

Control de Lectura 2

Robert Celi.

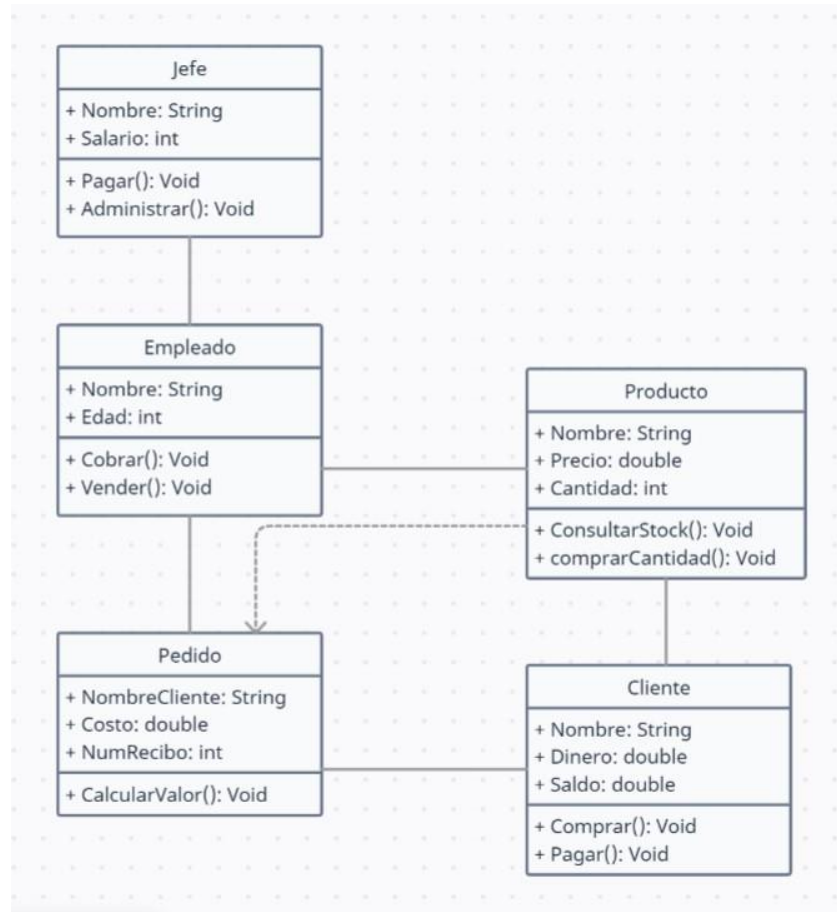
Universidad de las Fuerzas Armadas ESPE

NRC 1323: Programación Orientada a Objetos

08 de diciembre de 2024

Control de Lectura 2

Primero establecemos un diagrama UML, donde a partir de este creamos 5 objetos y establecemos las relaciones que estos van a tener en relación a cada uno de los demás objetos o clases. En este ejemplo hacemos el de un negocio el cual tiene un jefe, un empleado, un cliente, un producto y un pedido



Declaramos cada clase y también declaramos cada uno de los atributos al igual cada uno de las acciones o métodos a realizar cada uno de ellos.

Las correspondientes relaciones son las siguientes:

- Jefe tiene relación de asociación con Empleado
- Empleado tiene relación de asociación con Producto y Pedido
- Producto tiene relación de dependencia de Pedido
- Cliente tiene relación de asociación con Producto y Pedido

Las cuales se pueden observar en el gráfico, siendo las líneas punteadas la que representa la relación de dependencia y en cambio, siendo las líneas continuas las cuales representan la relación de asociación.

A continuación, muestro como queda estructurado el código. En esta primera imagen muestro como esta formado la clase principal llamada Negocio la cual alberga todos los objetos y métodos

```

1  package clases;
2  // Clase principal
3  public class Negocio {
4  public static void main(String[] args) {
5      // Creamos instancias de los objetos
6      Jefe jefe = new Jefe( nombre: "Carlos", salario: 5000);
7      Empleado empleado = new Empleado( nombre: "Juan", edad: 18);
8      Pedido pedido = new Pedido( nombreCliente: "Marco", costo: 350.57, numRecibo: 56981);
9      Producto producto = new Producto( nombre: "Procesador", precio: 350.57, cantidad: 10);
10     Cliente cliente = new Cliente( nombre: "Marco", dinero: 500, saldo: 0);
11
12     // Establecemos relación entre los objetos y vemos el resultado de los métodos
13     jefe.pagar(empleado);
14     jefe.administrar();
15
16     empleado.cobrar(pedido);
17     empleado.vender(producto);
18
19     pedido.calcularValor();
20
21     producto.consultarStock();
22     producto.comprarCantidad(pedido);
23
24     cliente.comprar(producto);
25     cliente.pagar(pedido);
26 }
27 }

```

En esta imagen creamos el código principal donde creamos instancias para cada uno de los objetos y terminamos con establecer los métodos y las relaciones entre los objetos.

En este caso se puede observar que primero se debe de crear un nuevo objeto en base al objeto principal, en este caso con la siguiente línea: “Jefe jefe = new Jefe()” donde dependiendo de las variables que se hayan puesto en base al objeto se le va dar diferentes atributos, en este caso “Jefe jefe = new Jefe(“Carlos”,5000)”.

Estos atributos vienen de las variables que se crean dentro de cada uno de los objetos, después de los gráficos explicare todo esto.

Por ultima parte instanciamos cada uno de los métodos disponibles dentro de cada objeto para comprobar sus resultados y si son correctos.

Ahora procedo a mostrar uno a uno cada una de los objetos y sus métodos

```

1 package clases;
2
3 // Objeto clase Jefe
4 public class Jefe { 2 usages
5     public String nombre; 3 usages
6     public int salario; 2 usages
7
8     public Jefe(String nombre, int salario){ 1 usage
9         this.nombre = nombre;
10        this.salario = salario;
11    }
12    //Opciones a realizar
13 @ public void pagar(Empleado empleado) { 1 usage
14     System.out.println("El jefe " + nombre + " está pagando a " + empleado.nombre);
15 }
16 public void administrar() { 1 usage
17     System.out.println("El jefe " + nombre + " está administrando su salario de "
18         + salario + "$");
19 }
20 }
21
22

```

```

1 package clases;
2
3 // Objeto clase Empleado
4 public class Empleado { 3 usages
5     public String nombre; 4 usages
6     public int edad; 1 usage
7
8     public Empleado(String nombre, int edad){ 1 usage
9         this.nombre = nombre;
10        this.edad = edad;
11    }
12    //Opciones a realizar
13 @ public void cobrar(Pedido pedido) { 1 usage
14     System.out.println("El empleado " + nombre + " está cobrando " + pedido.costo + "$ del pedido de "
15         + pedido.nombreCliente);
16 }
17 @ public void vender(Producto producto) { 1 usage
18     System.out.println("El empleado " + nombre + " está vendiendo " + producto.nombre + " a un precio " +
19         "de " + producto.precio + "$");
20 }
21 }
22

```

```

1 package clases;
2
3 // Objeto clase Producto
4 public class Producto { 4 usages
5     public String nombre; 4 usages
6     public double precio; 2 usages
7     public int cantidad; 2 usages
8
9     public Producto(String nombre, double precio, int cantidad){ 1 usage
10        this.nombre = nombre;
11        this.precio = precio;
12        this.cantidad = cantidad;
13    }
14    //Opciones a realizar
15 public void consultarStock() { 1 usage
16     System.out.println("Consultando el stock del producto, en total hay " + cantidad
17         + " " + nombre);
18 }
19 @ public void comprarCantidad(Pedido pedido) { 1 usage
20     System.out.println("Comprando cantidad del producto del pedido número: " + pedido.numPedido);
21 }
22 }
23

```

```

1 package clases;
2
3 // Objeto clase Cliente
4 public class Cliente { 2 usages
5     public String nombre; 3 usages
6     public double dinero; 2 usages
7     public double saldo; 2 usages
8
9     public Cliente(String nombre, int dinero, int saldo){ 1 usage
10         this.nombre = nombre;
11         this.dinero = dinero;
12         this.saldo = saldo;
13     }
14
15     //Acciones a realizar
16     @ public void comprar(Producto producto) { 1 usage
17         System.out.println("El cliente " + nombre + " está comprando " + producto.nombre + ", pagando " +
18             "con un total de " + dinero + "$");
19     }
20
21     @ public void pagar(Pedido pedido) { 1 usage
22         System.out.println("El cliente " + nombre + " está pagando " + pedido.costo + "$ , " +
23             " quedando con un saldo pendiente de: " + saldo + "$");
24     }
25 }

```

```

1 package clases;
2
3 // Objeto clase Pedido
4 public class Pedido { 5 usages
5     public String nombreCliente; 3 usages
6     public double costo; 4 usages
7     public int numRecibo; 3 usages
8
9     public Pedido(String nombreCliente, double costo, int numRecibo){ 1 usage
10         this.nombreCliente = nombreCliente;
11         this.costo = costo;
12         this.numRecibo = numRecibo;
13     }
14
15     //Acciones a realizar
16     public void calcularValor() { 1 usage
17         System.out.println("Calculando el valor del pedido de " + nombreCliente + ", cuyo valor "
18             + "es " + costo + "$ con el numero de recibo " + numRecibo);
19     }
20 }

```

Explicuemos esta última parte del código, ya que el comportamiento de estos objetos es muy similar. Primero tenemos “public class Pedido”, donde creamos un objeto publico para que pueda ser modificado por otros.

Luego está una lista de atributos, donde se comienza con “public String nombreCliente”, donde declaramos primero el modificador de acceso, luego declaramos de que tipo va ser la variable en este caso “String”, para luego declarar la variable.

En este caso se usa “public Pedido(String nombreCliente, double costo, int numRecibo”, donde primero se da el modificador de acceso al objeto, luego el nombre del objeto, luego se

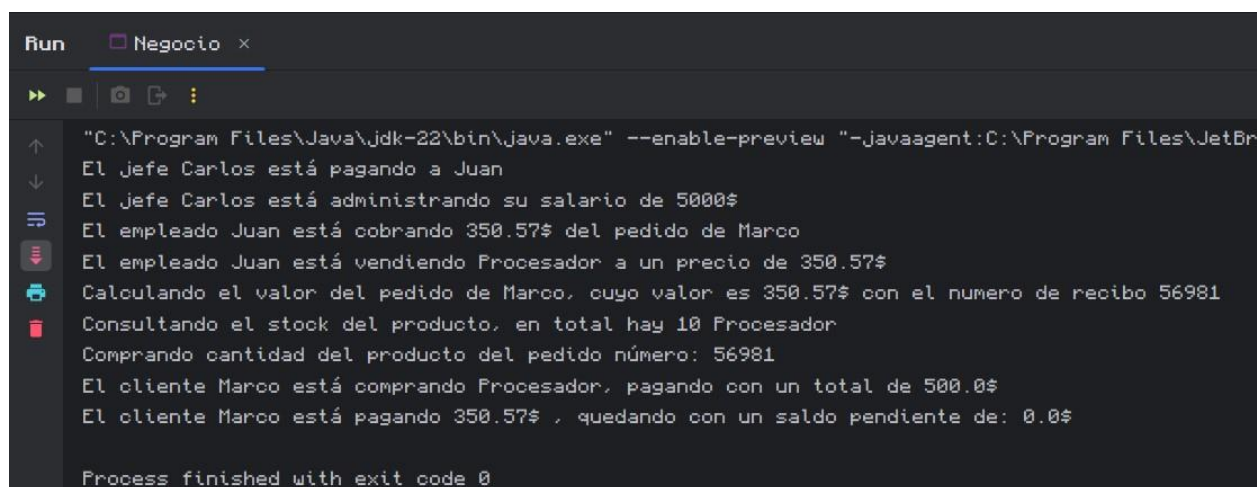
declara de que manera se le puede dar atributos al objeto, en este caso primero iría un “String” o cadena de texto, luego un “int” o numero entero y un “double” o número decimal.

Dependiendo del objeto a este se le va asignar los diferentes atributos los cuales van a ser seleccionados por el usuario en el código principal. De manera que estos objetos son creados con el objetivo de ser instanciados mas tarde en el código principal, para de igual manera poder relacionar con otros objetos.

Por ultimo tenemos “public void calcularValor() {
 System.out.println("Calculando el valor del pedido de " + nombreCliente + ", cuyo valor "
 + "es " + costo + "\$ con el numero de recibo " + numRecibo);”. esta línea de código es el método del objeto.

En este caso el método de nombre “calcularValor” se va representar mediante una línea de comando en la consola, por el siguiente comando “System.out.println”, además de eso también se usa las variables del objeto como “nombreCliente, costo, numRecibo”, donde dependiendo del valor que se les de se representan de una forma u otra.

Ahora vamos a comprobar el resultado en la terminal



```

Run  Negocio x
>> ■ | 📷 🔍 ⋮
↑
↓
📄
📌
🖨
🗑
"C:\Program Files\Java\jdk-22\bin\java.exe" --enable-preview "-javaagent:C:\Program Files\JetBr
El jefe Carlos está pagando a Juan
El jefe Carlos está administrando su salario de 5000$
El empleado Juan está cobrando 350.57$ del pedido de Marco
El empleado Juan está vendiendo Procesador a un precio de 350.57$
Calculando el valor del pedido de Marco, cuyo valor es 350.57$ con el numero de recibo 56981
Consultando el stock del producto, en total hay 10 Procesador
Comprando cantidad del producto del pedido número: 56981
El cliente Marco está comprando Procesador, pagando con un total de 500.0$
El cliente Marco está pagando 350.57$ , quedando con un saldo pendiente de: 0.0$
Process finished with exit code 0
  
```

Observando que cada uno de los métodos se demuestra con éxito además de demostrar sus relaciones con los demás objetos.

- Donde en el primer caso el Jefe realiza la acción de pagar al empleado, donde también este administra su salario.
- El Empleado esta vendiendo el Producto y cobrando el Pedido
- El Pedido calcula su valor
- El Producto calcula su stock y cuantos hay, de igual manera se compra la cantidad de Producto que viene a partir del Pedido
- El Cliente realiza el Pedido y termina pagando el precio final del Pedido

A continuación, inserto el código desarrollado uno a uno:

```
package clases; // Clase principal public
class Negocio {    public static void
main(String[] args) {        // Creamos
instancias de los objetos

    Jefe jefe = new Jefe("Carlos",5000);

    Empleado empleado = new Empleado("Juan",18);

    Pedido pedido = new Pedido("Marco",350.57,56981);

    Producto producto = new Producto("Procesador",350.57,10);

    Cliente cliente = new Cliente("Marco",500,0);

    // Establecemos relación entre los objetos y vemos el resultado de los métodos
jefe.pagar(empleado);    jefe.administrar();
```



```
    empleado.cobrar(pedido);
```

```
empleado.vender(producto);
```

```
    pedido.calcularValor();
```

```
    producto.consultarStock();
```

```
producto.comprarCantidad(pedido);
```

```
    cliente.comprar(producto);
```

```
cliente.pagar(pedido);
```

```
    }
```

```
} package
```

```
clases;
```

```
// Objeto clase Jefe
```

```
public class Jefe {
```

```
    public String nombre;
```

```
    public int salario;
```

```
    public Jefe(String nombre, int salario){        this.nombre = nombre;        this.salario = salario;
```

```
    }
```

```
    //Acciones a realizar    public void
```

```
pagar(Empleado empleado) {
```

```

        System.out.println("El jefe " + nombre + " está pagando a "+ empleado.nombre);
    }

    public void administrar() {

        System.out.println("El jefe " + nombre + " está administrando su salario de "
            + salario + "$");

    }

} package

clases;

// Objeto clase Empleado

public class Empleado {

    public String nombre;

    public int edad;

    public Empleado(String nombre, int edad){

        this.nombre = nombre;      this.edad = edad;

    }

    //Acciones a realizar    public void

    cobrar(Pedido pedido) {

        System.out.println("El empleado " + nombre + " está cobrando " + pedido.costos + "$ del

        pedido de "

            + pedido.nombreCliente);

    }

```

```

public void vender(Producto producto) {

    System.out.println("El empleado " + nombre + " está vendiendo " + producto.nombre + " a
un precio " +

        "de " + producto.precio + "$");

}
} package

```

clases;

// Objeto clase Pedido public

```

class Pedido {    public String
nombreCliente;    public
double costo;    public int
numRecibo;

```

```

    public Pedido(String nombreCliente, double costo, int numRecibo){
this.nombreCliente = nombreCliente;    this.costo = costo;

    this.numRecibo = numRecibo;

}

```

//Acciones a realizar

```

public void calcularValor() {

    System.out.println("Calculando el valor del pedido de " + nombreCliente + ", cuyo valor "

```

```

        + "es " + costo + "$ con el numero de recibo " + numRecibo);
    }
} package

clases;

// Objeto clase Producto

public class Producto {

    public String nombre;

    public double precio;

    public int cantidad;

    public Producto(String nombre,double precio,int cantidad){

        this.nombre = nombre;        this.precio = precio;

        this.cantidad = cantidad;

    }

    //Acciones a realizar

    public void consultarStock() {

        System.out.println("Consultando el stock del producto, en total hay " + cantidad

            + " " + nombre);

    }

    public void comprarCantidad(Pedido pedido) {

        System.out.println("Comprando cantidad del producto del pedido número: " +

            pedido.numRecibo);
    }
}

```

```

    }
} package

clases;

// Objeto clase Cliente

public class Cliente {

    public String nombre;

    public double dinero;

    public double saldo;

    public Cliente(String nombre, int dinero, int saldo){

        this.nombre = nombre;      this.dinero = dinero;

        this.saldo = saldo;

    }

    //Acciones a realizar

    public void comprar(Producto producto) {

        System.out.println("El cliente " + nombre + " está comprando " + producto.nombre + ",
pagando " +

            "con un total de " + dinero + "$");

    }

    public void pagar(Pedido pedido) {

        System.out.println("El cliente " + nombre + " está pagando " + pedido.costos + "$ , " +

```

" quedando con un saldo pendiente de: " + saldo + "\$");

}

}