



Ciclo 2

Semana 2

Modelamiento y Modelo Vista Controlador (MVC)

Lectura 1 - Nuevos elementos de modelado

| Nuevos elementos de modelado



1. Tipos Simples de Datos

En los lenguajes de programación existe unos tipos de datos con los cuales se representan los atributos de las clases. Se conoce al termino literal de un tipo de datos al valor constante de dicho tipo.

Entre los datos tipos simples de datos se encuentran:

- **Enteros (tipo int):** permiten modelar características donde sus valores posibles son numéricos y que sirven por ejemplo para definir los valores como fecha. Estos se usan cuando se manejan valores enteros sin decimales. Los literales de tipo int se enuncian como una secuencia de dígitos. Si este es negativo, la secuencia se precede con el símbolo "-".
- **Reales (tipo double):** representan valores numéricos de tipo real, como por ejemplo el interés de una tarjeta de crédito. Usado para facilitar el modelamiento de características que tienen valores reales, aquellos valores que contienen parte fraccionaria. La precisión de estos tipos de datos es dos veces más que la de los números del tipo float. Los literales de tipo double se expresan como una secuencia de dígitos donde la parte entera se separa de la decimal con un punto ".".

- **Cadenas de caracteres (tipo String):** representan dentro de una clase una característica como el nombre de una persona, pues son la sucesión de caracteres. Los literales de tipo String van entre comillas dobles y si estas comillas dobles van seguidas representan una cadena de caracteres vacía. Dos comillas dobles seguidas indican una cadena de caracteres vacía, pero sería diferente si tiene un espacio pues sería una cadena que tiene un solo carácter de espacio en blanco.

Ahora hablaremos de dos nuevos tipos de datos como los son el Boolean y Char.

- **Booleano (tipo boolean):** es un tipo simple de dato que representa valores lógicos como verdadero o falso. Este se usa en características que pueden tener dos opciones, por ejemplo, saber si un producto está gravado con IVA o no, siendo estos los únicos valores posibles y en cuyo caso en esta solo nos interesa saber si está o no grabado mas no el valor del IVA. El tipo boolean sólo tiene dos literales: true y false y son los únicos valores que se asignaran a los atributos o variables que tienen este tipo.
- **Caracter (tipo char):** utilizado para representar caracteres y se representa internamente con un código numérico llamado **UNICODE**. Los literales de tipo char se expresan entre comillas sencillas diferente a los literales de la clase string, que van entre comillas dobles. Se diferencia del tipo string porque el char es tomado por el sistema como un solo carácter, en cambio el string es una cadena de caracteres.

2. Constantes

Los lenguajes de programación permiten la relación de un nombre significativo con cada de los valores seleccionados y así representar una característica, en donde los métodos, en cambio de usar valores enteros seleccionados, pueden usar nombres asociamos a ellos denominados constantes, en otras palabras, las constantes representan aquellos posibles valores de una característica.

Otro uso dado a las constantes es que son inmutables, o sea que representan aquellos valores que no cambiaran durante la ejecución del programa y se les define un nombre para reemplazar

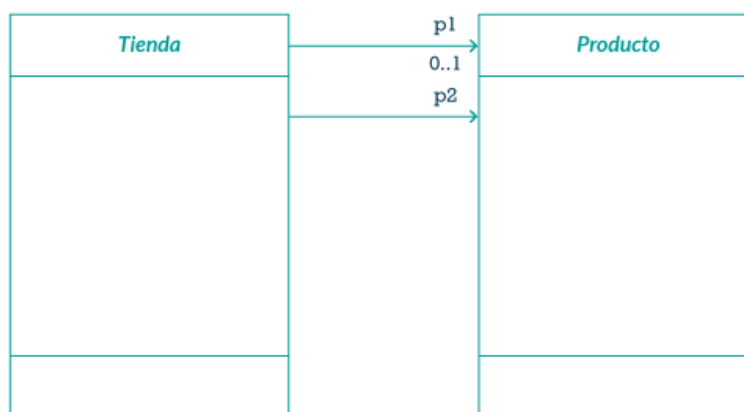


el valor numérico dentro del código. Estas constantes que pueden tener cualquier tipo de datos se les asigna un valor desde la declaración que no puede ser modificado en ningún punto del programa.

Los usos de constantes facilitan el mantenimiento de un programa, ya que, si dentro del programa se utiliza una constante recurrentemente y en algún momento dado su valor cambia, solo basta con reemplazar dicho valor en la declaración de la constante y no es necesario ir por todo el programa buscando aquel valor ingresado si por el contrario no lo hubiésemos definido como constante sino solo se hubiera colocado el valor directamente.

3. Asociaciones Opcionales

Las asociaciones que se tiene entre clases son opcionales, característica conocida como cardinalidad de la asociación y define el número de instancias de una clase que se usan a través de una asociación. Para expresar la cardinalidad, se utilizan dos números separados con dos puntos, pues en la asociación se pueden manejar un objeto de otra clase o ningún objeto.



La cardinalidad de la asociación llamada en la ilustración p1 entre una clase Tienda y una clase Producto es cero o uno (0..1), indicando que podría o no existir un objeto que representa la asociación p1. Si en el diagrama no se establecen cardinalidades en una asociación como p2, se dice que esta es 1, indicando que existe exactamente un objeto de la otra clase. Si dentro del método, el objeto correspondiente a una asociación no existe, por ejemplo, no hay un objeto de la clase Producto para la asociación p1, se coloca el valor especial **null** (**p1 = null;**), en cuyo caso si se intentara llamar un método a través de una asociación que tiene valor null, el computador mostrará el error: **NullPointerException**.