

## 2. Programación Orientada a Objetos (POO)

### 2.3. Interacción de objetos

#### Programación Modular y Orientada a Objetos

Felipe Ibañez y Juan Miguel Lopez

[felipe.anfurrutia@ehu.es](mailto:felipe.anfurrutia@ehu.es)

[juanmiguel.lopez@ehu.es](mailto:juanmiguel.lopez@ehu.es)

Dpto. de Lenguajes y Sistemas Informáticos  
UPV/EHU

## Contenido

- ▣ Interacción de objetos
  - Abstracción
  - Modularización
- ▣ Diagramas de secuencia UML

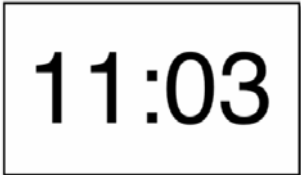
## Interacción de objetos

---

- ▣ Los objetos no son entes **individuales**  
... **cooperan**  
para llevar a cabo una tarea común
- ▣ Normalmente un programa no tiene  
objetos de una sola clase

## Ejemplo: un reloj digital

---



11:03

## Abstracción y Modularización

### □ Abstracción

- habilidad para **ignorar/prescindir de los detalles** de las partes
- para centrar la atención en un nivel más alto de un problema
  - Ejemplo: mapas

### □ Modularización

- proceso de **dividir un todo en partes** bien definidas que pueden ser construidas y examinadas separadamente,
- las cuales interactúan de maneras bien definidas
  - *Divide et vinces*

La **abstracción** permite ver el bosque y la **modularización** los árboles que hacen el bosque

## Modularización del visor del reloj

11:03

¿Un visor de cuatro dígitos?

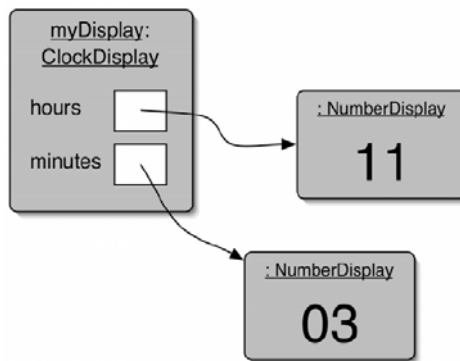
¿O dos visores de dos dígitos?

11

03

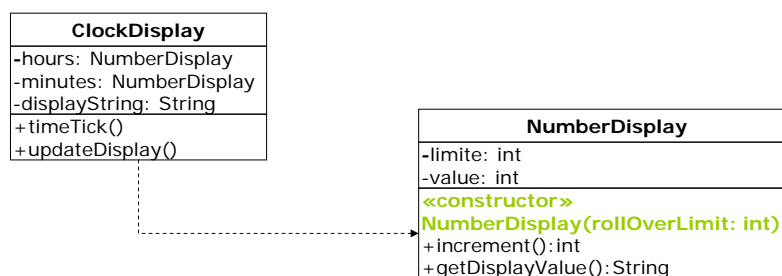
## Diagrama de objetos

- Describe los objetos reales (instancias)



## Diagrama de clases

- Modela los objetos del mundo real



## Implementación: NumberDisplay

```
public class NumberDisplay{
    private int limit;
    private int value;

    public NumberDisplay(int rollOverLimit) {
        limit = rollOverLimit;
        value = 0;
    }
    public void increment() {
        value = (value + 1) % limit;
    }
    public String getDisplayValue(){
        if(value < 10) {
            return "0" + value;
        }
        else {
            return "" + value;
        }
    }
}
```

El máximo número  
que se visualizará

Gestiona el estado  
interno del objeto

Observar que el estado  
interno es un entero, pero  
lo que se visualizará será  
un String

## Implementación: ClockDisplay

```
public class ClockDisplay{
    private NumberDisplay hours;
    private NumberDisplay minutes;
    private String displayString;

    public ClockDisplay() {
        hours = new NumberDisplay(24);
        minutes = new NumberDisplay(60);
        updateDisplay();
    }
    public void timeTick() {
        minutes.increment();
        if(minutes.getValue() == 0) {
            // it just rolled over!
            hours.increment();
        }
        updateDisplay();
    }
}
```

Un objeto que se  
construye creando  
otros objetos

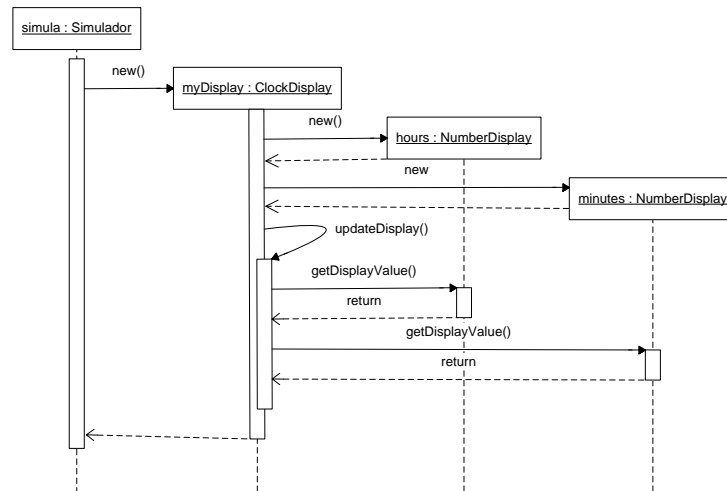
Pasa mensajes a los  
objetos que ha construido

Método privado

```
/**
 * Update the internal string that
 * represents the display.
 */
private void updateDisplay() {
    displayString =
        hours.getDisplayValue() + ":" +
        minutes.getDisplayValue();
}
```

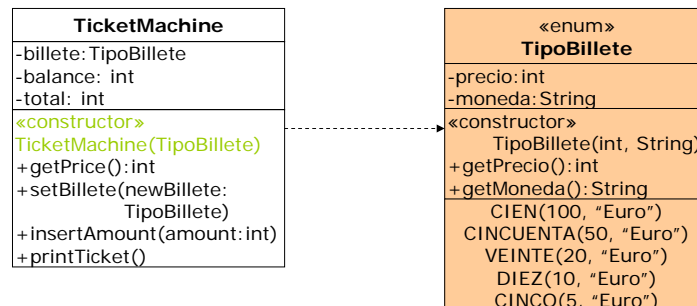
## Diagrama de secuencia UML

- Modela la interacción de objetos



## Ejercicio

- Mejorando la maquina expendedora de dinero (TicketMachine):
  - Establecer los posibles billetes definiendo un tipo de dato (TipoBillete)



- Escribe el código del método getPrice del TicketMachine
- Escribe un programa que instancie una maquina con billetes de 10 € y devuelva el precio de los billetes de la maquina

## Ejercicio

---

- ▣ Simular el cambio de resolución de la pantalla del ordenador
  - un ordenador tiene una única pantalla, además del nombre del fabricante
  - el fabricante establece una resolución por defecto (800x600) a la pantalla y un refresco de 60GHz
  - desde el ordenador se puede modificar bien la resolución o el refresco: diciendole el número de pixeles horizontales y verticales; ó el valor de refresco, respectivamente
  - la resolución de la pantalla se describe mediante el número de pixeles horizontales y verticales, además del valor de refresco
  - la pantalla calcula el número de pixeles
- ▣ Escribe un programa que instancie un ordenador y cambie su resolución a 1024x768 y el valor de refresco a 75Ghz

## Ejercicio

---

- ▣ Mejora la definición anterior:
  - describiendo 3 valores permitidos para la resolución de pantalla
  - y otras 3 para el de refresco