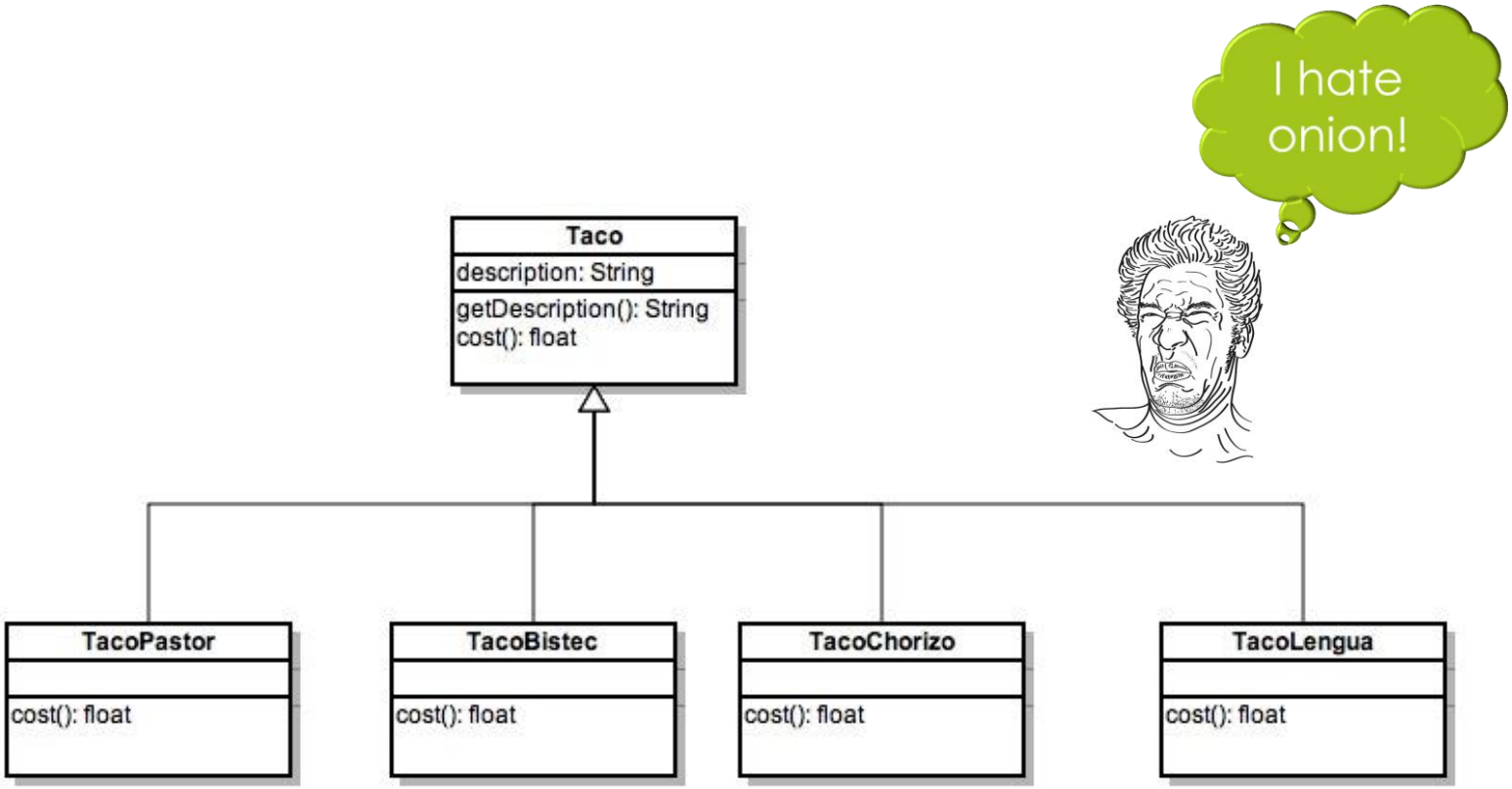
Decorator

Software Design

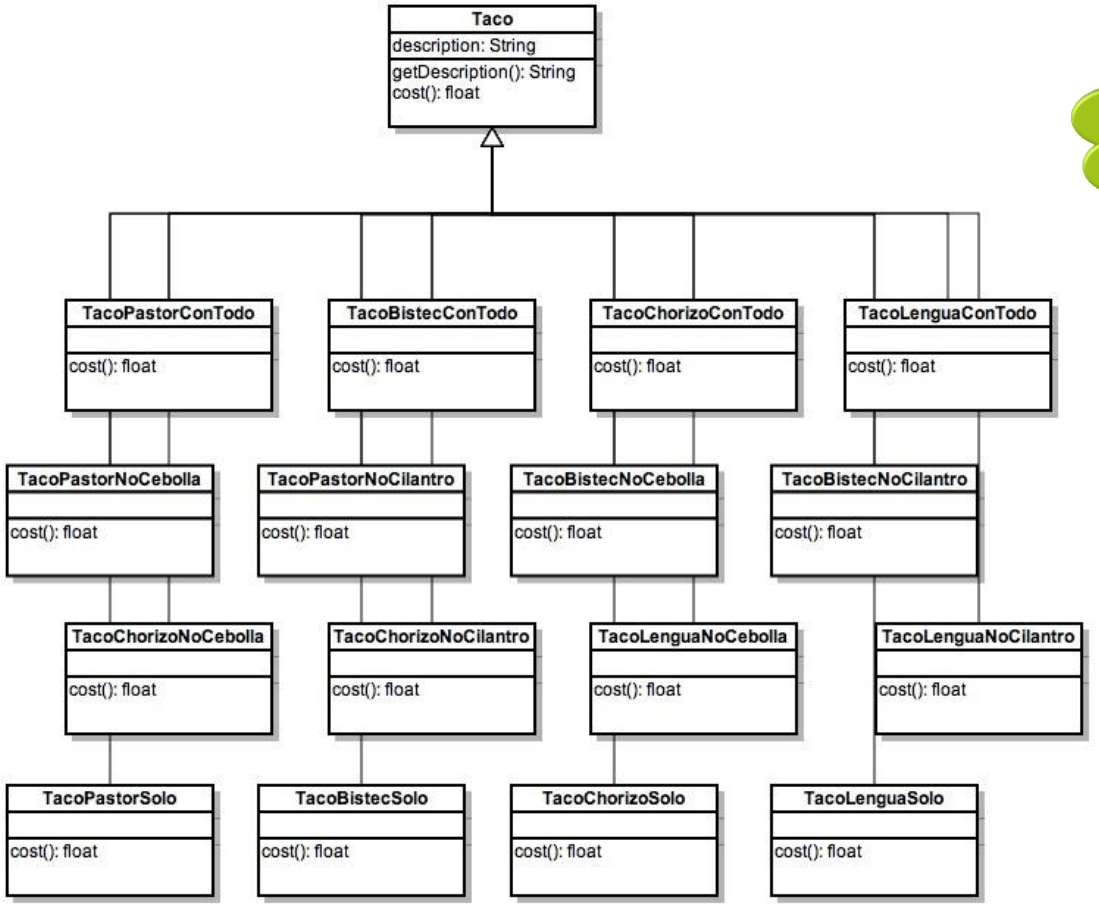
TuTaco V1.0 by Tacosoft

- Tacosoft is a small software company that has developed a point of sales system for Taco restaurants
- TuTaco has released the first version, but they want to improve their design as they have faced some issues.

TuTaco V1.0


I hate onion!
Taco
description: String
getDescription(): String
cost(): float
TacoPastor
cost(): float
TacoBistec
cost(): float
TacoChorizo
cost(): float
TacoLengua
cost(): float

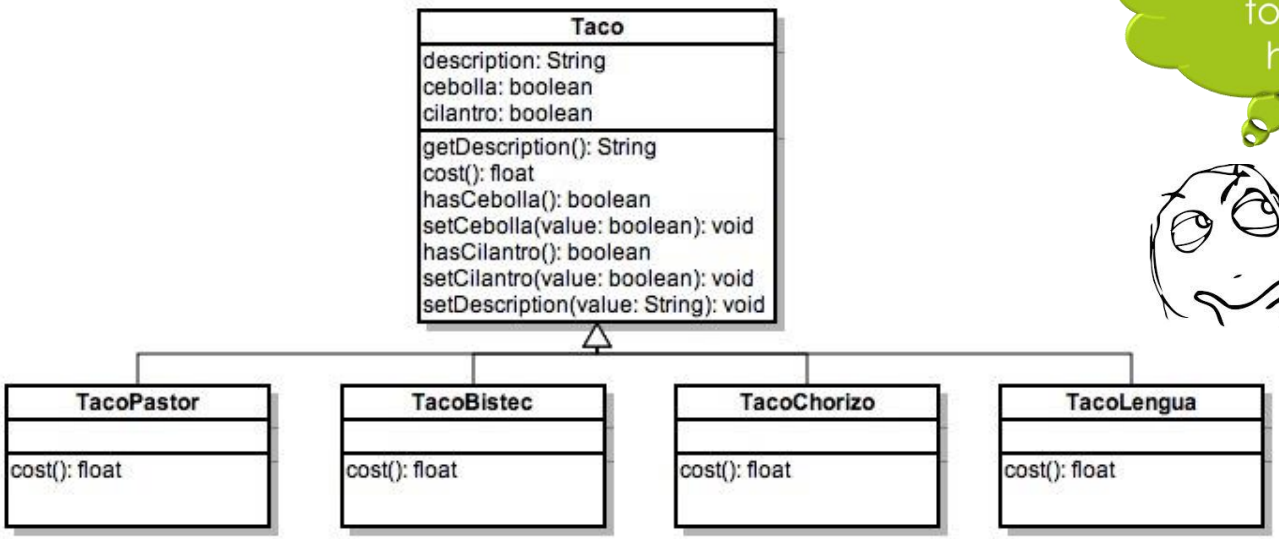
TuTaco V1.1


Taco
description: String
getDescription(): String
cost(): float
TacoPastorConTodo
cost(): float
TacoBistecConTodo
cost(): float
TacoChorizoConTodo
cost(): float
TacoLenguaConTodo
cost(): float
TacoPastorNoCebolla
cost(): float
TacoPastorNoCilantro
cost(): float
TacoBistecNoCebolla
cost(): float
TacoBistecNoCilantro
cost(): float
TacoChorizoNoCebolla
cost(): float
TacoChorizoNoCilantro
cost(): float
TacoLenguaNoCebolla
cost(): float
TacoLenguaNoCilantro
cost(): float
TacoPastorSolo
cost(): float
TacoBistecSolo
cost(): float
TacoChorizoSolo
cost(): float
TacoLenguaSolo
cost(): float


Can you say "Class explosion"?

TuTaco V1.2


I would like a quesadilla in tortilla de harina!


Taco
description: String
cebolla: boolean
cilantro: boolean
getDescription(): String
cost(): float
hasCebolla(): boolean
setCebolla(value: boolean): void
hasCilantro(): boolean
setCilantro(value: boolean): void
setDescription(value: String): void
TacoPastor
cost(): float
TacoBistec
cost(): float
TacoChorizo
cost(): float
TacoLengua
cost(): float

Identify possible changes

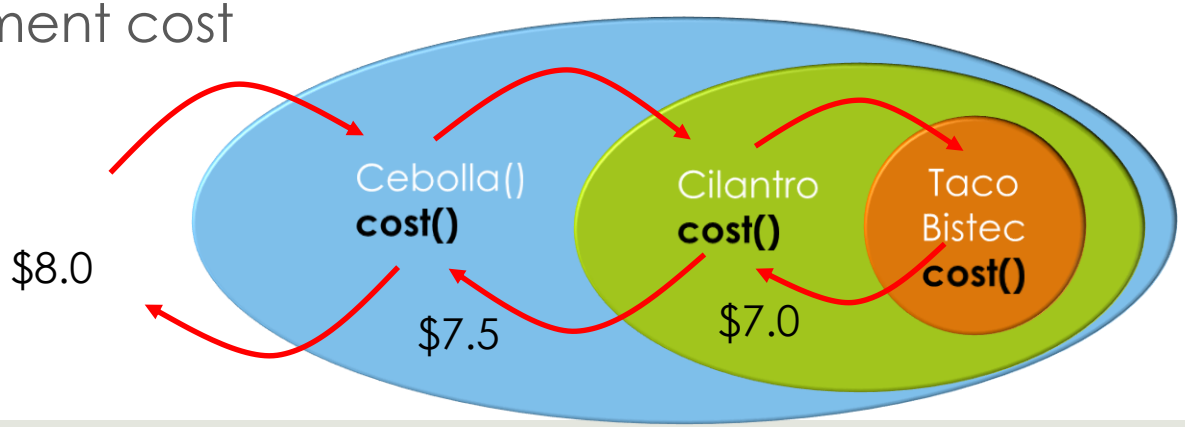
- New condiments: Avocado, cheese, tortilla de maíz, tortilla de harina…
- Some condiments may alter the cost
- New tacos: Tripa, Sesos, Adobada, Pescado…
- Some tacos may have different cost
- What if I only want 1 tortilla

Design Principle

- Classes should be open for extension, but closed for modification:
  - The Open-Closed principle is one of the most important design principles.
  - Allow classes to be easily extended to incorporate new things without modifying existing code

Building a Taco de Bistec con todo!

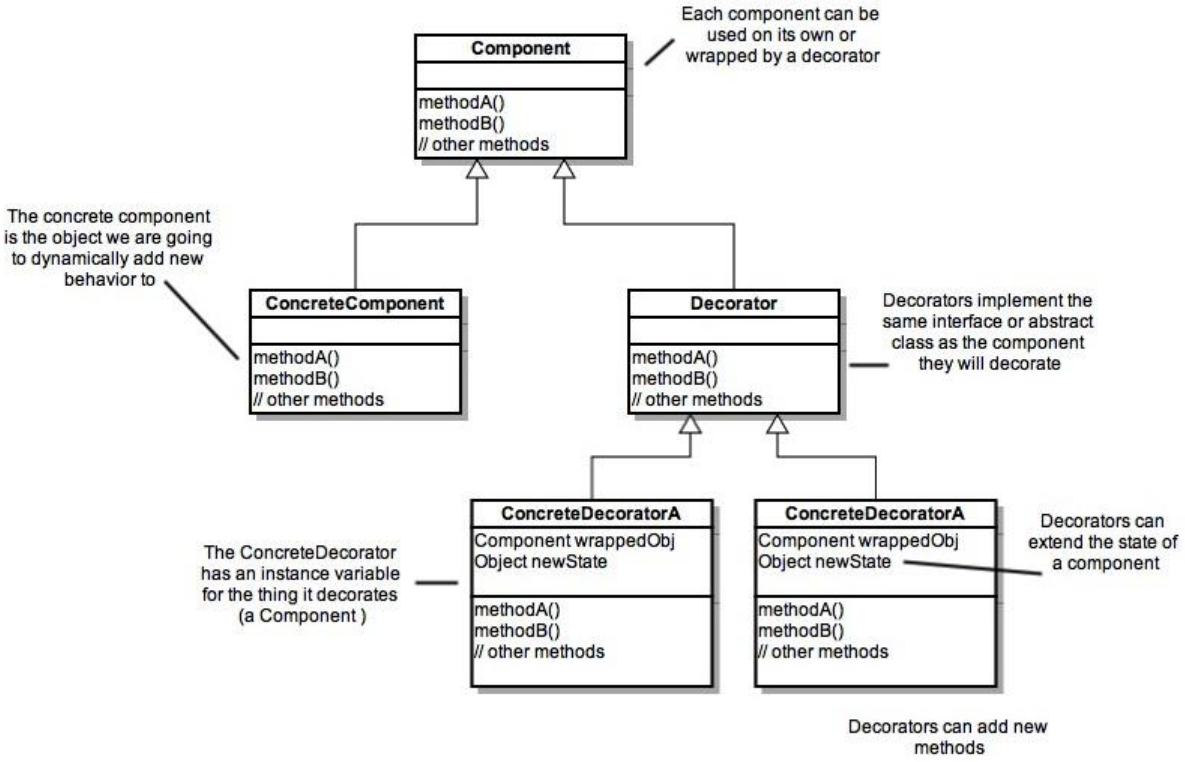
- Take a TacoBistec object
- Decorate it with “Cebolla”
- Decorate it with “Cilantro”
- Call the cost() method and rely in delegation to add the condiment cost



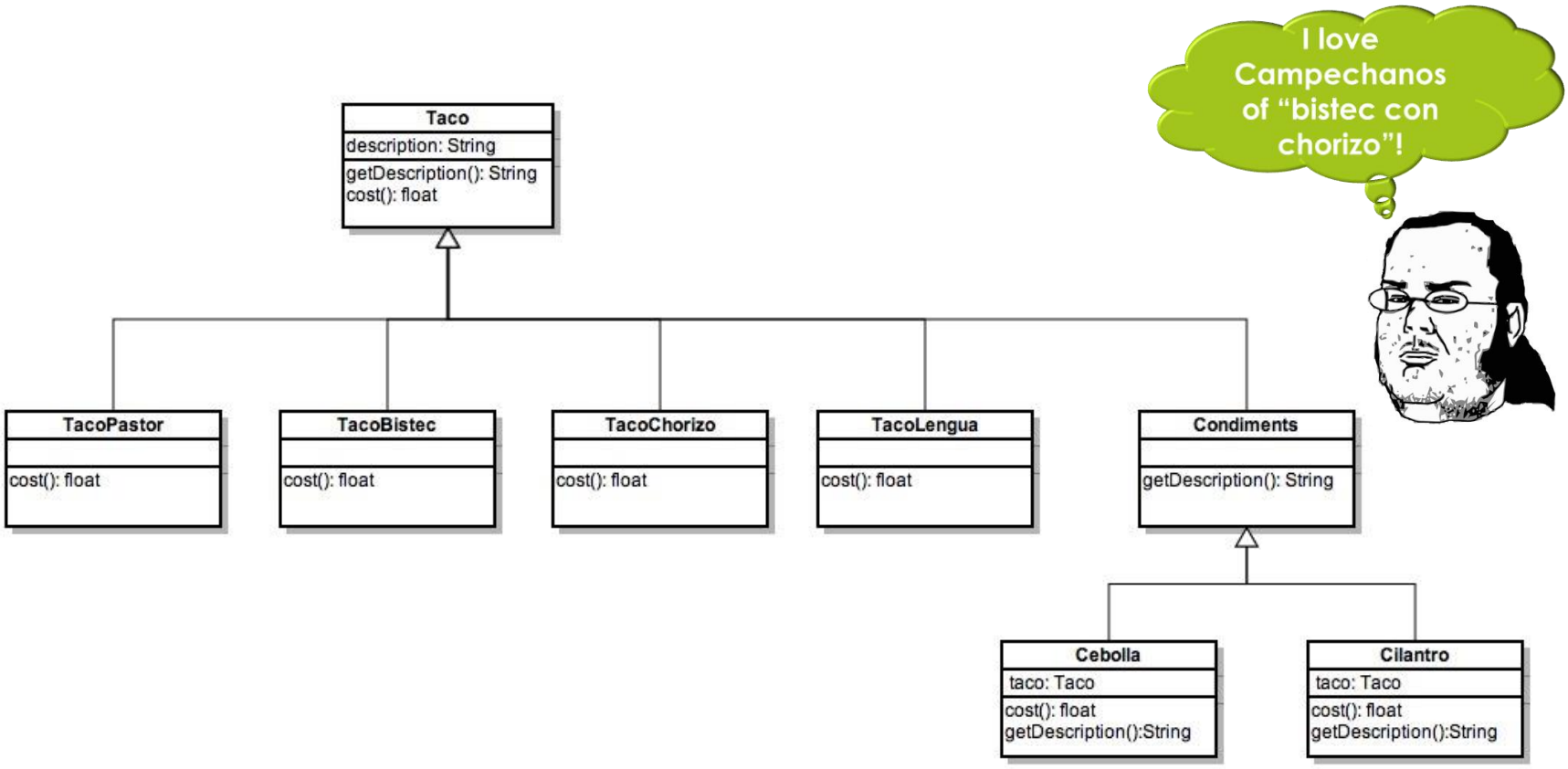
Decorator Pattern

- The decorator pattern attaches additional responsibilities to an object dynamically
- Decorators provide a flexible alternative to subclassing for extending functionality

Decorator class diagram

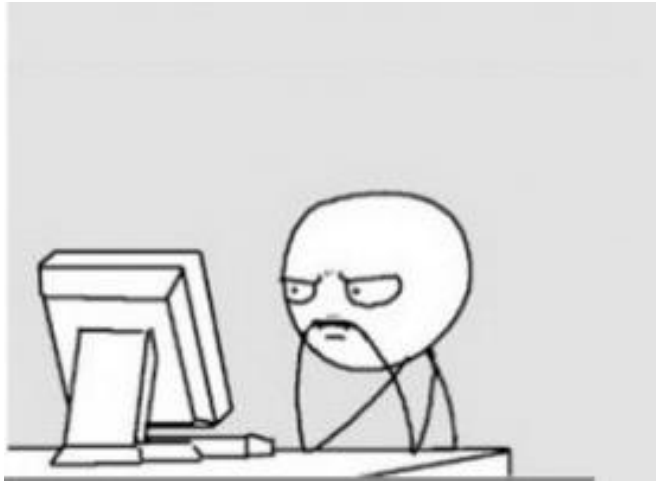


Decorator in TuTaco V1.3

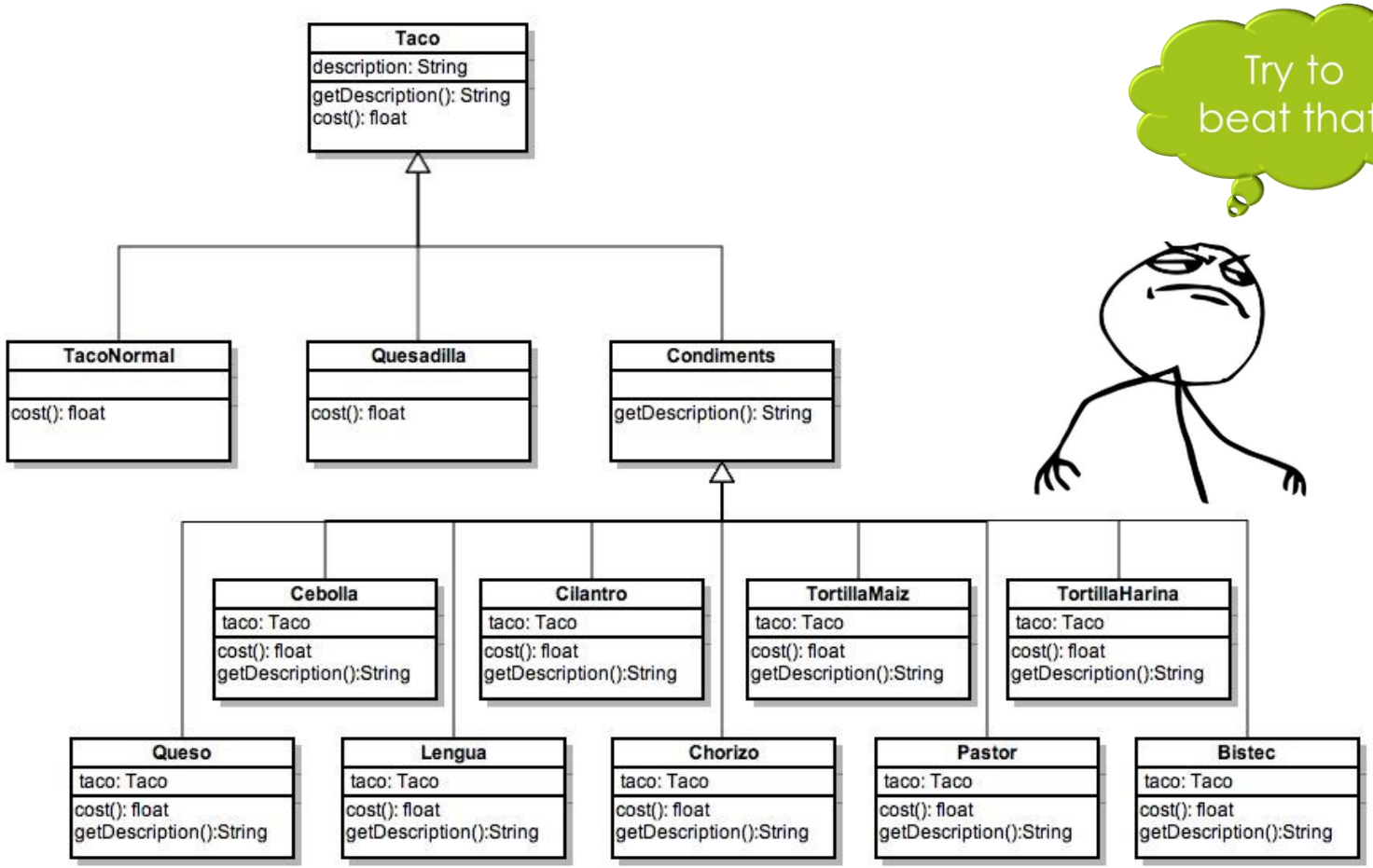


How to improve?

- Consider the following scenarios:
  - Someone wants to mix meats (Bistec con chorizo)
  - You want to sell quesadillas
  - Someone is on diet and does not want tortillas



TuTaco V1.4


Taco
description: String
getDescription(): String
cost(): float
TacoNormal
cost(): float
Quesadilla
cost(): float
Condiments
getDescription(): String
Cebolla
taco: Taco
cost(): float
getDescription():String
Cilantro
taco: Taco
cost(): float
getDescription():String
TortillaMaiz
taco: Taco
cost(): float
getDescription():String
TortillaHarina
taco: Taco
cost(): float
getDescription():String
Queso
taco: Taco
cost(): float
getDescription():String
Lengua
taco: Taco
cost(): float
getDescription():String
Chorizo
taco: Taco
cost(): float
getDescription():String
Pastor
taco: Taco
cost(): float
getDescription():String
Bistec
taco: Taco
cost(): float
getDescription():String
Try to beat that!

The implementation