Programación III

Clase V

Andrés Fortier

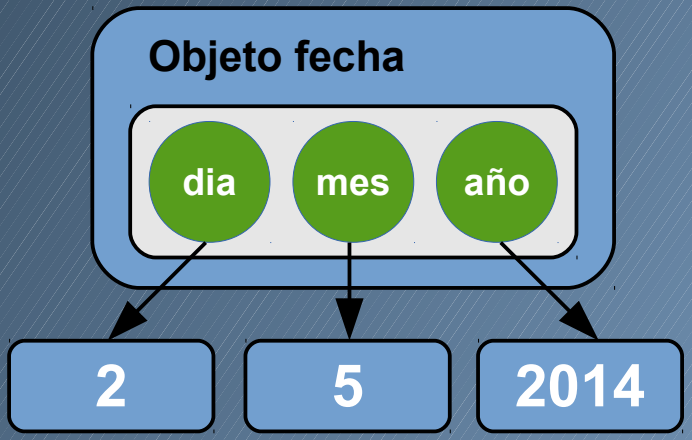
Importante

- La semana que viene no hay clase de Programación.
- Deben cerrar la práctica 1 durante esa semana y comenzar la práctica 2.

Repaso

- Modelar un objeto que represente una fecha
  - `dia / dia: unNumero.`
  - `mes / mes: unNumero.`
  - `año / año: unNumero.`
  - `= otraFecha.`
  - Mensajes de comparación (`>, <, >=, <=`).
  - `entre: unaFecha y: otraFecha.`

Repaso




```
dia: unNumero
	dia := unNumero.
```

```
dia
	^dia.
```

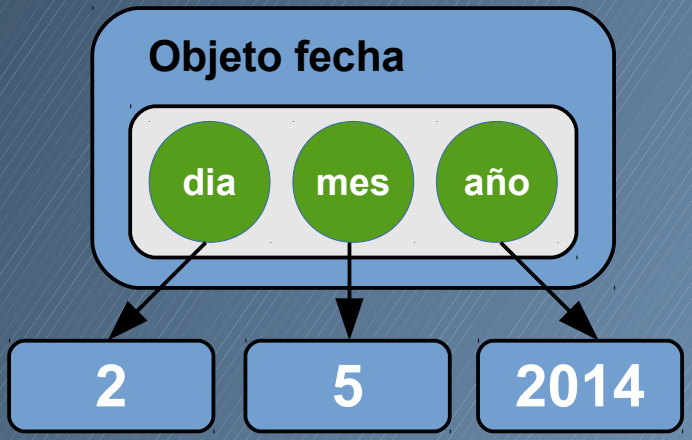
```
mes: unNumero
	mes := unNumero.
```

```
mes
	^mes.
```

```
año: unNumero
	año := unNumero.
```

```
año
	^año.
```

Repaso




```
= otraFecha

	^(dia = otraFecha dia)
		& (mes = otraFecha mes)
		& (año = otraFecha año).
```

```
< otraFecha

	^(self año < otraFecha año)
		| ((self año = otraFecha año) & (self mes < otraFecha mes))
		| ((self año = otraFecha año) & (self mes = otraFecha mes) & (self dia < otraFecha dia)).
```

```
<= otraFecha

	^(self < otraFecha) | (self = otraFecha).
```

Fechas

- ¿Cómo hacemos para tener mas de una fecha?

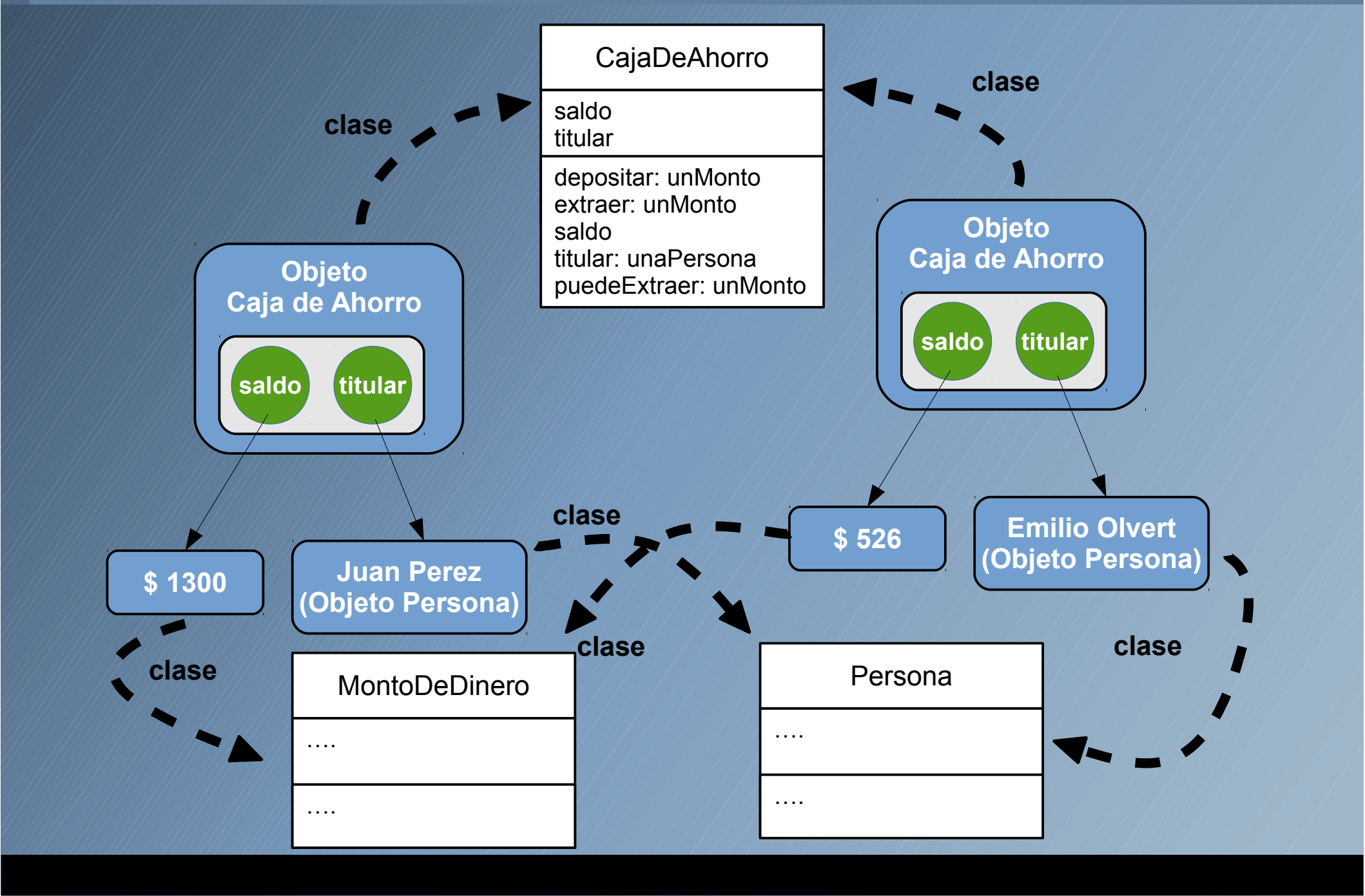
Comportamiento común entre objetos

- Pensemos en un banco: ¿Cuántos objetos Caja de Ahorro habrá?
- ¿Qué cosas son comunes a todas las Cajas de Ahorro y qué cosas son particulares de cada una?
- Entonces… ¿Cómo representamos este comportamiento común, de manera que cada Caja de Ahorro pueda reutilizarlo?

Clases e instancias

- Una clase es una descripción abstracta de un conjunto de objetos.
- Las clases cumplen tres roles:
  - Agrupan el comportamiento común a sus instancias.
  - Definen la “forma” (*shape*) de sus instancias.
  - Crean objetos que son instancia de ellas.
- En consecuencia todas las instancias de una clase se comportan de la misma manera.
- Cada instancia mantendrá su propio estado interno.

Ejemplo de Clases e Instancias

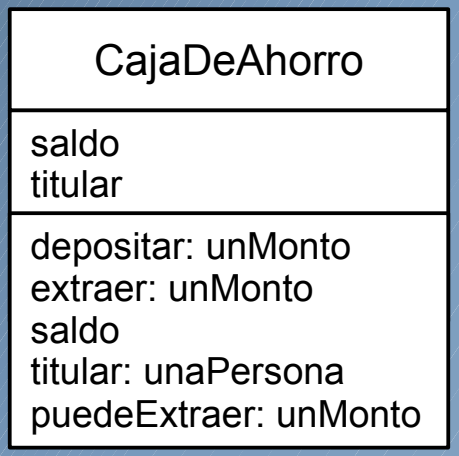


Especificación de clases

- Las clases se especifican por medio de:
  - Un nombre.
  - El estado o estructura interna que tendrán sus instancias.
  - Los mensajes y métodos asociados que definen el comportamiento.

Especificación de clases

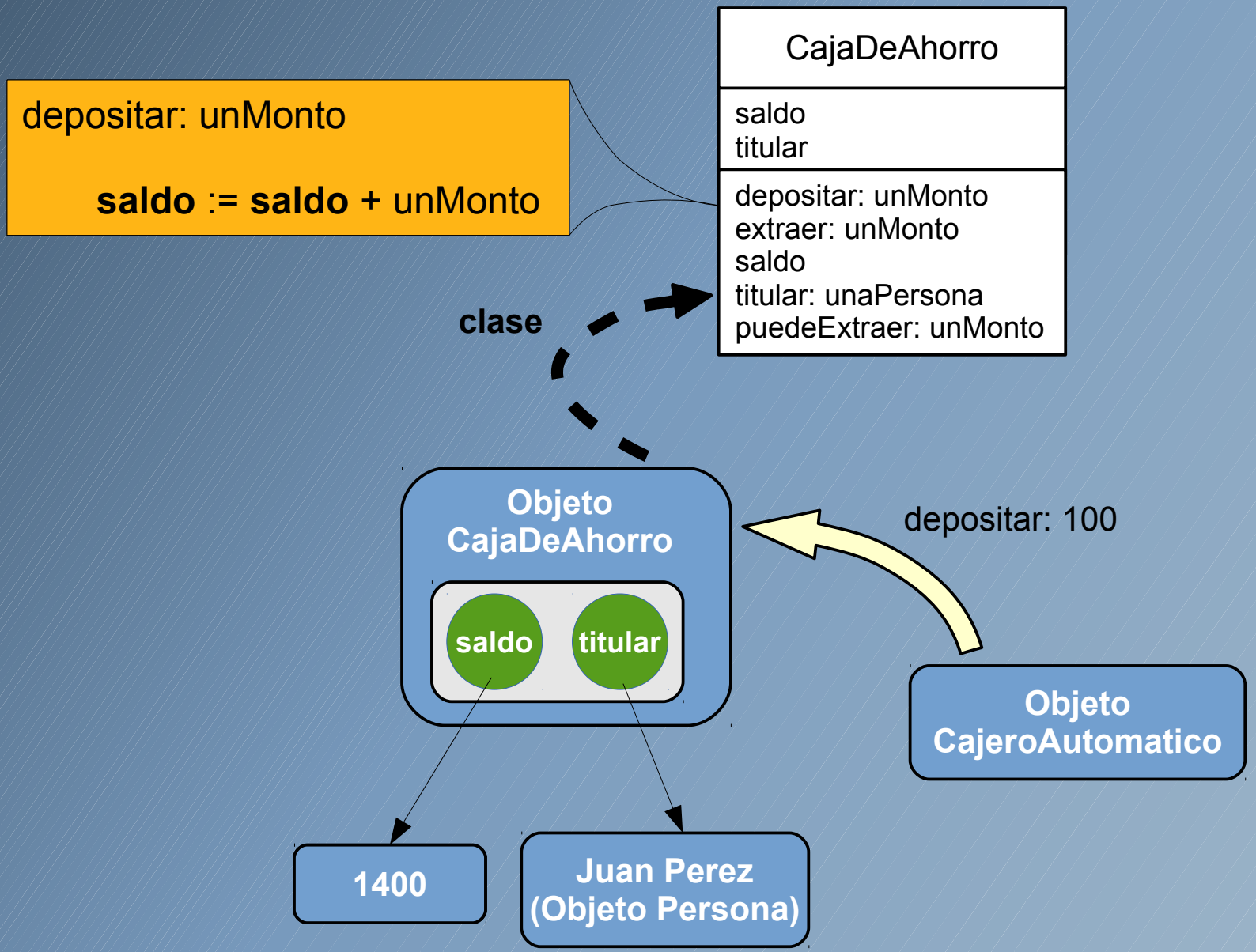
Variables de instancia: Comienzan con minúscula y no pueden contener espacios.



Nombre de la clase: Comienza con mayúscula y no puede tener espacios.

Protocolo: Por cada mensaje se debe especificar el nombre y los parámteros.

Envío de mensajes con clases

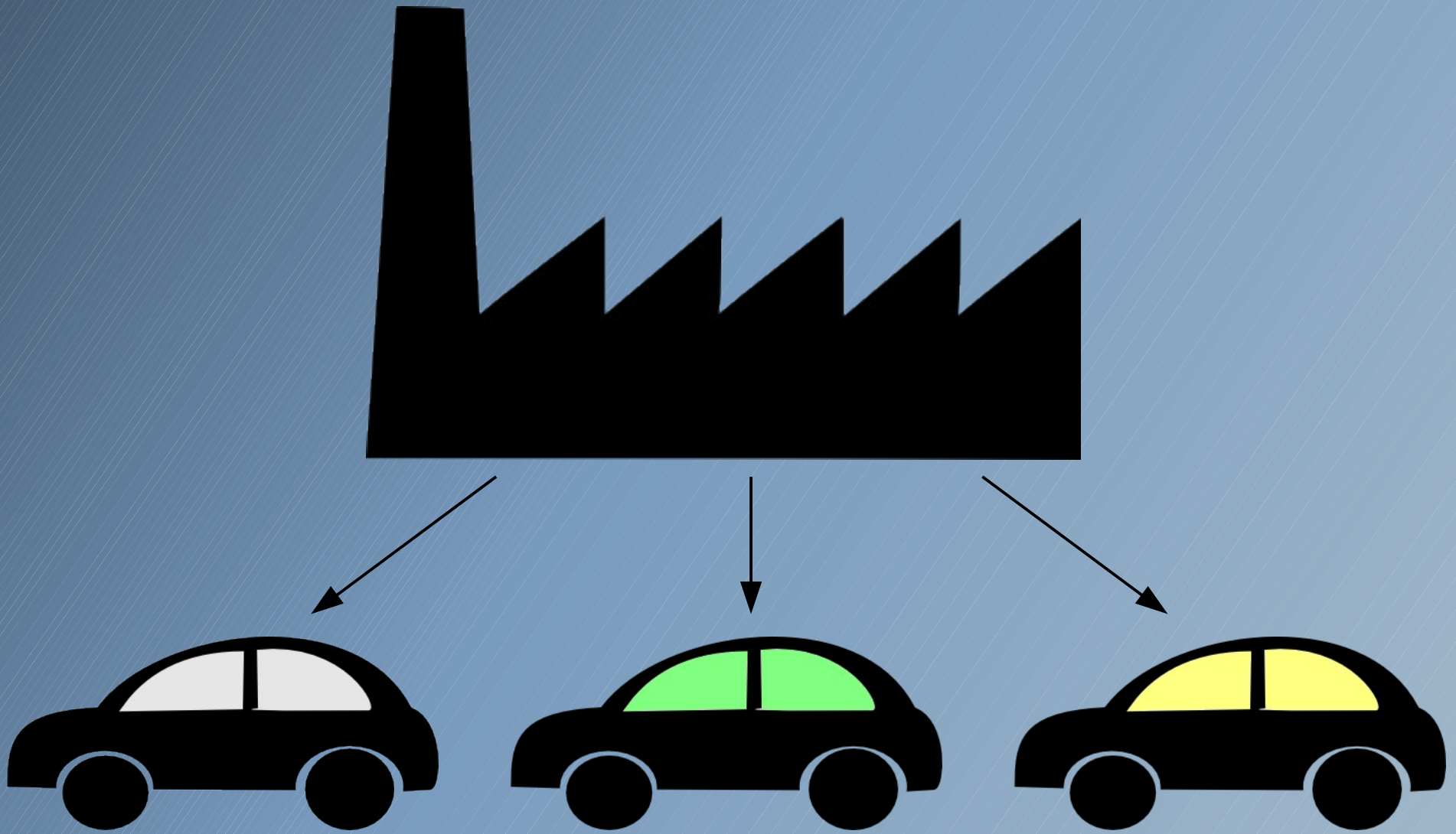


Clases

- Ya vimos que
  - Las clases agrupan el comportamiento común de sus instancias.
  - Definen la “forma” que tendrán sus instancias (estado interno).
- ¿Cómo creamos nuevos objetos?

Clases como fábricas de objetos

- Proceso de instanciación



Instanciación

- Es el mecanismo de creación de objetos.
- Los objetos se instancian a partir de un molde (la clase).
- Un nuevo objeto es una instancia de una clase.
- Todas las instancias de una misma clase
  - Tendrán la misma estructura interna.
  - Responderán al mismo protocolo.

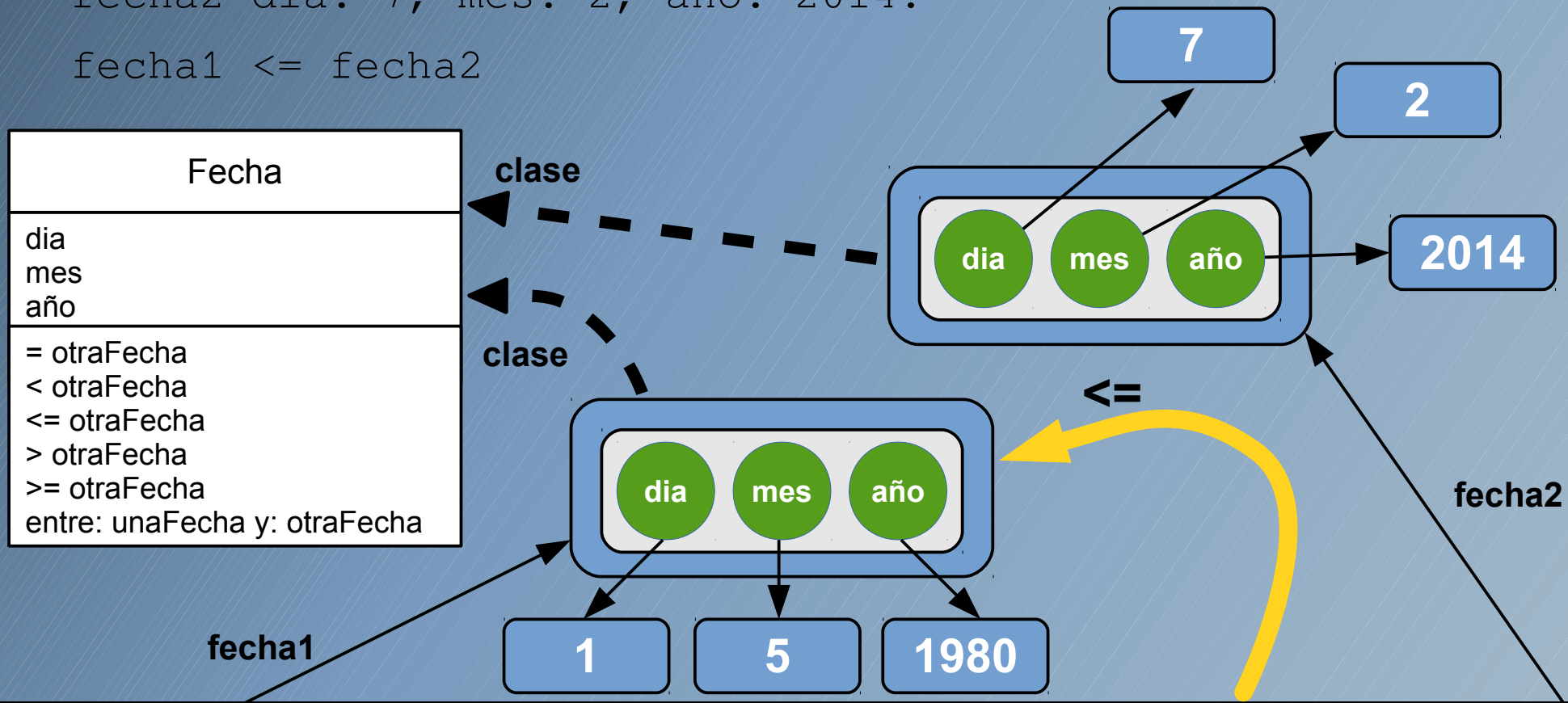
Instanciación: mensaje #new

- Mantenemos el principio de objeto-mensaje.
- Todas las clases entienden el mensaje `#new`.
- Instanciación: `<clase> new.`
- Ejemplo:

```
miCaja := CajaDeAhorro new.
miCaja inicializar.
miCaja depositar: 100.
miCaja saldo.
```

Ejemplo con fechas

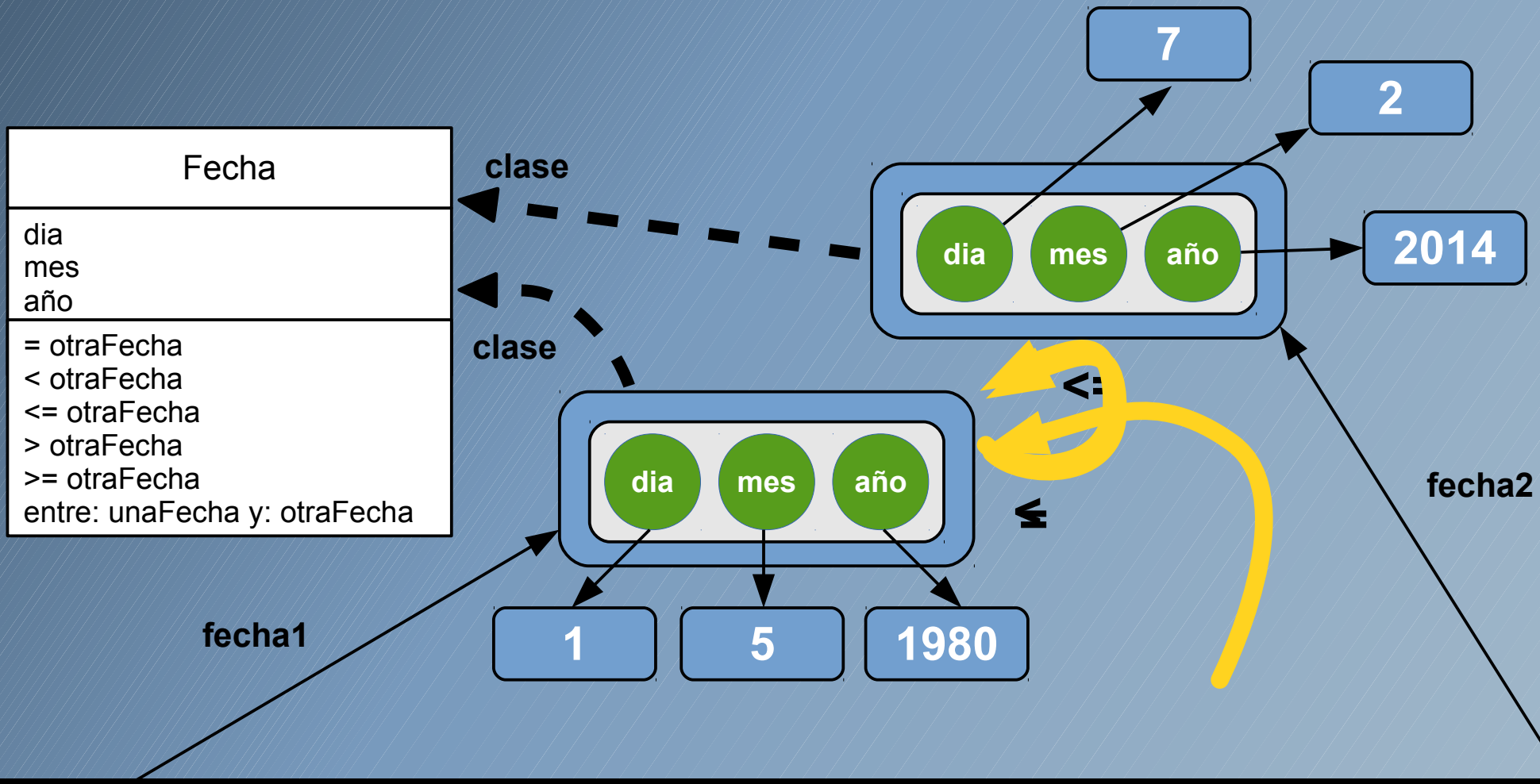
```
| fecha1 fecha2 |
fecha1 = Fecha new.
fecha2 = Fecha new.
fecha1 dia: 1; mes: 5; año: 1980.
fecha2 dia: 7; mes: 2; año: 2014.
fecha1 <= fecha2
```



Ejemplo con fechas

```
<= otraFecha

    ^(self = otraFecha) | (self < otraFecha).
```



Ejercicio

- `CajaDeAhorro new.`
  - Si.
- `CajaDeAhorro depositar: 1000.`
  - No.
-
```
| caja |
caja := CajaDeAhorro new.
caja depositar: 1500.
```
  - Si.

Contador

- Llevemos el objeto contador a la clase Contador.