, es decir, la plantilla a partir de la cual fueron creados.

Desde el punto de vista del software como tal, un objeto es una elemento independiente de un programa diseñado para realizar tareas específicas y que hace parte de un grupo asociado de características llamadas clases.

Cuando se habla que el paradigma orientado a objetos tiene como modelo el modelo objeto, lo que se quiere indicar es que la idea fundamental de este paradigma es combinar en una sola unidad los datos y las funciones que operan sobre esos datos. Esta combinación es la que permite en términos de los conceptos del paradigma orientado a objetos, hablar de encapsulación, ya que los datos y las operaciones (funciones, métodos) se concentran o encapsulan en el mismo espacio.

Una manera de identificar objetos en un problema consiste en examinar la descripción del problema o la concepción que se tiene del dominio del problema, y seleccionar los sustantivos presentes. Una vez se hayan identificado los objetos entonces se procede a la identificación de los atributos y las operaciones presentes en ellos.

Los objetos dentro de un programa pueden representar cosas del mundo real, tangibles, roles o papeles desempeñados por personas, incidentes u ocurrencias, interacciones que implican una transacción que involucra dos o mas objetos (transacciones de venta, compra), especificaciones de cosas o de elementos, lugares.

Los atributos toman valores en un objeto ya que están describiendo un objeto particular. Los métodos en un objeto cambian el comportamiento o el estado de un objeto, es decir, cambian el valor de uno o más atributos que posee el objeto.

Así como en la vida real, los objetos en un programa se relacionan con otros objetos. El paradigma orientado a objetos traduce esas relaciones que se presentan en el mundo real, de tal manera que identifica relaciones tales como la herencia, la composición.

Herencia

Es la propiedad que permite a los objetos ser contruidos a partir de otros objetos. La herencia permite implementar las relaciones en las cuales un objeto pertenece a una clase que es hija de otra clase, lo que significa que, la clase hija hereda de su clase padre, la estructura y el comportamiento de ésta.

Las relaciones de herencia se establecen entre clases, puesto que, cobijan a todos los objetos que se van a crear como instancias de esas clases.

La clase padre se denomina **superclase**, es decir, la clase de la cual heredan o se derivan otras clases y las clase hijas, o las que heredan se denominan **subclases**.

Así como los objetos con estructura y comportamiento comunes se clasifican en una clase, de igual modo, las clases con propiedades y funciones comunes se agrupan en una superclase.

Existe dos tipos de herencia, a saber:

- Herencia Simple: cuando una subclase comparte la estructura y el comportamiento de una sola superclase. En otras palabras una subclase puede heredar datos y métodos de una única clase.
- Herencia Múltiple: se presenta cuando una subclase comparte la estructura y el comportamiento de una o más superclases, es decir, heredan datos y métodos de más de una clase.

La herencia múltiple es el tipo de herencia que encontramos más común en el mundo real. Obsérvese a sí mismo: Usted es una subclase que hereda de la superclase padre y la superclase madre. Quizá heredó entre otros, el color de

ojos, el color de cabello y la forma de las orejas, de su superclase madre. Heredó el tipo de nariz, los lunares y la forma de su boca, entre otros, de su superclase padre.

Al heredar Usted de su superclase padre y de su superclase madre, usted ya hace parte de una jerarquía de clases: el árbol genealógico de su familia. Sus superclases padres, han heredado a su vez de sus superclases abuela-materna, abuelo-materno, abuela-paterna y abuelo-paterno, y así sucesivamente. Lo que hace que, en definitiva, el objeto Usted herede características de las superclases de su superclases padre y madre.

Las jerarquías de clases se organizan en forma de árbol, como puede ser el árbol que agrupa a las plantas, a los animales, a los minerales, y tantos otros que se pueden identificar en la vida diaria.

Java implementa solo el tipo de herencia simple, es decir, una subclase solo puede ser hija de una sola superclase o clase padre. Sin embargo, Java utiliza un mecanismo que le permite implementar de manera diferente el tipo de herencia múltiple. Cuando usted crea un subclase de una clase dada, su subclase recibe automáticamente todo de su superclase, la estructura y el comportamiento, la cual a su vez las obtiene de su superclase y así sucesivamente por toda la jerarquía de clases. La clase que usted está creando se convierte en una combinación de todas las características de las clases superiores en la jerarquía, así como las propias.

La herencia por lo tanto, define una jerarquía de clases o jerarquía de tipos. En Java todas las clases se derivan de una clase raíz (base) o primitiva que es la clase **Object**, la cual es la clase más generalizada de la estructura de clases.

Cuando usted define una subclase de una clase dada, usted puede aumentar (añadir más datos y añadir código especial para la subclase) o restringir (cambiar elementos que necesita que sean diferentes) la estructura y el comportamiento existentes en sus superclases. Cuando una subclase aumenta la estructura y/o el comportamiento de su superclase se dice que utiliza **herencia por extensión**. Una subclase que modifica de alguna manera la estructura y/o el comportamiento de sus superclase se dice que usa **herencia por restricción**. En la práctica no siempre está claro cuando una subclase aumenta o restringe a su superclase, de hecho es habitual que las subclases hagan las dos cosas.

El objetivo final de la herencia es la **reutilización**, es decir la capacidad de utilizar código existente para crear nuevos objetos o nuevo código sin necesidad de empezar desde cero, ya que se pueden definir nuevos objetos a partir de clases existentes, lo que significa que, se está ahorrando el proceso de desarrollo de código ya que se hereda de una superclase que ya está implementa dentro de

una jerarquía especificada por el lenguaje de programación y si se necesita añadir o restringir algo de este código, se trabaja sobre lo que se hereda de la plantilla de la clase padre para hacerle las modificaciones que se requieran. Es decir, la herencia permite crear muchas clases que son similares entre sí, sin tener que re-escribir cada vez las partes que son similares. Esta propiedad permite modificar una clase existente sin modificar realmente el código original.

La herencia implementa la relación que se llama de generalización/especialización (**es un, is-a**), en la que una subclase hereda de una o más superclases. El comportamiento de los datos asociados con las clases hijas son siempre una extensión (estrictamente un conjunto más grande) de las propiedades asociadas con las clases padres. Una subclase puede tener todas las propiedades de la clase padre y otras adicionales propias de ella misma. Por otro lado, dado que la clase hija es una forma más especializada de la clase padre, es también, en cierto sentido, una nueva definición del tipo padre.

Clase Abstracta

Una clase abstracta es una clase que no posee instancias, es decir una clase a partir de la cual no se pueden crear objetos. Sirve como plantilla para crear otras clases, que se llaman **Clases Concretas** o **Clases Derivadas**.

Una clase abstracta en su definición posee métodos que únicamente son nombrados, pero cuya programación, implantación o implementación se deja para ser realizada dentro del cuerpo de sus subclases concretas o derivadas. Una clase abstracta se define con la intención que sus subclases concretas añadan elementos nuevos a su estructura y comportamiento, normalmente escribiendo el código de sus métodos abstractos. Por lo general, las clases abstractas se utilizan como raíz de una jerarquía de clases.

Las clases abstractas se utilizan para agrupar otras clases y capturar información que es común a un grupo. Se dice que las clases abstractas corresponden a un concepto abstracto del mundo real y que por ende no existen representantes de ella en el mundo. Sin embargo las subclases de clases abstractas, o sea, las clases concretas que corresponden a objetos del mundo real pueden tener instancias.

Visibilidad

Son especificadores que desde la definición de una clase, se utilizan para controlar el acceso a datos y métodos de la clase fuera de ella. Los datos y los métodos de una clase pueden tener uno de los siguientes especificadores de visibilidad:

- Privado (private): define a aquellos datos o métodos que no pueden ser accedidos por subclases o por otras clases. Cuando los datos y/o los métodos están marcados con este tipo de visibilidad solo los métodos de la misma clase pueden interactuar con ellos.
- Público (public): define a aquellos datos y métodos a los cuales se puede acceder libremente desde dentro o desde fuera de la clase. Estos datos y métodos son accesibles por otras clases con las cuales la clase actual se relaciona o con sus propias subclases si las tiene.
- Protegido (protected): define a aquellos datos y métodos que son parcialmente visibles, es decir son accesibles desde ella misma y desde sus subclases pero no son accesibles por otras clases con las que se relaciones diferentes de sus subclases.

Durante el análisis de un problema, se pueden ignorar perfectamente los problemas de visibilidad, pero una vez se comienza a idear cómo se comenzará a programar hay que considerar la visibilidad, teniendo en cuanta las relaciones que una clase puede tener con otras, porque las decisiones que se tomen al respecto influyen dentro del ámbito del programa.

Recapitulación

Se ha visto que en el mundo de los objetos, el elemento fundamental es el objeto. Un objeto es algo que es tangible, identificable y hace parte del mundo real. Los objetos con características similares se agrupan en una Clase. Un clase es la definición de una agrupación de objetos con características similares, donde dichas características están representadas por la estructura y el comportamiento. Entonces una clase tiene Estructura y Comportamiento. La estructura está definida por los atributos que los objetos agrupados en esa clase poseen y el comportamiento está definido por los métodos u operaciones que los objetos agrupados en esa clase realicen. Los métodos se pueden clasificar en métodos que el objeto realiza por sí mismo y métodos que se efectúan como respuesta a un evento o suceso que otro objeto ha generado. Nuestro alrededor esta compuesto de objetos.

A los objetos también se les llama instancias. Hay objetos que se componen de otros objetos, o sea son la agregación de partes, en donde las partes son objetos también. A esto se le llama Composición.

Las clases que tienen estructura y comportamiento similares se pueden agrupar en una clase más general que contenga tanto los atributos como los métodos comunes a dichas clases. A esto se le denomina jerarquía de clases.

Los objetos se relacionan entre sí. La comunicación entre objetos a través de órdenes se produce gracias al envío de mensajes. Un mensaje es una orden o una información que un objeto manda a otro con el fin que el objeto que recibe el mensaje pueda activar un método.

Así mismo se han presentado las diferencias entre el enfoque de objetos y el enfoque procedimental o algorítmico. Las definiciones básicas de la programación orientada a objetos así como las características principales del enfoque de objetos.

Ejercicios

1. Identifique en su salón de clase los objetos existentes. Caracterícelos como objetos.
2. Tome los siguientes objetos: un teléfono celular, una calculadora de bolsillo, una lavadora eléctrica, una licuadora, un horno microondas. Para cada uno de ellos:

Identificarlo y caracterizarlo como objeto.

Determinar sus atributos y métodos.

Dibujar el esquema de representación como objeto.

3. Tome los siguientes animales domésticos, referidos a objetos reales que existen en algún lugar: un perro, un gato, un loro, un caballo, un conejo. Establecer sus atributos y características como objetos. Establecer la clase que agrupa a cada uno de ellos. Establecer la jerarquía de clases de cada uno de ellos hasta llegar a la superclase común de Seres Vivos.
4. Tome los siguientes objetos plantas, que usted localice en la realidad: tomate, rosa, naranjo, zapayo, mango, frijol. Establecer sus atributos y características como objetos. Establecer la clase que agrupa a cada uno de ellos. Establecer la jerarquía de clases de cada uno de ellos hasta llegar a la superclase común de Seres Vivos.
5. Para cada uno de los objetos de los dos problemas anteriores, hacer la representación gráfica, indicando solamente tanto los atributos como los métodos que le son propios como objeto, si los tiene, ya que los demás son heredados de sus superclases o ancestros.
6. Ilustre mediante un diagrama una jerarquía de generalización, donde los objetos puedan tener herencia de varias superclases.

7. Tome los objetos del problema de obtener un bloque de madera a partir de un arbol y establezca las clases respectivas con sus atributos y métodos.
8. Ilustre las relaciones de composición para el caso de una bicicleta.
9. Identifique los objetos involucrados en el caso del prestamo y devolucion de material bibliografico en la Biblioteca de la Universidad.
10. Diseñar una clase Rectangulo, con atributos largo y ancho, ambos con valor por omisión de 1. La clase debe tener métodos para calcular el perímetro y el área. Debe agregarse metodos para verificar que los valores suministrados sean mayores que cero.
11. Diseñar una clase Triangulo, que tenga como atributos base y altura, así mismo de métodos para calcular su área y perímetro.
12. Diseñar una jerarquía de clases a partir de la clase abstracta FigurasGeometricas, para las clase Triangulo, Cuadrado, Circulo. En estas clases se implementarán métodos para calcular el área y el perímetro respectivamente.
13. Diseñar una clase Sumatoria para calcular la siguiente serie:

$$S = 1 - 2! + 3! + 4! - 5! + 6! + 7! + 8! - 9! + 10! + 11! \dots n!$$

Incluya un método para el cálculo del factorial.