INGENIERÍA DE SOFTWARE I

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Unidad III

Tema: Modelado del Sistema con OO

UNTDF – 2020

Problemas del software

- Los sistemas de software suelen ser complejos porque modelan situaciones del mundo real.
  - Modificar estas aplicaciones involucra un esfuerzo que no debería desperdiciarse y si cambia el mundo real, deberá poderse modificar la aplicación sin que eso ocasiones problemas
  - De igual manera, agregar una nueva funcionalidad a las aplicaciones no debería consistir mas que en agregar solo el código de la funcionalidad
  - Debería ser posible construir nuevas aplicaciones a partir de componentes previamente implementadas y probadas

Problemas de metodologías

- Las metodologías convencionales de análisis, diseño y programación no favorecen el reuso y mantenimiento, pues información (datos) y el comportamiento (funcionalidad) se encuentren dispersos en el sistema.

- Las modificaciones realizadas a la funcionalidad o a la representación datos de un componente en un sistema tiene un impacto difícil de especificar en otros componentes.

Problemas que enfrenta OO

- La orientación a objetos es una herramienta conceptual que sirve para atacar los siguientes problemas:
  1. Necesidad de construir software modificable y extensible.
  2. Necesidad de disponer de medios para obtener software reusable
  3. Necesidad de disminuir el gap semántico entre el mundo real y el de los modelos (modelo natural para representar el mundo).

Por qué OO

- **Proximidad de los conceptos de modelado respecto de las entidades del mundo real**
  - Mejora captura y validación de requerimientos
  - Acerca el “espacio del problema” y el “espacio de la solución”
- **Modelado integrado de propiedades estáticas y dinámicas del ámbito del problema**
  - Facilita construcción, mantenimiento y reutilización

Por qué OO

- **Conceptos comunes de modelado durante el análisis, diseño e implementación**
  - Facilita la transición entre distintas fases
  - Favorece el desarrollo iterativo del sistema
  - Disipa la barrera entre el “qué” y el “cómo”

- **La forma de pensar en objetos es más natural**. El diseñador piensa en términos de objetos y no en detalles de bajo nivel.

Beneficios de OO

- **Facilidad en el mantenimiento y la extensión del software**

- **Reutilización.** Permite el reuso de código y de diseño empleando menos tiempo de desarrollo.

- **Integridad**. Los mecanismos de encapsulación protegen sus propios componentes contra los procesos que no tengan derecho a acceder a ellos.

- **Programación más sencilla**. Los programas se crean a partir de piezas pequeñas.

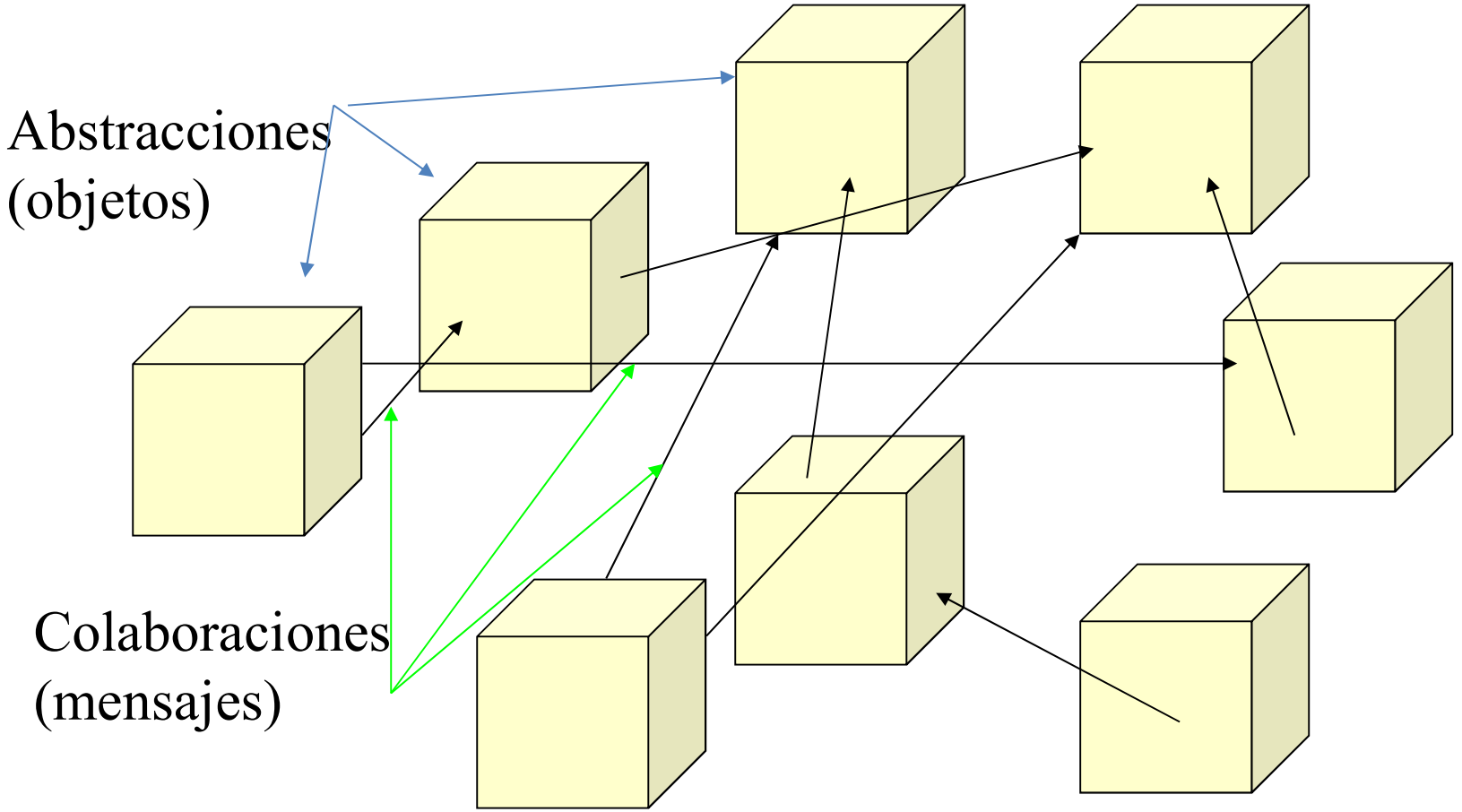
El Paradigma OO

- En el paradigma de orientación a objetos, *un sistema se concibe como un conjunto de objetos que interactúan entre si a través de* mensajes.

- Mediante este modelo se construyen más fácilmente sistemas complejos a partir de componentes individuales.

El Paradigma OO

Aplicación: Conjunto de objetos cooperantes



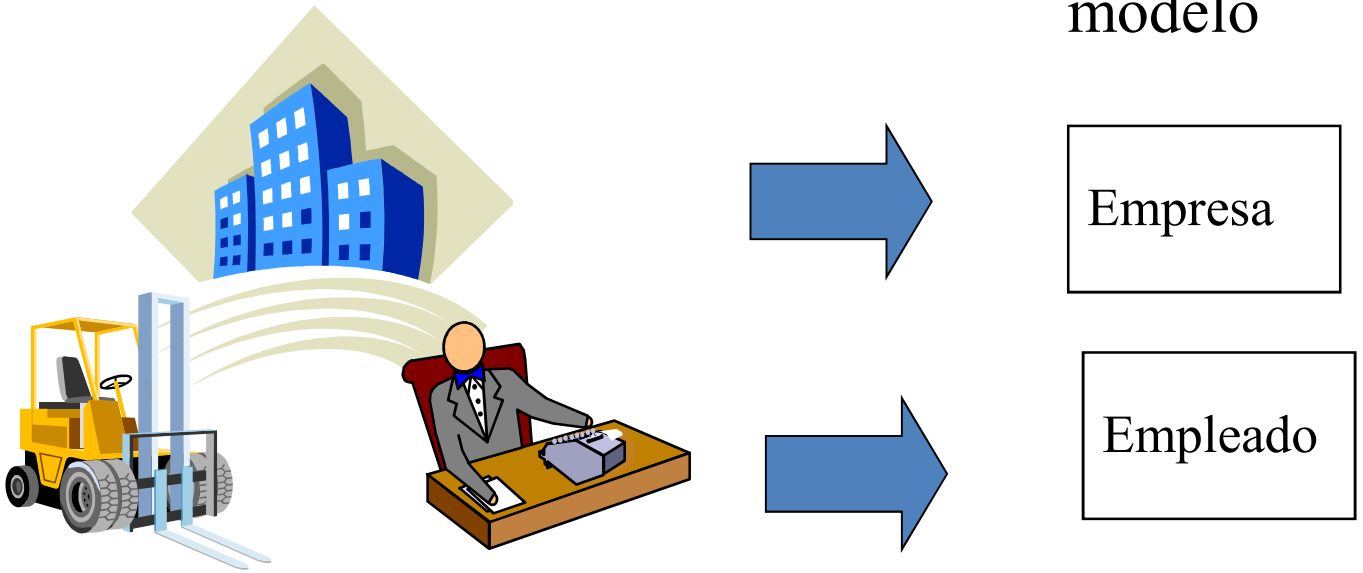
Se descompone el sistema en OBJETOS, la componente básica

Modelado del Sistema con OO

- **OO provee objetos como el principal medio para abstraer y estructurar un sistema**.
- Modelar consiste en identificar (abstraer) qué objetos hay en el mundo del problema, cómo son, cómo se comportan y cómo se relacionan.

  - Cada objeto tiene un rol en la solución del problema
  - Cada objeto provee un conjunto de servicios (o métodos)
  - Los servicios de un objeto son usados por otros objetos.

Modelado del sistema con OO



"El modelado captura las partes esenciales del sistema"

Modelado con Objetos

- *Un objeto representa una entidad individual identificable, real o abstracta, con un papel definido y preciso en el dominio del problema* (abstracción de un elemento del contexto del problema)

- Un objeto puede modelar:
  - entidades del mundo real concretas (un avión, una organización, un empleado…),
  - abstractas (un vuelo, una reserva, ….),
  - transacciones (préstamos, devoluciones)
  - artefactos de software (gráficos, ventanas…)
  - estructuras de datos (pilas, listas, …)

Modelado con Objetos

- El *análisis y diseño orientado a objetos* modela el mundo en términos de objetos que tienen **estado** y **comportamiento**, y **eventos** que activan operaciones que modifican el estado de esos objetos.

- Cada objeto del sistema se hace responsable de una parte de las responsabilidades de la aplicación.

- Los objetos interactúan de manera formal con otros objetos mediante **mensajes**.

Modelado con Objetos

- El estado interno del objeto se implementa mediante un conjunto de propiedades o *atributos encapsulados*

- Las responsabilidades (o servicios) que realiza el objeto se implementan mediante *métodos*, los cuales describen el comportamiento del objeto.
- Existen dos tipos de responsabilidades que pueden ser asignadas a un objeto:

  1.- El conocimiento que mantiene el objeto.

  2.- Las acciones que puede realizar.

Modelado con Objetos. Objetos SW

- Todo **objeto SW** tiene estado, comportamiento e identidad
  - El **estado** está representado por los valores de los atributos
  - El **comportamiento** agrupa las competencias de un objeto y describe las acciones y reacciones de ese objeto
  - El c**omportamiento** es la actividad “visible” del objeto o su “interfaz”. Le provee comunicación con el resto del sistema.

Ejemplo. Objeto SW

Mundo real

Abstracción del modelo de Análisis

Empleado

Objeto Software

unEmpleado

nombre
antigüedad
hsTrabajadas
salario….

estado interno: atributos

Responde a los mensajes:
mostrarNombre
aceptarNombre
calcularSalario
……

comportamiento: métodos

Interfaz pública

Un objeto software *es cualquier entidad, real o abstracta, acerca de la cual almacenamos datos (estado) y las operaciones que controlan dichos datos (comportamiento).*

Conceptos OO: Encapsulación

- El acto de agrupar en un objeto simple datos y operaciones que afectan los datos.
- El conocimiento encapsulado en un objeto, puede esconderse de la vista externa.
  - ✓ Así el objeto tiene una interfaz pública que presenta a las otras entidades dentro del sistema.
- La intefaz pública consiste de qué puede hacer y decir.
  - ✓ Las otras entidades conocen sólo cómo pueden interactuar con él.
- El concepto de encapsulación se refiriere a construir una cápsula, es decir una barrera conceptual alrededor de una colección de datos

Conceptos OO: Ocultamiento información

- En OO, un objeto tiene una *interfaz pública y una representación privad*a y *mantiene estas dos facetas en forma distinta*.
  - ✓ Este principio es conocido como ocultamiento de la información.
- El ocultamiento de la información permite remover de la vista aquello que ha sido encapsulado por el objeto.
- Gracias a esto, los otros objetos conocen sólo qué operaciones pueden pedirle que ejecute al actual.
- *La encapsulación y el ocultamiento de la información trabajan juntos para aislar una parte del sistema de otras partes, permitiendo al código ser modificado y extendido.*

Conceptos OO: Ocultamiento información

De lo anterior se deriva que los objetos tienen:

1. Una *parte privada* → Estado interno

2. Una *parte pública* → Operaciones

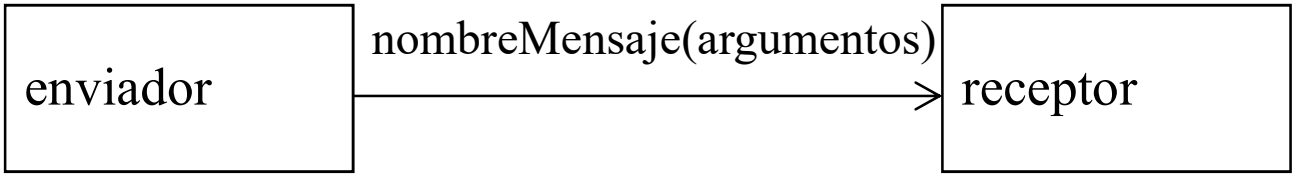
- La parte privada es información oculta al resto de los objetos
- El estado de un objeto es un conjunto de atributos que pueden ser modificados o consultados por sus operaciones; sólo las operaciones del objeto pueden modificar su estado interno.

Conceptos OO: Ejemplo Ocultamiento información

- **Método calcularSalario()**
- Un objeto Empleado contiene un dato *salario* y un método *calcularSalario()* escrito con un código para el cobro normal del salario.
- Si cambia la forma de calcular el salario de un empleado, por ej. por una bonificación especial, solamente de debe modificar el código que se encarga de calcular Salario en los objetos Empleado.
- El ocultamiento de información permite asegurar que el dato salario sólo se encuentra en el objeto empleado y sólo puede modificarse por las operaciones de este objeto.
- Por lo tanto, no es necesario revisar todo el sistema buscando efectos secundarios.

Conceptos OO: Mensajes

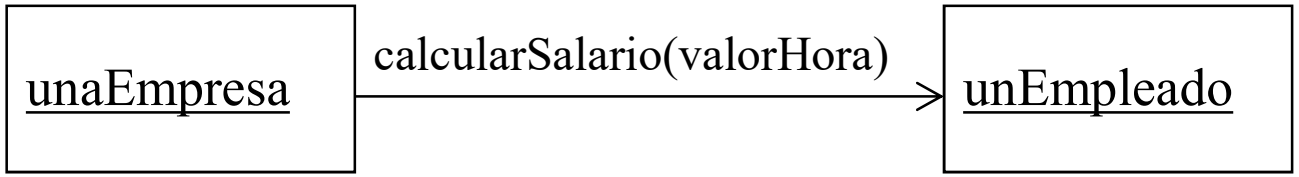
- *Un objeto accede a otro objeto enviándole un mensaje.*
- Un mensaje consiste del nombre de la operación y de los argumentos requeridos.



- Cuando un objeto envía un mensaje a otro, el receptor ejecuta la operación nombrada.
- ***El conjunto de mensajes a los que un objeto puede responder se denomina "comportamiento del objeto".***

Conceptos OO: Métodos

- Cuando un objeto recibe un mensaje, responde a la operación requerida, ejecutando un método.
- Un método es un algoritmo ejecutado en respuesta a un mensaje recibido, cuyo nombre coincide con el nombre del método.



El objeto unEmpleado ejecuta el método calcularSalario en respuesta al mensaje recibido

- La **Signatura** consiste del nombre de método, el tipo de sus parámetros y el tipo del objeto que el método retorna.
- La signatura es una especificación formal de las entradas y salidas de un método.

Conceptos OO: Ejemplos de Objetos

- Cualquier entidad a la cual se le puede asignar un conjunto de responsabilidades, puede considerarse un objeto

- Pila:
  Comportamiento: puede recibir los mensajes “apilar”, “desapilar”, “estar_vacia”, “ver_tope”.
  Su estado interno estará representado por un conjunto de elementos y un indicador de cuál es el tope.

- Arreglo:
  Su comportamiento es devolver el elemento en el lugar i, modificar el contenido del lugar i, colocando el objeto x , recorrer los objetos del arreglo
  Su estado interno será su nombre y dimensión

Conceptos OO: Mecanismos de abstracción

- No es práctico repetir definiciones de objetos
- Se necesitan herramientas conceptuales para realizar abstracciones.

- En el *paradigma de objetos* existen dos ejes conceptuales para *abstraer conceptos:*

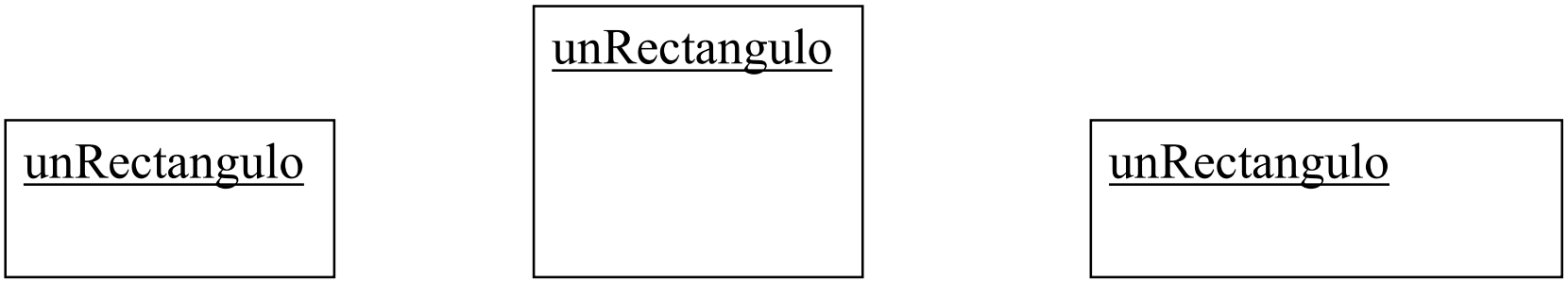
➢ *Clasificación.*

➢ *Generalización / especialización.*

Conceptos OO: Clasificación. Clases

Clases:

- Se puede observar que existen objetos que tienen el mismo comportamiento



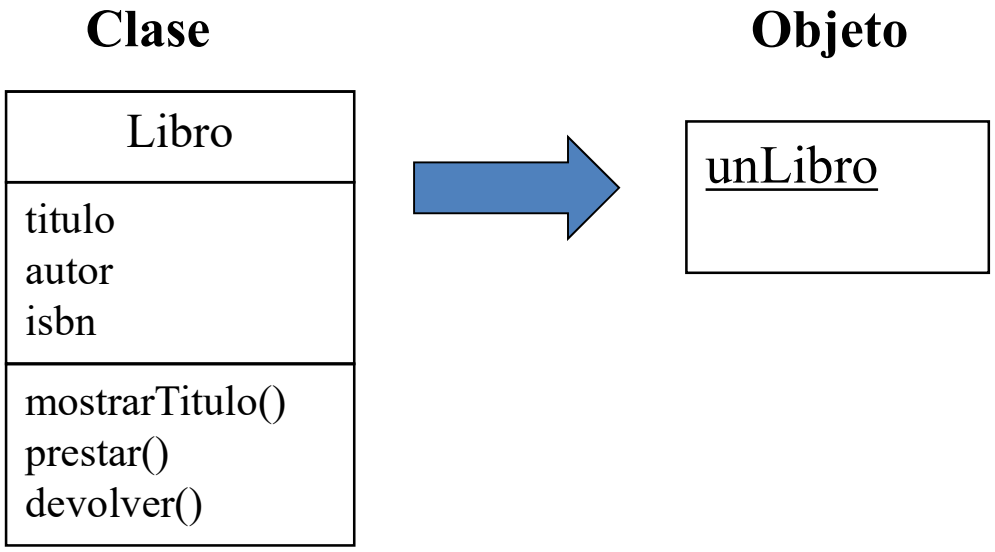
- Estos objetos tienen el mismo comportamiento: dibujarse, pintarse de un color, devolver ancho, devolver altura, etc.

- *Los objetos con comportamiento similar se agrupan en clases*

Conceptos OO: Clasificación. Clases

Clases:

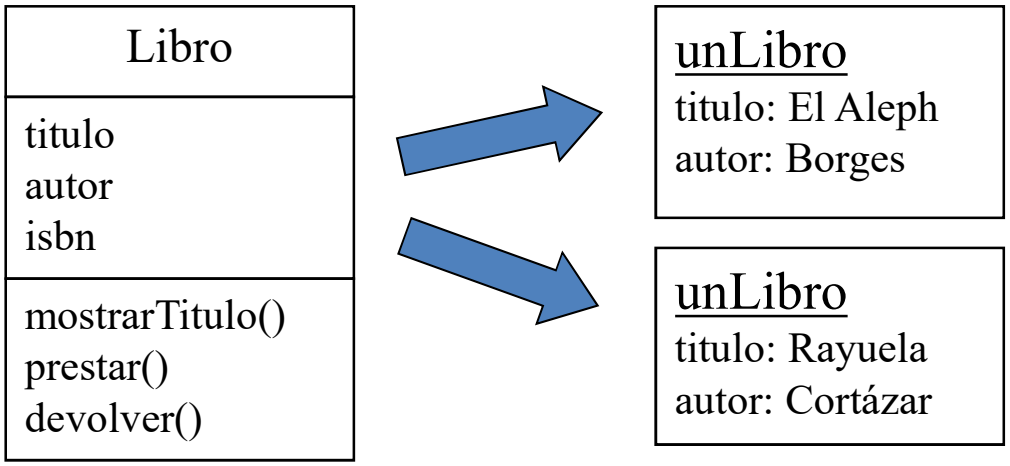
- Una clase es una especificación genérica para un número arbitrario de objetos similares.
- Las clases permiten describir en un lugar el *comportamiento genérico de un conjunto de objetos* y luego, crear objetos que se comportan en esa manera, cuando se necesitan.



Conceptos OO: Clasificación. Instancias

Instancias:

- Los objetos que se comportan de una manera especificada por una clase, se llaman instancias de esa clase.
- Una vez creada alguna instancia de una clase, se comporta como todas las otras instancias de su clase, con capacidad para recibir mensajes y ejecutar las operaciones para las cuales tiene métodos especificados en su clase



Cada instancia tiene un identificador interno (OID)

Las instancias toman diferentes valores de atributos

Conceptos OO: Clasificación. Instancias

Instancias – Estado y comportamiento:

- **El Estado** agrupa los valores instantáneos de todos los atributos de un objeto; un atributo es una información que cualifica al objeto que la contiene.
  - ✓ El estado evoluciona con el tiempo, es variable y puede verse como la consecuencia de sus comportamientos pasados.

- **El comportamiento** agrupa todas las competencias de un objeto y describe sus acciones y reacciones.
  - ✓ Cada átomo de comportamiento se llama operación.
  - ✓ Las operaciones de un objeto se desencadenan a consecuencia de un estímulo externo, representando en forma de un mensaje enviado por otro objeto.

Conceptos OO: Clasificación. Instancias

Instancias – Identidad

- *Distingue los objetos de forma no ambigua, independientemente de su estado.*
- Distingue dos objetos en los que todos los valores de atributos son idénticos.
- La identidad es un concepto, no se representa de manera específica en el modelado. Cada objeto posee una identidad de manera implícita.
- *Cada objeto posee un oid. El oid establece la identidad del objeto.* Es determinado en el momento de la creación del objeto

Conceptos OO: Clases e Instancias

- La clase es un molde para aquellos objetos que poseen las mismas características, que pueden recibir los mismos mensajes y que responden de la misma manera.
- Es una fábrica de objetos
- Contiene el código de los métodos

- A las variables que describen atributos de objetos se las llama variables de instancia.
- ***Métodos de clase***: son aquellos que reflejan responsabilidades asignadas a la clase en sí. Por ejemplo, la creación de objetos.
- ***Métodos de instancia***: son aquellos que implementan responsabilidades asignadas a las instancias de una clase.

Conceptos OO: Clasificación. Instancias

Clases como fábrica de objetos

- La clase es la responsable de la creación de los objetos que pertenecen a ella.
- Esta responsabilidad se traduce en un mensaje de la clase, cuyo resultado es un objeto de la clase que recibió el mensaje.
- Si denominamos el mensaje de creación “new”, la instanciación de la clase “Libro”, se realiza de la siguiente manera:

Libro new instanciación de la clase Libro.

Se envía el mensaje new a la clase Libro

Ej: unLibro := Libro new

La clase Libro recibe un mensaje new, que crea un objeto Libro.

Conceptos OO: Herencia

- Existe otro mecanismo de abstracción: la herencia.

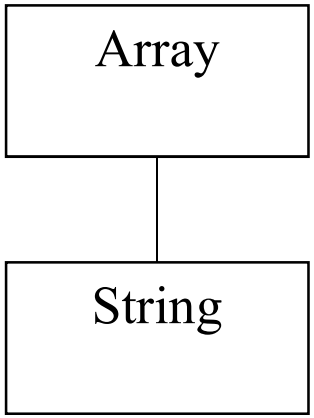
- *Es la habilidad de una clase de definir el comportamiento y la estructura de datos de sus instancias como un superconjunto de la definición de otra clase o clases.*

- Permite concebir una nueva clase de objetos como un refinamiento de otra, abstraer las similaridades entre clases y diseñar y especificar sólo las diferencias para la nueva clase.
- La herencia permite el reuso del código. Las clases pueden heredar comportamiento. Si la clase puede heredar algo del código que hereda, entonces no es necesario reimplementar esos métodos.

Conceptos OO: Herencia

Subclase

- Una subclase es una clase que hereda comportamiento de otra clase. Usualmente agrega su propio comportamiento para definir su propio tipo de objeto.



Un string se comporta como un array, porque sus elementos pueden accederse por un índice. Sin embargo tienen características adicionales, por ej. ponerse en orden alfabético.

Conceptos OO: Herencia

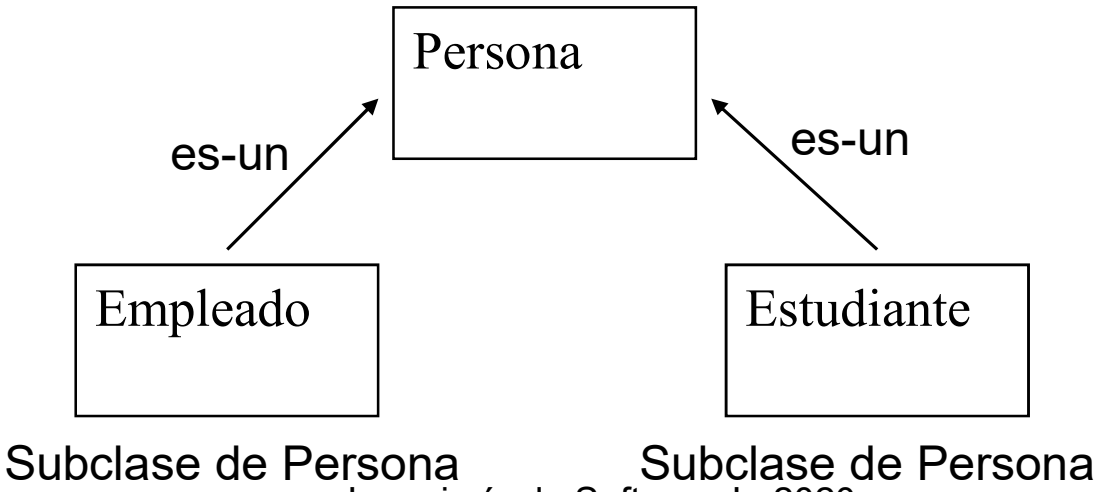
Superclase

- Una superclase es una clase de la cual se heredan sus atributos y su comportamiento.
- Una clase puede tener una o varias superclases, combinando comportamiento de distintas fuentes y agregando sólo un poco de si misma para producir su propio tipo de objeto.

Conceptos OO: Herencia

Jerarquías de Herencia

- El *mecanismo de abstracción* de la generalización / especialización, consiste en construir *jerarquías de clase / subclase*, donde, en general las clases están vinculadas por la relación es-un.
- Por ej., describimos la clase empleado como sub-clase de persona (un empleado es-una persona)
- Análogamente la clase estudiante es sub-clase de persona.



Conceptos OO: Polimorfismo

Polimorfismo

- Es la habilidad de dos o más clases de objetos de una misma jerarquía, de responder a mensajes con la misma signatura, cada uno con su propio comportamiento.