

Recognizing Objects

The screenshot shows a Google Meet window with a presentation titled "Recognizing Objects". The presentation content includes a list of objects and their attributes, and a UML class diagram.

Objects:

- 1- Shoe: OK a. idShoe: int, size: String, brand: String, price: float, color: String, shoeType: ShoeType
- 2- Inventory: OK a. listShoe: Shoe[], place: String, dateInventory: Date, startItem: int, endItem: int, idShoeItem: int, idShoeItem: int, idShoeItem: int
- 3- Order: OK a. orderShoe: Shoe, dateOrder: Date, idOrder: int
- 4- Store: OK a. shoe: Shoe, order: Order
- 5- List: ? a. listShoe: Shoe[]
- 6- Archive: OK a. listShoe: Shoe[]
- 7- ManagerStore: X b. writeFile, readFile, modifyFile
- 8- Payment: OK a. typeOfPayment: TypeOfPayment (cash, credit card, debit card, PayPal)
- 9- BranchOffice: X a. string main street, number, crossing street
- 10- Seller: OK a. string name
- 11- Shelving: X
- 12- Credit: X
- 13- Client: OK a. int id, string Name, paymentMethod
- 14- Warehouse: X
- 15- Bill: X
- 16- Classification: X
- 17- Devolution: X

UML Class Diagram:

```
classDiagram
    class Shoe {
        +idShoe: int
        +size: String
        +brand: String
        +price: float
        +color: String
        +shoeType: ShoeType
    }
    class Inventory {
        +listShoe: Shoe[]
        +place: String
        +dateInventory: Date
        +startItem: int
        +endItem: int
        +idShoeItem: int
        +idShoeItem: int
        +idShoeItem: int
    }
    class Order {
        +listShoe: Shoe[]
        +date: Date
        +idOrder: int
    }
    class Store {
        +shoe: Shoe
        +order: Order
    }
    class List {
        +listShoe: Shoe[]
    }
    class Archive {
        +listShoe: Shoe[]
    }
    class ManagerStore {
        +writeFile()
        +readFile()
        +modifyFile()
    }
    class Payment {
        +typeOfPayment: TypeOfPayment
    }
    class BranchOffice {
        +string main street, number, crossing street
    }
    class Seller {
        +string name
    }
    class Shelving {
    }
    class Credit {
    }
    class Client {
        +int id
        +string Name
        +paymentMethod
    }
    class Warehouse {
    }
    class Bill {
        +int id
        +string client
        +listShoe: Shoe
        +date: Date
        +totalPrice: float
    }
    class Classification {
        +idShoe: int
        +size: float
        +brand: String
        +price: float
        +color: String
        +shoeType: ShoeType
    }
    class Devolution {
        +listShoe: Shoe
        +dateInventory: Date
    }
    Shoe "1" -- "1" Inventory
    Shoe "1" -- "1" Order
    Shoe "1" -- "1" Store
    Shoe "1" -- "1" List
    Shoe "1" -- "1" Archive
    Shoe "1" -- "1" ManagerStore
    Shoe "1" -- "1" Payment
    Shoe "1" -- "1" BranchOffice
    Shoe "1" -- "1" Seller
    Shoe "1" -- "1" Shelving
    Shoe "1" -- "1" Credit
    Shoe "1" -- "1" Client
    Shoe "1" -- "1" Warehouse
    Shoe "1" -- "1" Bill
    Shoe "1" -- "1" Classification
    Shoe "1" -- "1" Devolution
```

Modeling class diagram

The screenshot shows a Google Meet window with a presentation titled "Modeling class diagram". The presentation content includes a UML class diagram.

UML Class Diagram:

```
classDiagram
    class Shoe {
        +idShoe: int
        +size: String
        +brand: String
        +price: float
        +color: String
        +shoeType: ShoeType
    }
    class Inventory {
        +listShoe: Shoe[]
        +place: String
        +dateInventory: Date
        +startItem: int
        +endItem: int
        +idShoeItem: int
        +idShoeItem: int
        +idShoeItem: int
    }
    class Order {
        +listShoe: Shoe[]
        +date: Date
        +idOrder: int
    }
    class Store {
        +shoe: Shoe
        +order: Order
    }
    class List {
        +listShoe: Shoe[]
    }
    class Archive {
        +listShoe: Shoe[]
    }
    class ManagerStore {
        +writeFile()
        +readFile()
        +modifyFile()
    }
    class Payment {
        +typeOfPayment: TypeOfPayment
    }
    class BranchOffice {
        +string main street, number, crossing street
    }
    class Seller {
        +string name
    }
    class Shelving {
    }
    class Credit {
    }
    class Client {
        +int id
        +string Name
        +paymentMethod
    }
    class Warehouse {
    }
    class Bill {
        +int id
        +string client
        +listShoe: Shoe
        +date: Date
        +totalPrice: float
    }
    class Classification {
        +idShoe: int
        +size: float
        +brand: String
        +price: float
        +color: String
        +shoeType: ShoeType
    }
    class Devolution {
        +listShoe: Shoe
        +dateInventory: Date
    }
    Shoe "1" -- "1" Inventory
    Shoe "1" -- "1" Order
    Shoe "1" -- "1" Store
    Shoe "1" -- "1" List
    Shoe "1" -- "1" Archive
    Shoe "1" -- "1" ManagerStore
    Shoe "1" -- "1" Payment
    Shoe "1" -- "1" BranchOffice
    Shoe "1" -- "1" Seller
    Shoe "1" -- "1" Shelving
    Shoe "1" -- "1" Credit
    Shoe "1" -- "1" Client
    Shoe "1" -- "1" Warehouse
    Shoe "1" -- "1" Bill
    Shoe "1" -- "1" Classification
    Shoe "1" -- "1" Devolution
```

Salió con x Sistema x https://e x En Defen x Meet x Asociati x Recongu x + -

meet.google.com/mur-gpbv-tvt?pli=1&authuser=1

Aplicaciones Google Facebook WhatsApp Mi ESPE GitHub Otros marcadores

DARIAN MICHAEL MARTINEZ ALFONSO está presentando

Detalles de la reunión

Personas (7) Chat

JOHAO ALEJANDRO MORALES...

DARIAN MICHAEL MARTINEZ ...

DARIAN MICHAEL MARTINEZ ...

EDNAN JOSUE MERINO CALD...

LEONCIO ELIAN MORAN MOR...

LESLEY ALEXANDRA MOSCOSO...

Detalles de l... ^

DARIAN MICHAEL MARTINEZ ... está presentando

Escribe aquí para buscar

23:01 14/06/2020

Modeling Use Case Diagram

LEONCIO ELIAN MORAN MORENO está presentando

untitled - Visual Paradigm Community Edition (not for commercial use)

Diagram Navigator

Figure 1

Diagram Navigator

Cursor

Use Case

Association

Actor

System

Include

Extend

Dependency

Generalization

Collaboration

Element Group

Note

Anchor

Constraint

Containment

Allocate

Rationale

Problem

Package

Diagram Overview

Generic Connector

Client

LiquorStoreSystem

select product

choose type of drink

select drink brand

create order

pay order

extension points

pay with cash

pay with card

credit card

debit card

cancel order

modify order

LEONCIO ELIAN MORA...

EDNAN JOSUE MERIN...

LEONCIO ELIAN MORA...

EDNAN JOSUE MERIN...

JOHAO ALEJANDRO M...

LESLEY ALEXANDRA MO...

Activar Windows

Ve a Configuración para activar Windows.

11:49 10/6/2020

