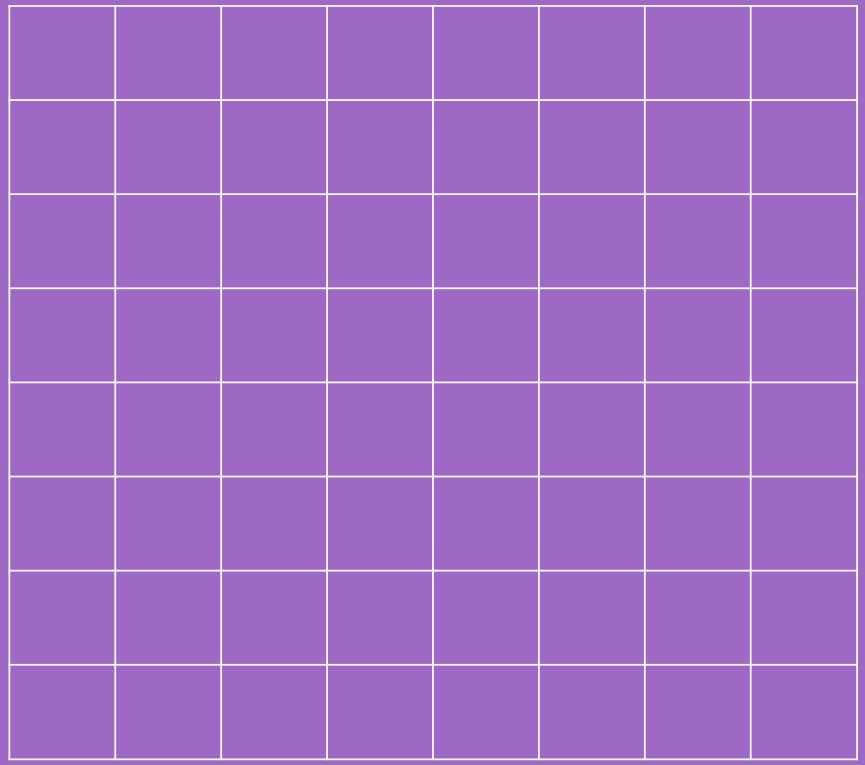
Argentina programa 4.0

Introducción a la programación OO en Java

"Desarrollador Java Inicial"

Concepto básico y definición



Clases y Objetos - De tipos simples a conceptos complejos

Argentina programa 4.0

Hasta ahora trabajamos con conceptos simples como un número, una letra, un texto o vectores de los mismos, pero ¿qué pasa si queremos representar cosas más complejas?

- ***una persona***
- ***un producto***
- ***un carrito de compras***
- ***un torneo***

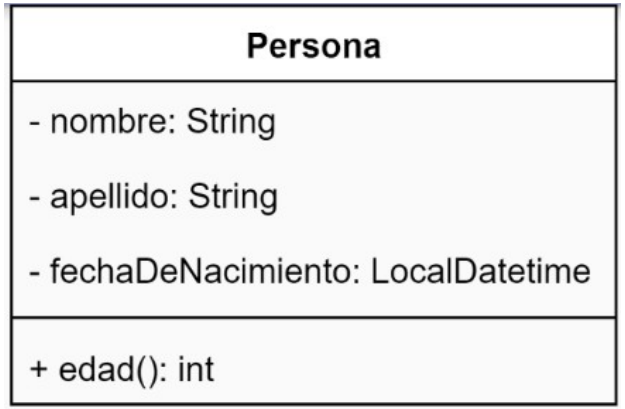
También, relaciones entre las mismas, por ejemplo que una persona tiene un carrito de compras y el carrito de compras tiene productos.

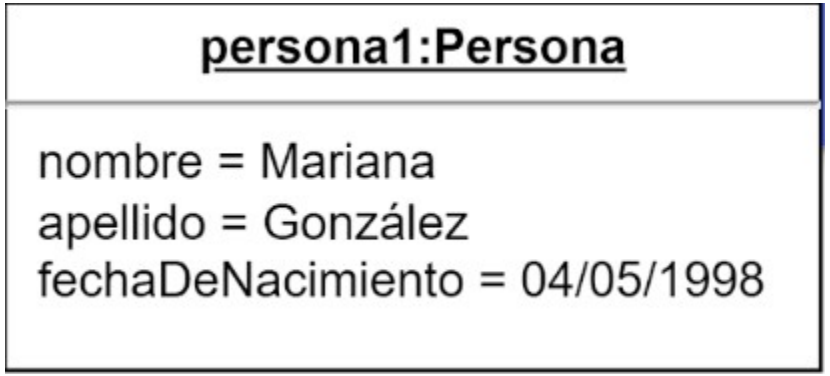
Clases y Objetos - Definición

Argentina programa 4.0

Para representar conceptos más complejos utilizaremos clases y objetos:

- Una clase es la representación de una entidad o concepto, la cual posee atributos (concepto similar a variable) y métodos (concepto similar a función). Sin embargo la clase describe características generales
- La forma más sencilla de definir un objeto, es describirla como una instancia de una clase. Es decir, posee las características generales de la misma, pero con valores concretos





Clases y Objetos - Representación

Argentina programa 4.0

Persona
- nombre: String - apellido: String - fechaDeNacimiento: LocalDatetime
+ edad(): int

Producto
- nombre: String - codigo: String - precio: float
+ costoFinal(int cantCuotas): float

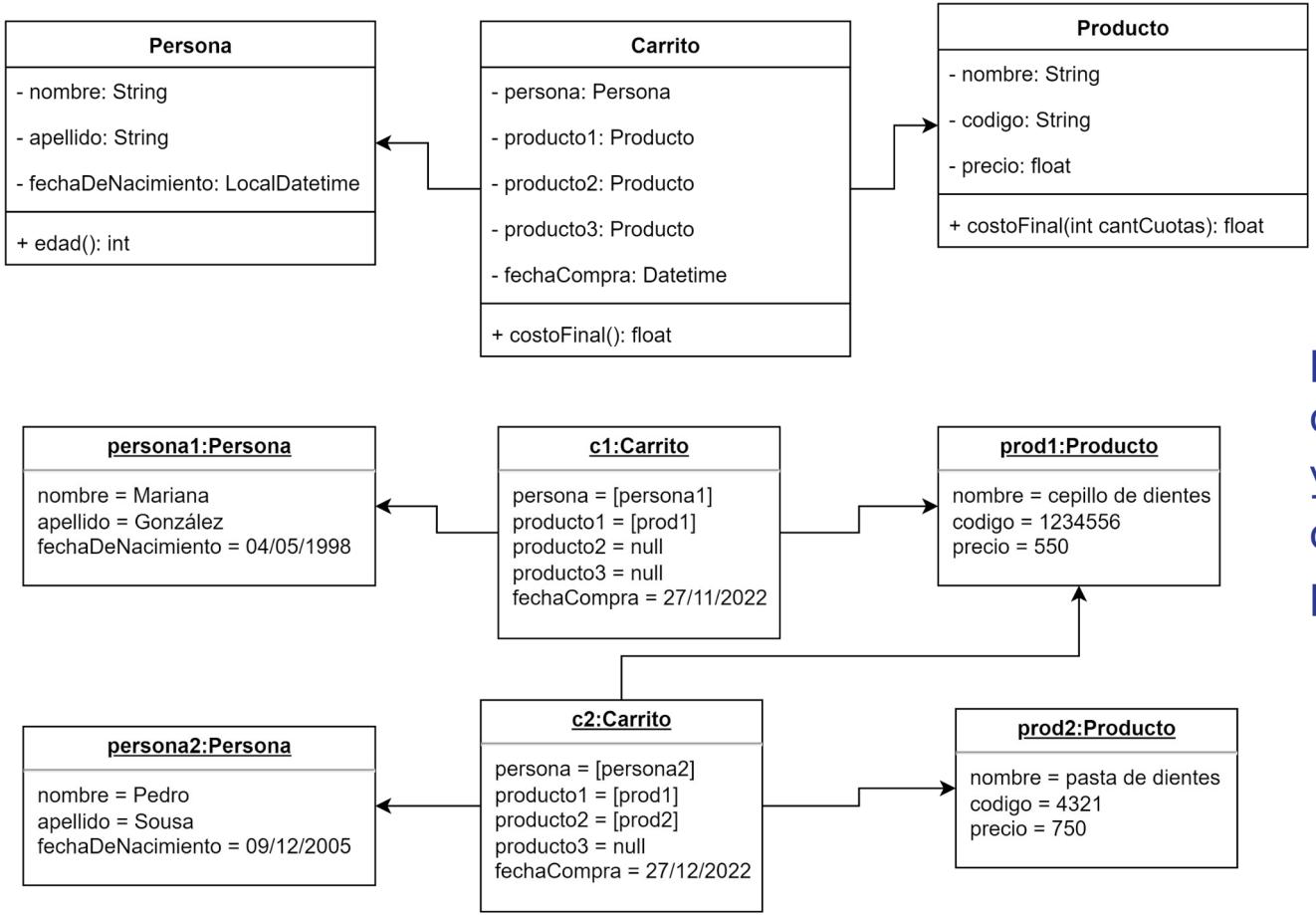
persona1:Persona
nombre = Mariana apellido = González fechaDeNacimiento = 04/05/1998

persona2:Persona
nombre = Pedro apellido = Sousa fechaDeNacimiento = 09/12/2005

prod1:Producto
nombre = cepillo de dientes codigo = 1234556 precio = 550

prod2:Producto
nombre = pasta de dientes codigo = 4321 precio = 750

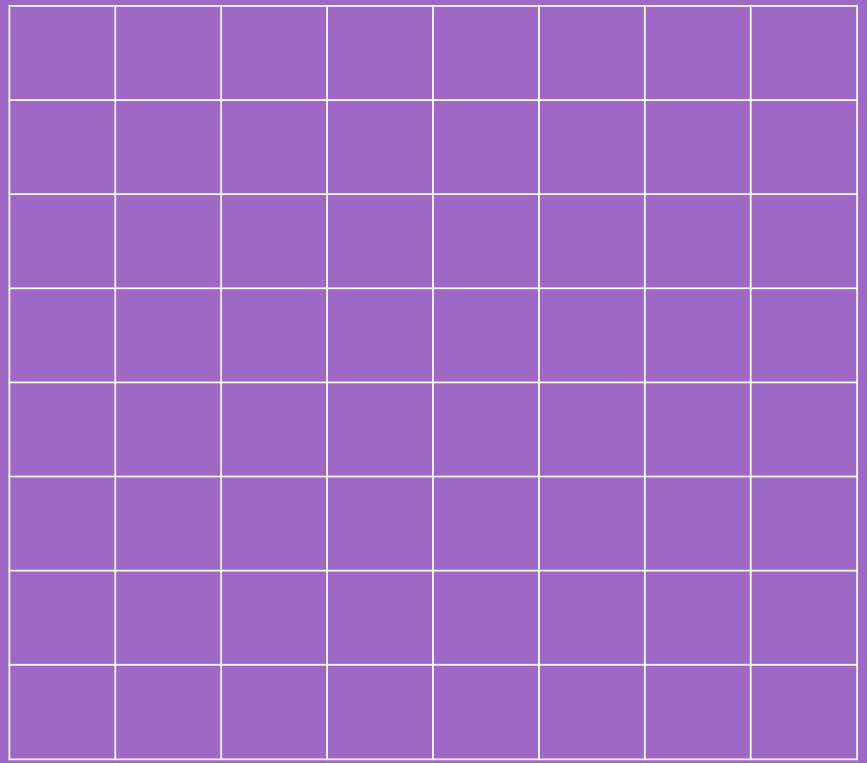
Clases y Objetos - Relaciones



Argentina programa 4.0

persona tiene un carrito de compras y el carrito de compras tiene productos

Implementación en Java



Clases y Objetos - Java

```
package ar.edu.utn;
```

El paquete de una clase se declara en la primer línea y permite que existan varias clases con el mismo nombre

```
import java.time.LocalDate;
```

Los import permiten utilizar clases de otros paquetes más convenientemente

```
public class Producto {
```

Nombre de la clase

```
 private String nombre;
 private String descripcion;
 private LocalDate fechaAlta;
 private Integer pesoKg;
 private Double precio;
```

Atributos, todos los objetos de una misma clase tienen los mismos atributos, pero con distintos valores

```
 private static final Double minPrecio = 0.1;
```

Atributos de clase o estáticos, su valor es compartido por todos los objetos de la clase

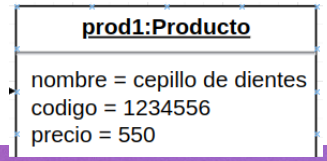
```
 public Producto(String nombre) {
   this.nombre = nombre;
   this.fechaAlta = LocalDate.now();
   this.precio = minPrecio;
 }
```

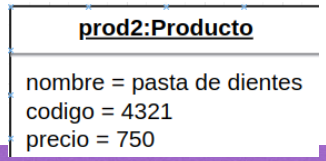
Constructor, se utiliza para crear objetos de la clase en cuestión. En este caso si queremos crear un nuevo producto la sintaxis es:

```
Producto prod = new Producto("Atún");
```

```
public String getNombre() {
   return this.nombre;
 }
 public String getDescripcion() {
   return this.descripcion;
 }
 public void setDescripcion(String descripcion) {
   this.descripcion = descripcion;
 }
}
```

Métodos de instancia, funciones que se ejecutan en el contexto del objeto. Con la palabra clave "***this***" se accede al estado (valores de los atributos) de un objeto en particular. Por ejemplo, el método prod1.getNombre() y prod2.getNombre() retornan el valor correspondiente a su instancia.





Clase

Argentina programa 4.0

- Comportamiento y atributos (no valores) compartidos por un conjunto de objetos. Los atributos son de una clase o tipo primitivo determinado
- Elementos de Instancia:
    - VAtributos: tienen un valor distinto por Objeto
    - Métodos: código a ejecutar, tiene acceso a variables de instancia
- De Clase o Estáticos

VAtributos: compartida por todos las instancias de una clase

- Constantes: Variable de clase que no se puede cambiar en tiempo de ejecución, hay que recompilar.
- Métodos: código a ejecutar, no está asociado con una instancia en particular
- Por ejemplo:

```
LocalTime.now()
```

```
Integer.parseInt("1234")
```

```
Math.abs(-37)
```

Objeto (Instancia)

Argentina programa 4.0

- Un objeto es una instancia de una determinada Clase
- Todo objeto se crea a través de un constructor. Con Java, el mismo se llama utilizando la palabra reservada new. Si no se declara ninguno, por defecto tienen un constructor vacío. Si se declara uno con parámetros, el constructor vacío debe declararse explícitamente.

```
public class Persona {

public Persona(String nombre, LocalDateTime fechaNacimiento) {

this.nombre = nombre;

this.fechaNacimiento = fechaNacimiento;

}

public Persona() { }

Persona unaPersona = new Persona("Mariana",  LocalDateTime.parse("1998-05-04T00:00:00"));
```

- Para utilizar un método de un objeto, decimos que al mismo se le envía un “mensaje”. Para representar esto, se pone un punto “.” luego de una variable referenciando un objeto:

```
unaPersona.edad()
```

Clase - Visibilidad / Alcance

Argentina programa 4.0

Público: atributos o métodos pueden ser utilizados por instancias de cualquier clase. (`public`)

Protegido: atributos o métodos pueden ser utilizados sólo por instancias de la misma clase o sus subclases. (`protected`)

Se utiliza con herencia

Privado: atributos o métodos pueden ser utilizados sólo por instancias de la misma clase. (`private`)

```
private String nombre;
public String getNombre()
```

Alcance o *scope*

Wrappers

Argentina programa 4.0

- Todo tipo primitivo tiene su Wrapper

```
Integer unNumero = 1;
Double otroNumero = 1.0;
Character unCaracter = 'X';
Boolean unBooleano = true;
```

- Esto nos permite utilizar métodos sobre dichos tipos, por ejemplo:

```
int unNumero2 = 1;
unNumero = unNumero + unNumero2; //compatible
otroNumero.equals(unNumero) // false
unBooleano.toString(); // "true"
```

- Y también nos provee de varios métodos útiles

```
Integer.parseInt("123"); // entero 123
Integer.toHexString(13); // "b5" string de número en hexadecimal
```

Argentina
programa
4.0

Gracias!