Programación III

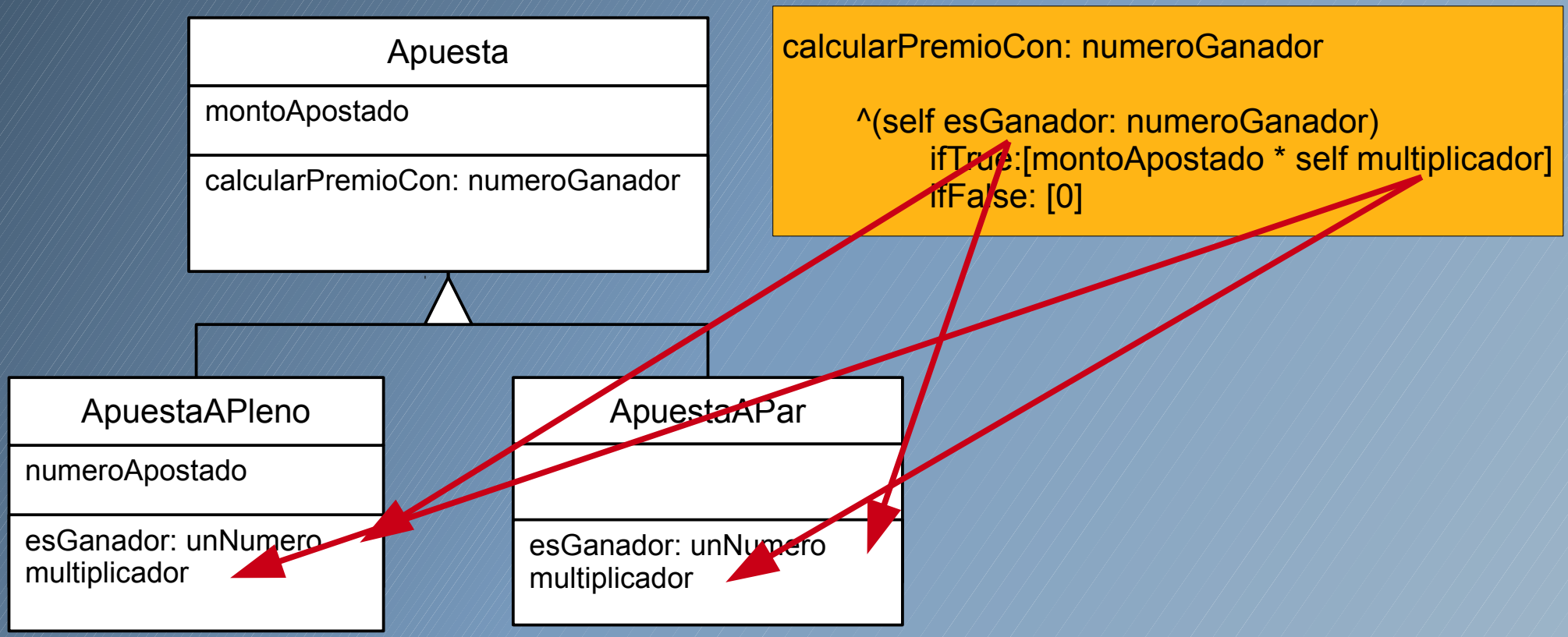
Clase XII

Andrés Fortier

La pregunta de la Rhodesia

- Defina qué es un mensaje abstracto y una clase abstracta e indique para qué se utilizan.

Repaso - ejercicio Ruleta



Repaso - Jerarquías de clases

- Para crear un tipo nuevo de apuesta debemos subclasificar Apuesta.
- ¿Cómo indicamos que todas las subclases deben implementar los mensajes #esGanador: y #multiplicador?.
- ¿Cómo indicamos que Apuesta todavía no tiene todo el comportamiento necesario y que, por ende, no se pueden crear instancias de la misma?

Repaso - Mensajes abstractos

- Indican que una clase debería entender un determinado mensaje, pero no posee la información suficiente para implementarlo.
- Las subclases deben proveer una implementación.
- En UML se denotan con la letra cursiva.
- En Smalltalk se implementan utilizando el mensaje #subclassResponsibility.

Repaso - Clases abstractas

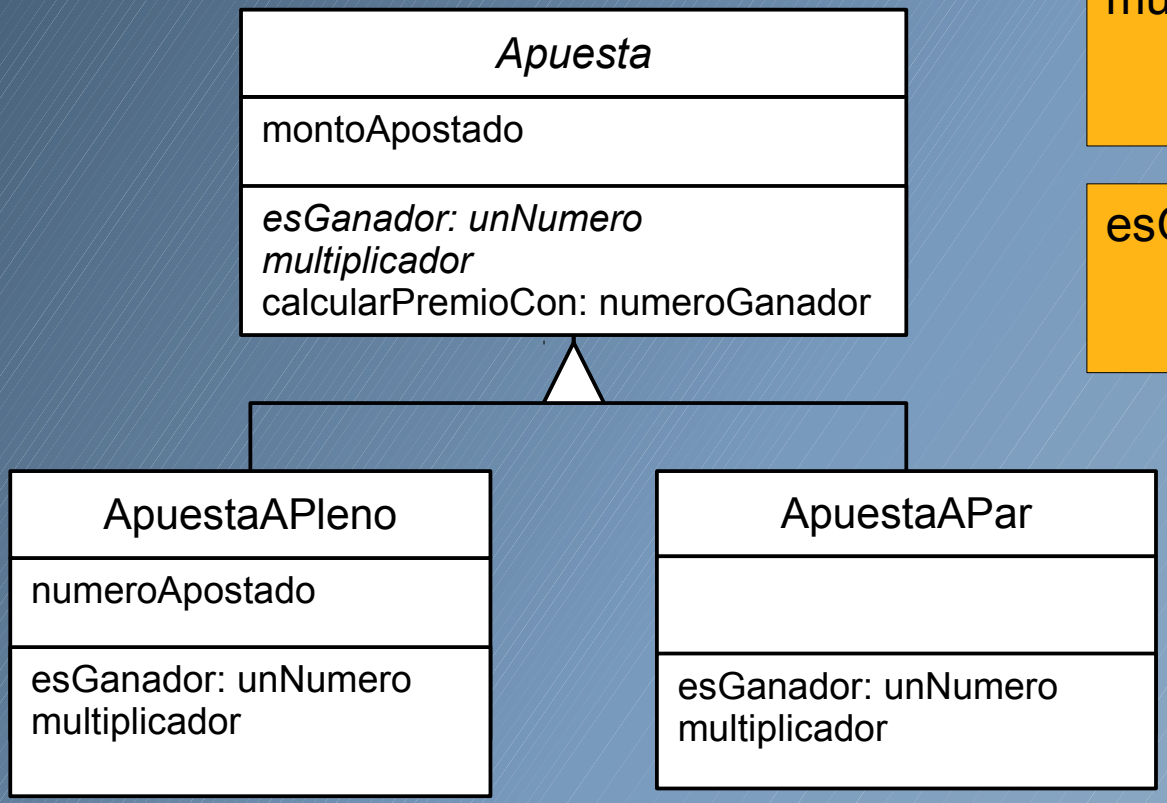
- Son aquellas que tienen al menos un mensaje abstracto.
- En UML se denotan con la letra cursiva.
- No hay forma de indicar esto en Smalltalk.

Repaso - Diagrama de clases

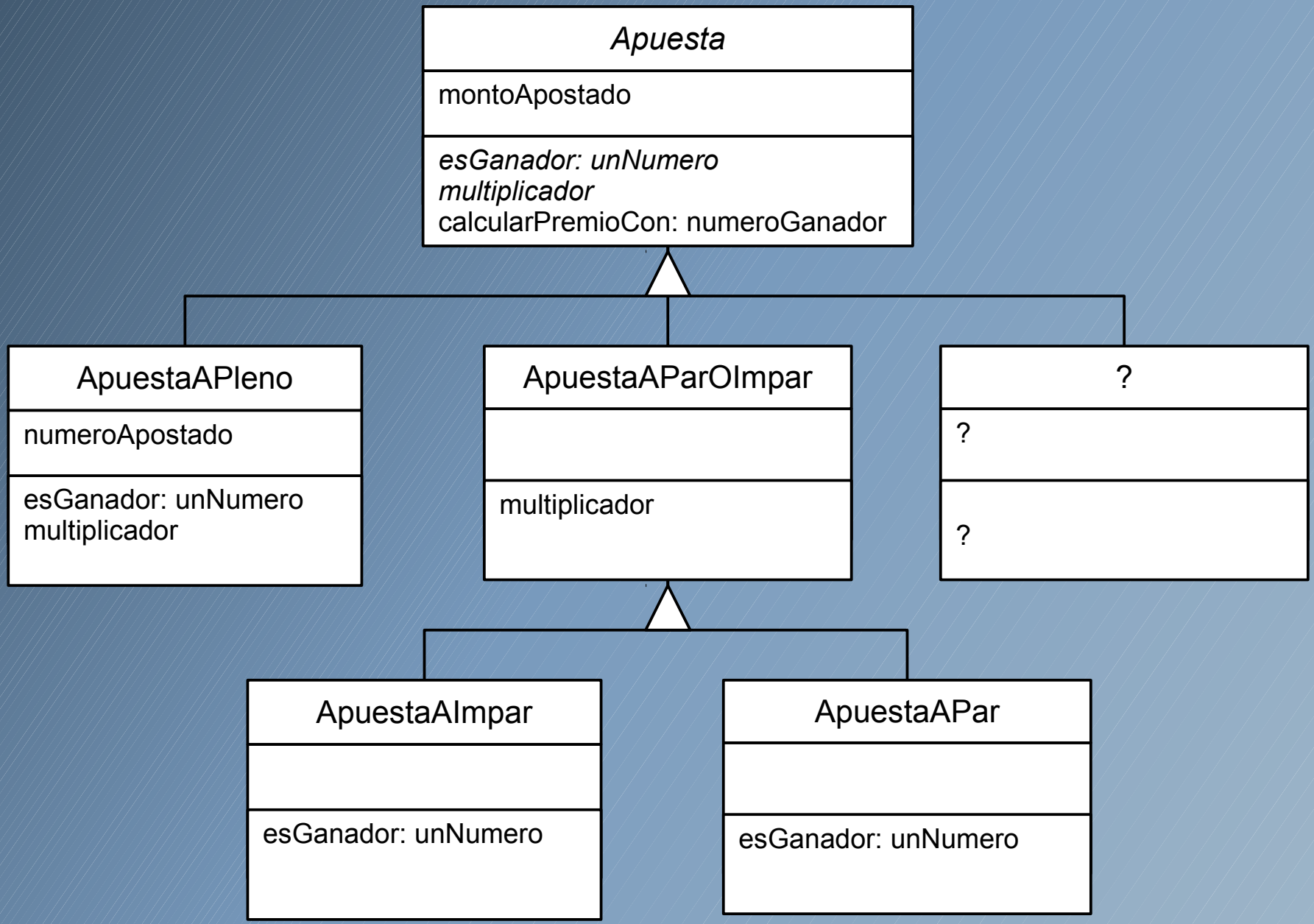
```
calcularPremioCon: numeroGanador

    ^(self esGanador: numeroGanador)
        ifTrue:[montoApostado * self multiplicador]
        ifFalse: [0]
```

```
multiplicador

    ^self subclassResponsibility
```

```
esGanador: unNumero

    ^self subclassResponsibility
```



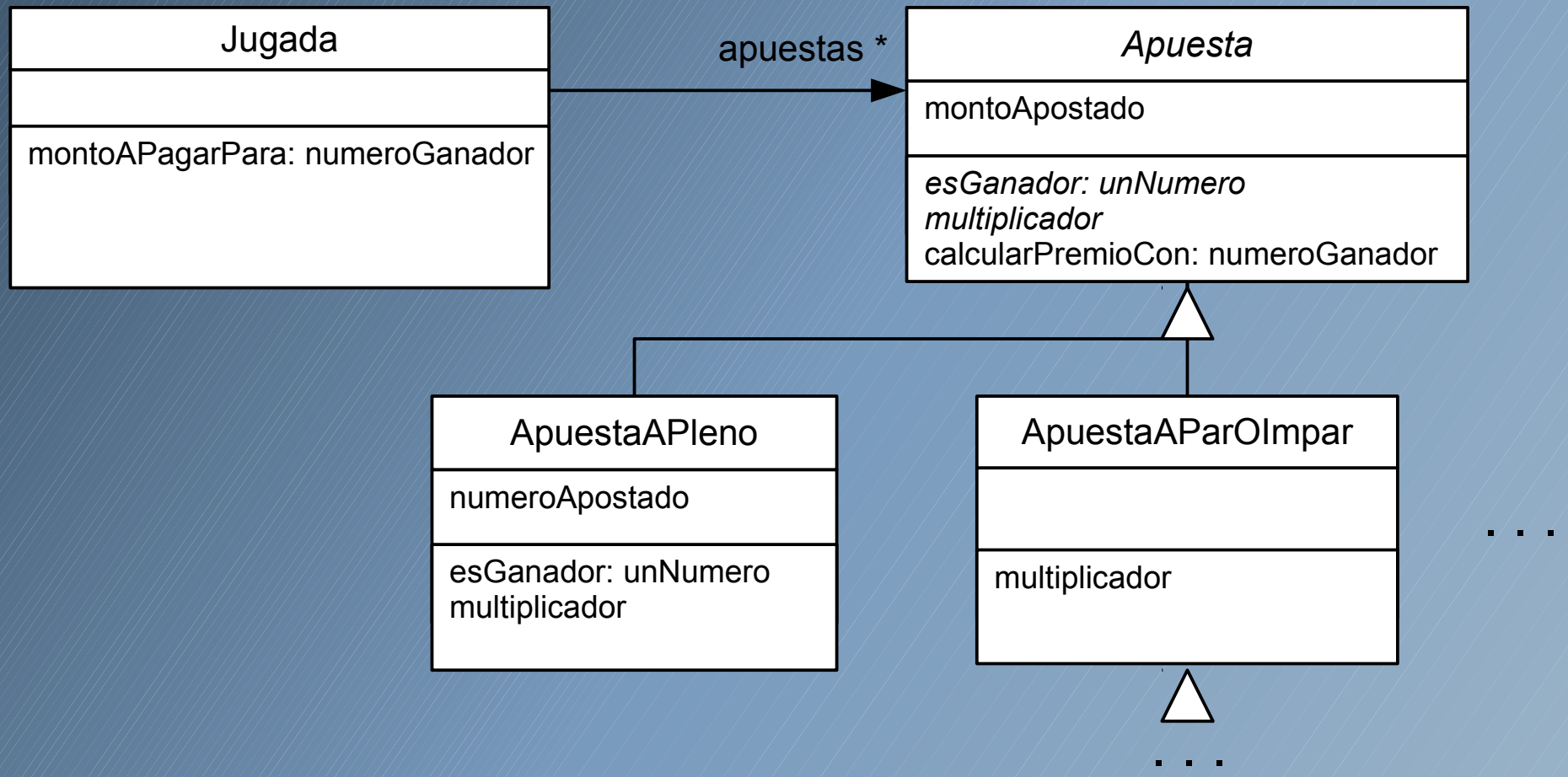
Repaso - Diagrama de clases



Monto total a pagar

- Modelamos el concepto de realizar una apuesta.
- Ahora queremos saber el monto total a pagar para todas las apuestas.

Diagramas de instancia y colaboración




```
montoAPagarPara: numeroGanador

    ^(apuestas collect: [:apuesta | apuesta calcularPremioCon: numeroGanador]) sum.
```