



Tema 3: Disseny

Anna Puig

Enginyeria Informàtica

Facultat de Matemàtiques i Informàtica,

Universitat de Barcelona

Curs 2020/20201



UNIVERSITAT DE
BARCELONA

Temari

1	Introducció al procés de desenvolupament del software	
2	Anàlisi de requisits i especificació	
3	Disseny	
4	Del disseny a la implementació	
5	Ús de frameworks de testing	
		3.1 Introducció
		3.2 Patrons arquitectònics
		3.3 Criteris de Disseny: G.R.A.S.P.
		3.4 Principis de Disseny: S.O.L.I.D.
		3.5 Patrons de disseny

3.4. Patrons de disseny

Propòsit → Àmbit ↓	CREACIÓ	ESTRUCTURA	COMPORTAMENT
CLASSE	<ul style="list-style-type: none">• Factory method	<ul style="list-style-type: none">• class Adapter	<ul style="list-style-type: none">• Interpreter• Template method
OBJECTE	<ul style="list-style-type: none">• Abstract Factory• Builder• Prototype• Singleton• Object pool	<ul style="list-style-type: none">• Object Adapter• Bridge• Composite• Decorator• Facade• Flyweight• Proxy	<ul style="list-style-type: none">• Chain of Responsibility• Command• Iterator• Mediator• Memento• Observer• State• Strategy• Visitor

Patró Iterador

Nom del patró: Iterador

Context: Comportament

Problema:

- Aïllar el recorregut dels objectes que formen part d'un TOT

Solució:

- Defineix una interfície comuna per a col·leccions d'objectes per independitzar de l'estructura de dades que implementa la col·lecció.
- La interfície proveeix d'un conjunt de mètodes per a recórrer els elements d'una col·lecció.
- Java proporciona la interfície `java.util.Iterator` (amb els mètodes `hasNext()`, `next()` i `remove()`)
- tExisteixen tipus a Java com `Array` o `HashMap` que no proporcionen la implementació d'`Iterator` i el programador ha de implementar els seus propis mètodes.

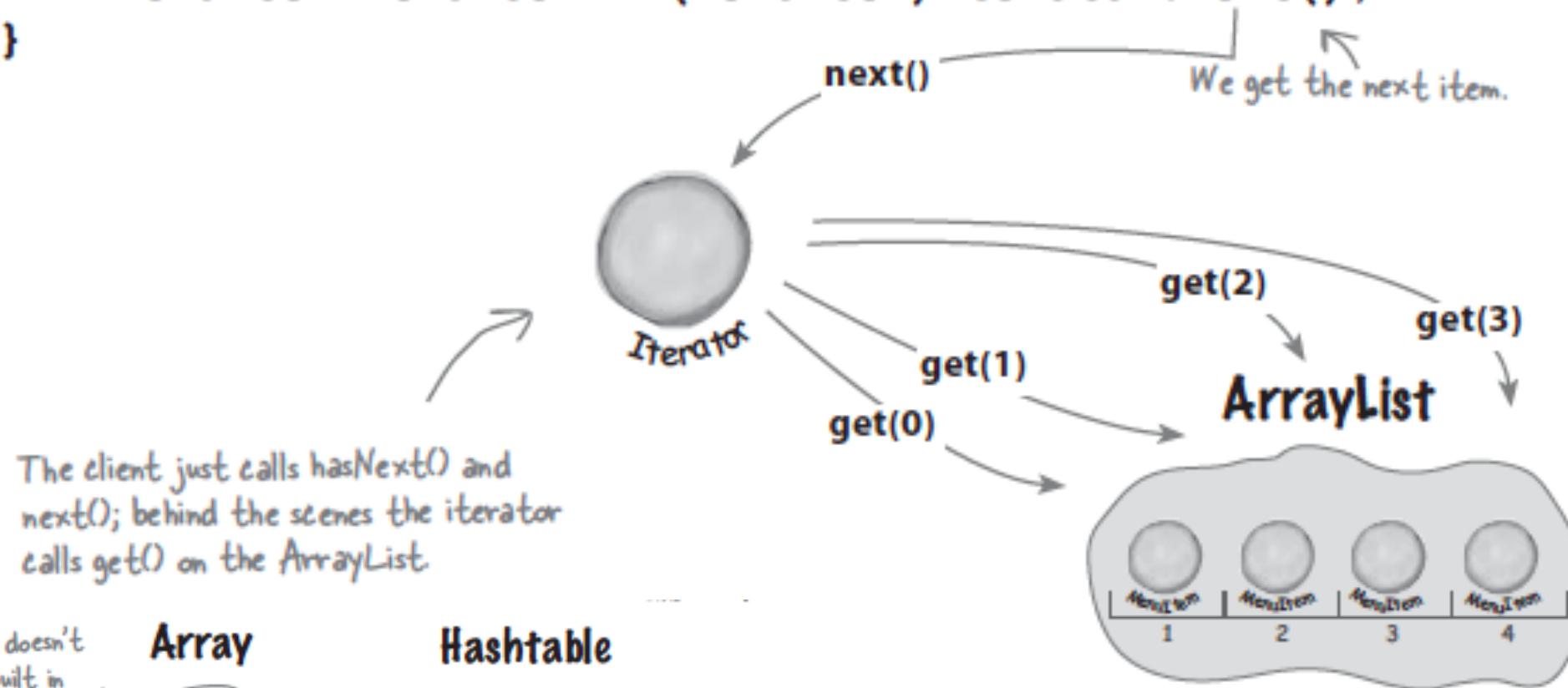
Patr3 Iterador

Iterator iterator

```
while (iterator.hasNext()) {  
    MenuItem menuItem = (MenuItem) iterator.next();  
}
```

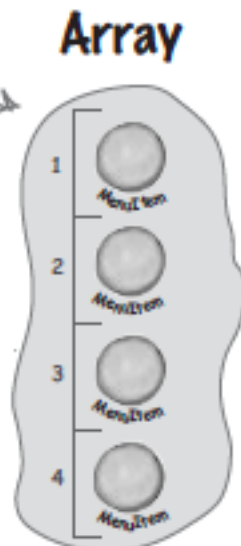
← And while there are more items left...

↑ We get the next item.

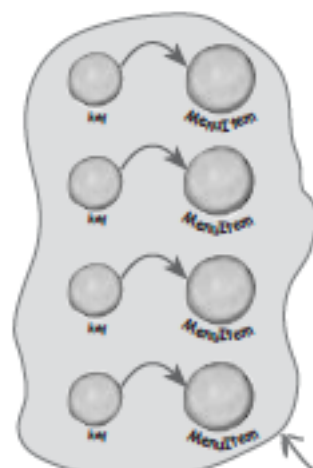


The client just calls `hasNext()` and `next()`; behind the scenes the iterator calls `get()` on the **ArrayList**.

... Array doesn't have a built in Iterator so we built our own.

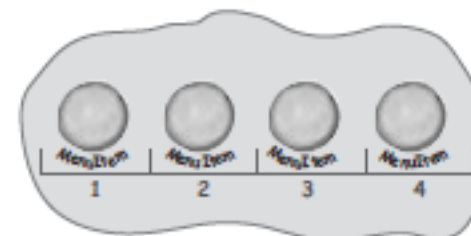


Hashtable

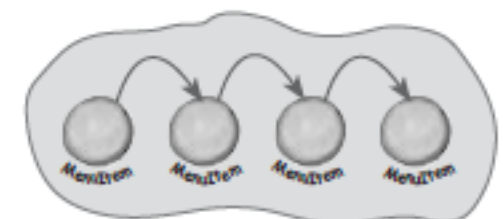


Making an Iterator for the Hashtable values was easy; when you call `values.iterator()` you get an Iterator.

Vector



LinkedList



...and more!

Patró Composite

Nom del patró: Composite

Context: Creació d'objectes

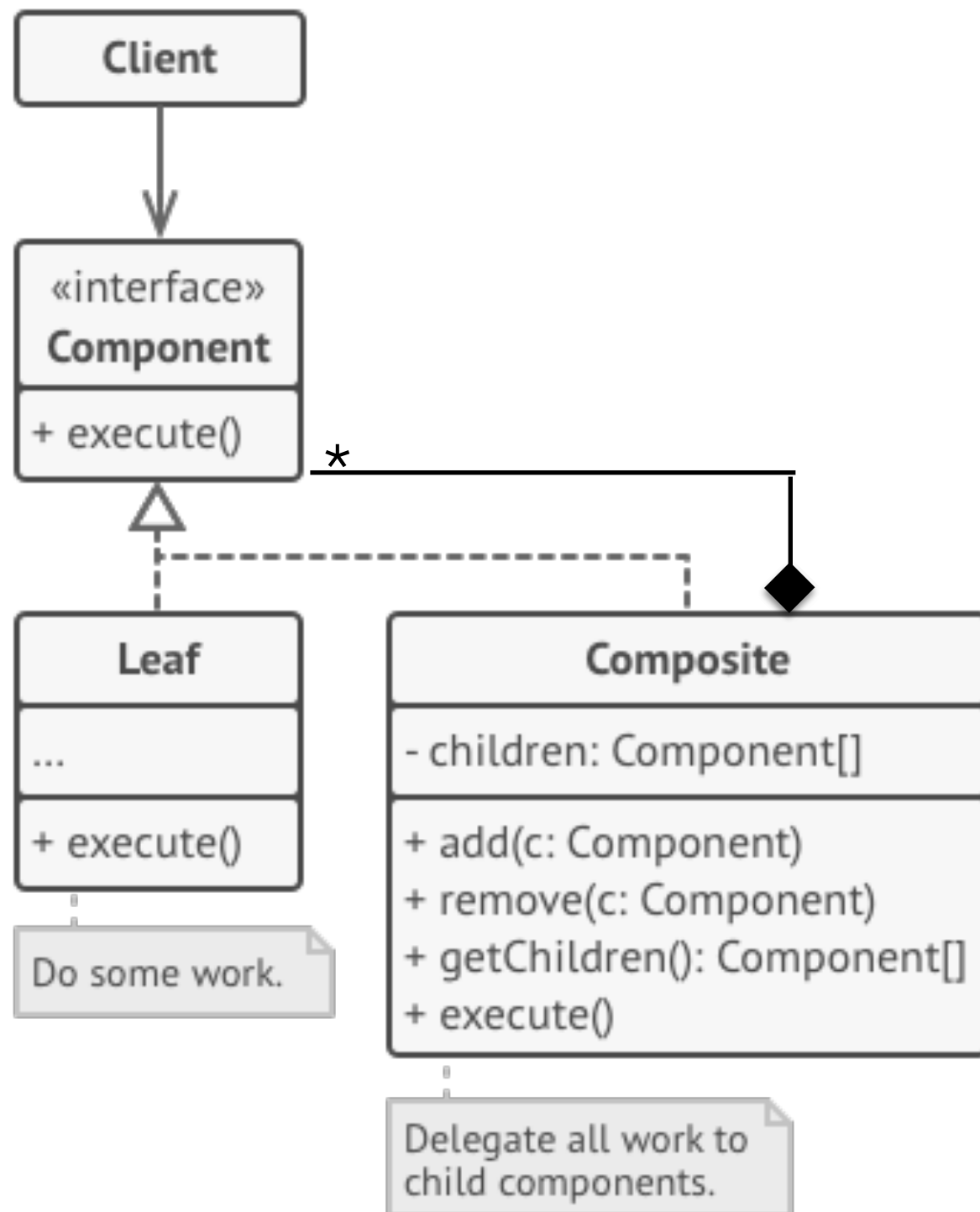
Problema:

- Composar objectes en jerarquies **TOT-PART** i permetre als clients tractar objectes simples i compostos de manera uniforme

Solució:

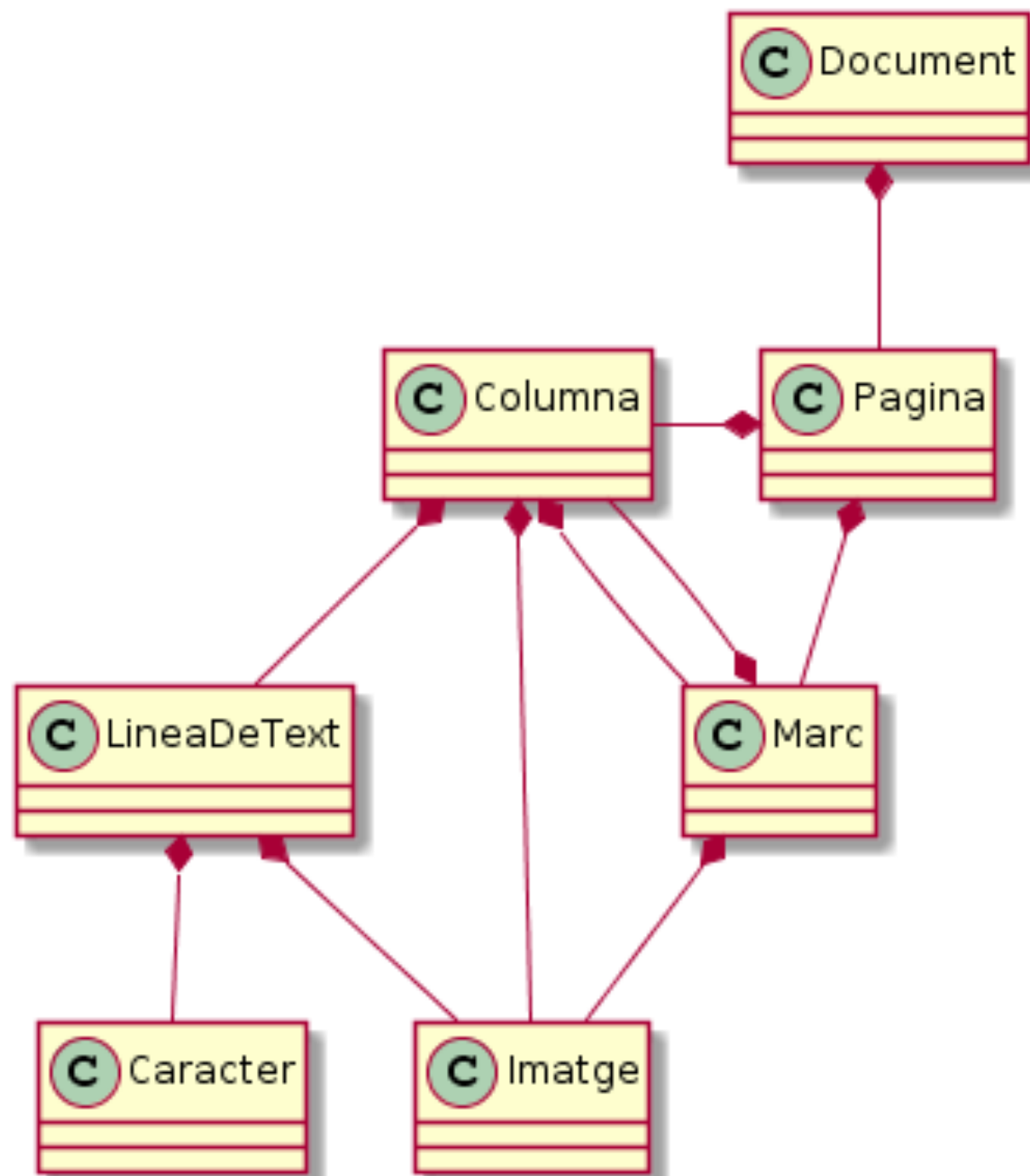
- Defineix jerarquies de classes que tenen objectes primitius i objectes compostos a la vegada que els compostos estan formats per objectes primitius o altres objectes compostos.
- Utilització de l'herència per modelar els diferents tipus d'objectes (simples i compostos)
- Utilització de la composició per modelar les relacions TOT-PART dels compostos

Patró Composite



- **Component**: declara una classe abstracta per la composició d'objectes
- **ElementSimple (Leaf)** representa els objectes de la composició que no tenen fills i implementa les seves operacions
- **ElementCompost (Composite)**: implementa les operacions per als components amb fills i emmagatzema els fills
- **Client**: utilitza objectes de la composició mitjançant la interfície de **Component**

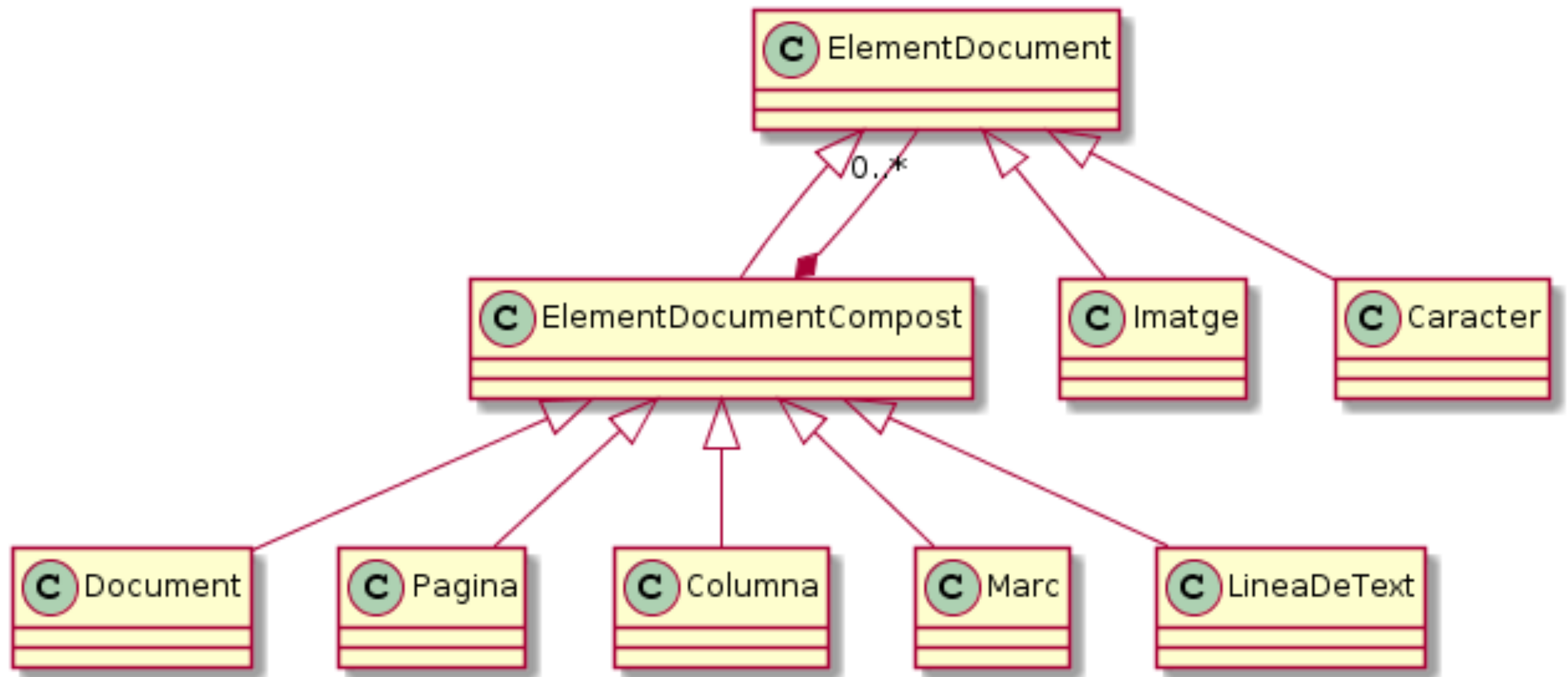
Patró Composite



Exemple:

- Un document està format per varies pàgines, les quals estan formades per columnes que contenen línies de text, formades per caràcters
- Les columnes i pàgines poden contenir marcs. Els marcs poden contenir columnes
- Les columnes, marcs i línies de text poden contenir imatges

Patró Composite



- Un document està format per varies pàgines, les quals estan formades per columnes que contenen línies de text, formades per caràcters
- Les columnes i pàgines poden contenir marcs. Els marcs poden contenir columnes
- Les columnes, marcs i línies de text poden contenir imatges

Patró Composite

Nom del patró: Composite

Context: Creació d'objectes

Pros:

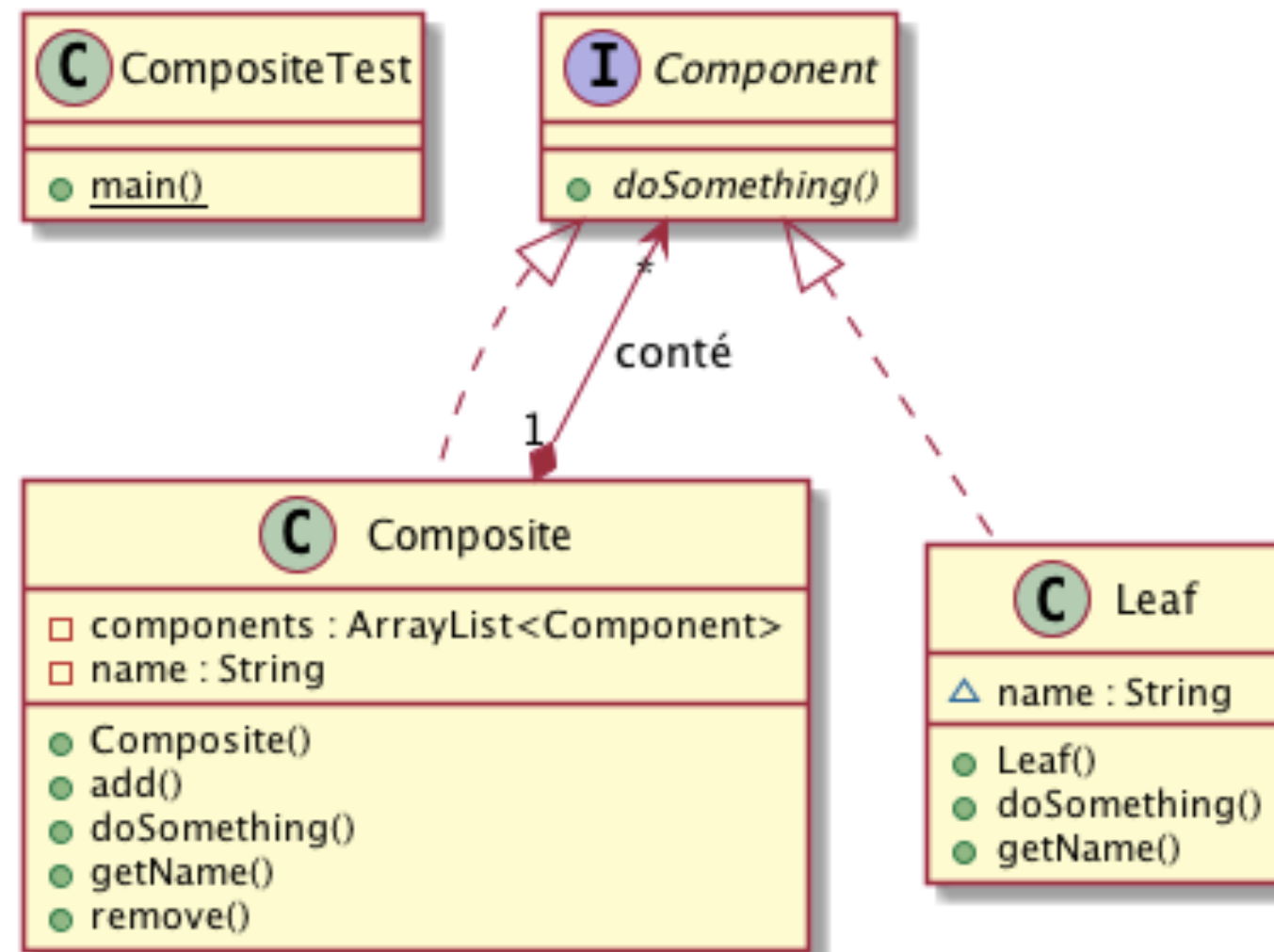
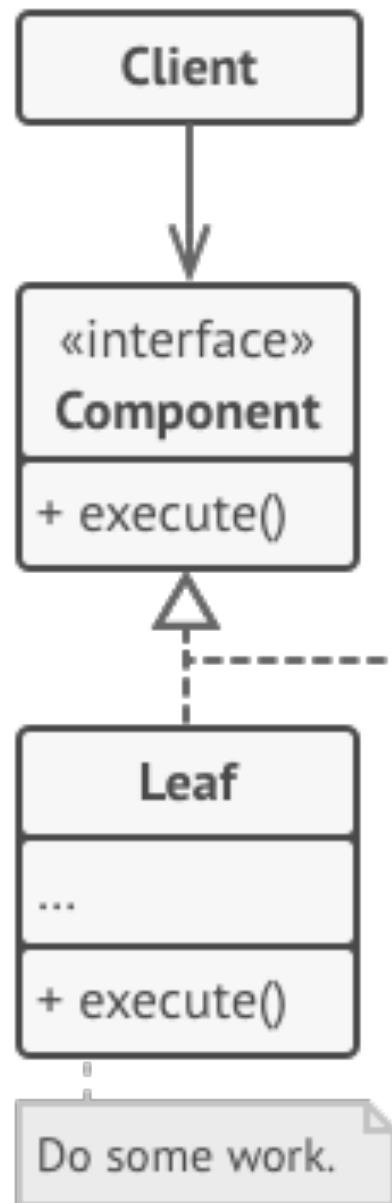
- Permetre el tractament uniforme d'objectes simples i complexes, així com les seves composicions recursives
- Simplifica el codi dels clients, que només usen una interfície
- Facilita afegir nous components sense afectar als clients
- Exemples en Java: JFrame, JPanel, JComponents

Cons:

- És difícil restringir els tipus dels fills
- Les operacions de gestió dels fills en els objectes compostos s'han de codificar les comprovacions en temps d'execució.

Exercici

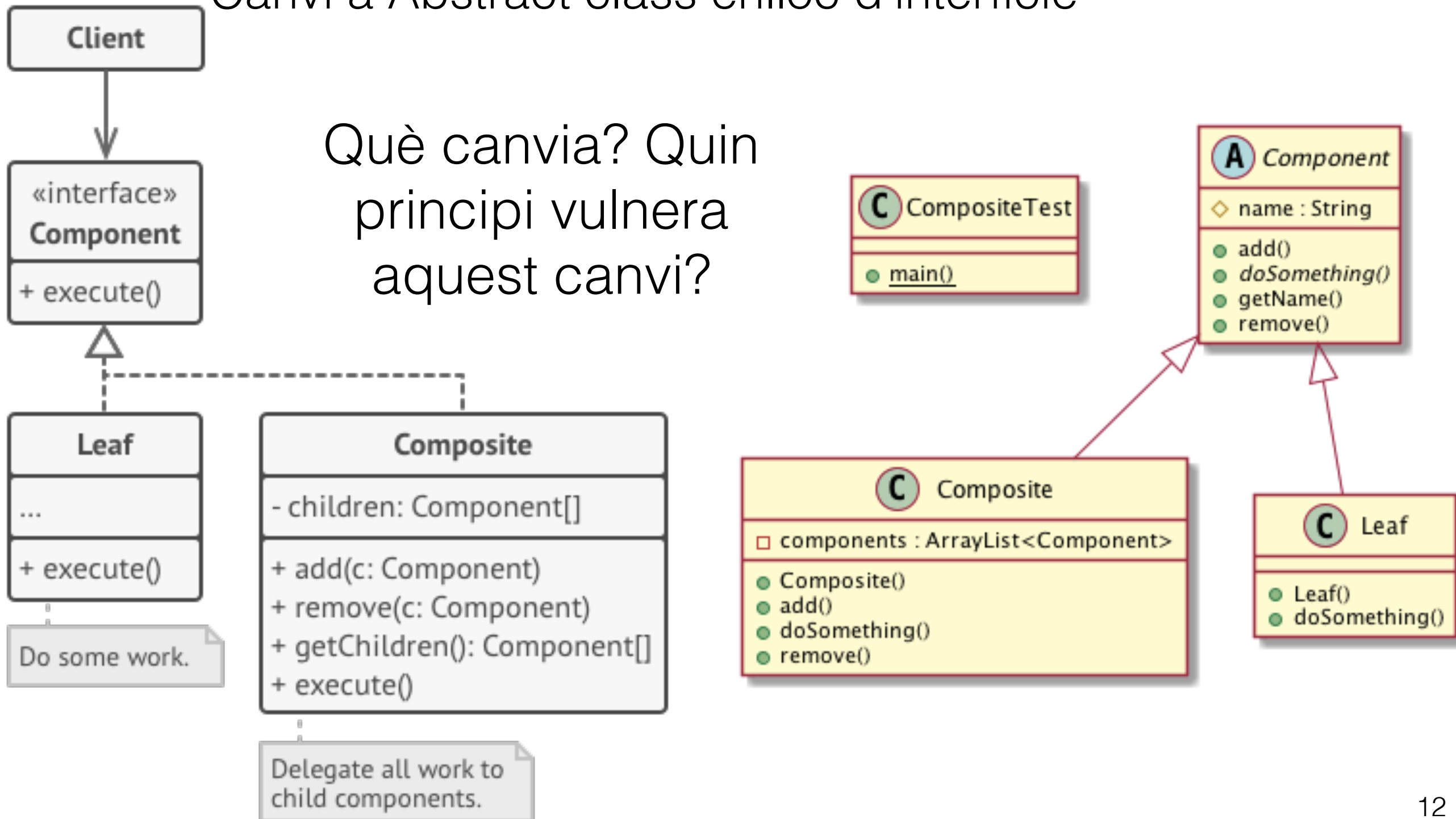
Veure la carpeta patternBasic del projecte del campus: Què passa en el main?



Exercici

Veure la carpeta pattern del projecte del campus:
Canvi a Abstract class enlloc d'interfície

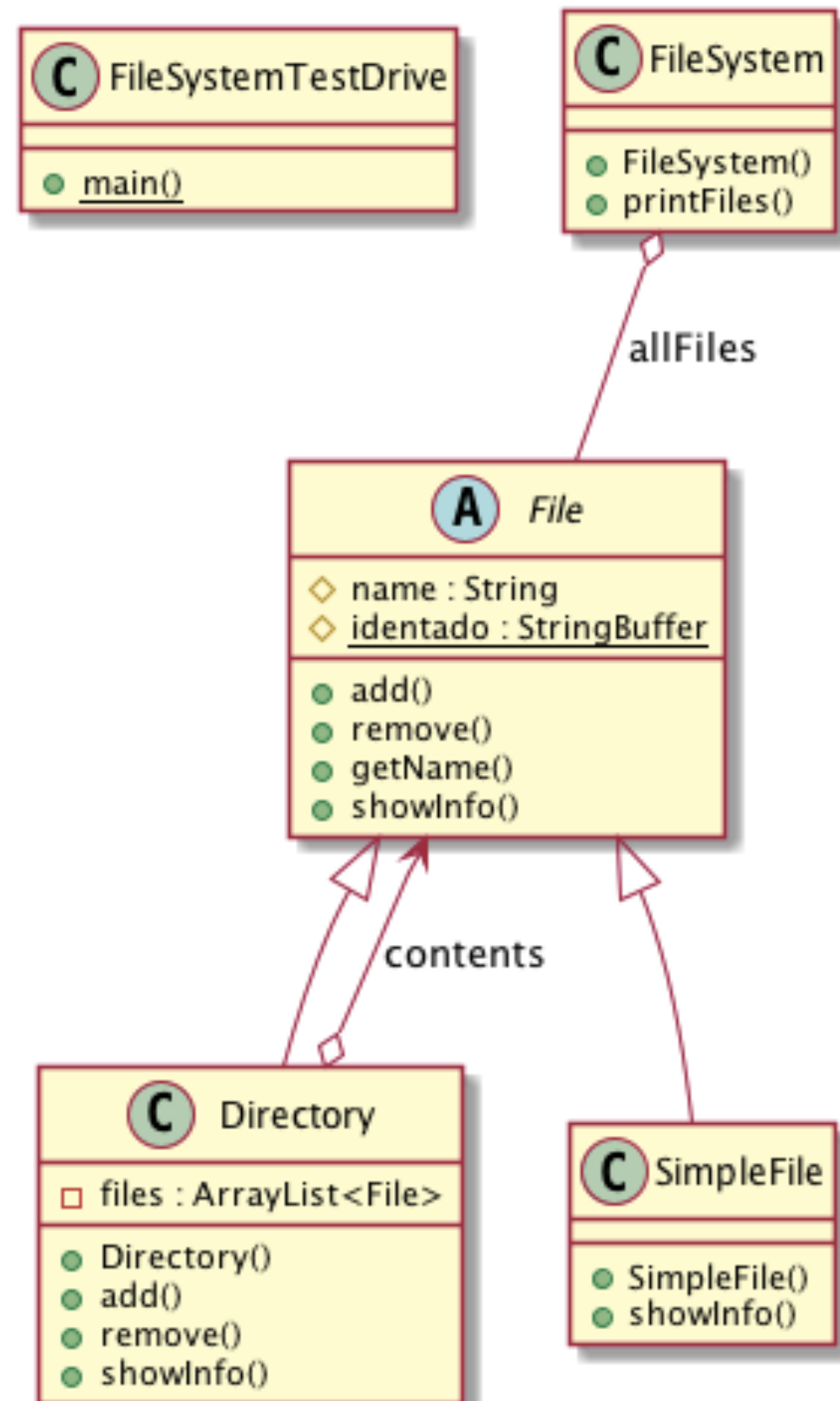
Què canvia? Quin principi vulnera aquest canvi?



Exercicis

1. En el projecte Composite del campus, explora l'aplicació del patró Composite que modela un sistema de Fitxers:

- quins principis vulnera?
- identifica quina classe és la Component del patró
- identifica quina classe és el Composite del patró
- identifica quina classe és la Leaf del patró



Exercicis

2. En el mateix Projecte Composite: Completa l'exercici que modela el llistat d'un menú d'un restaurant. Es vol obtenir el llistat següent:

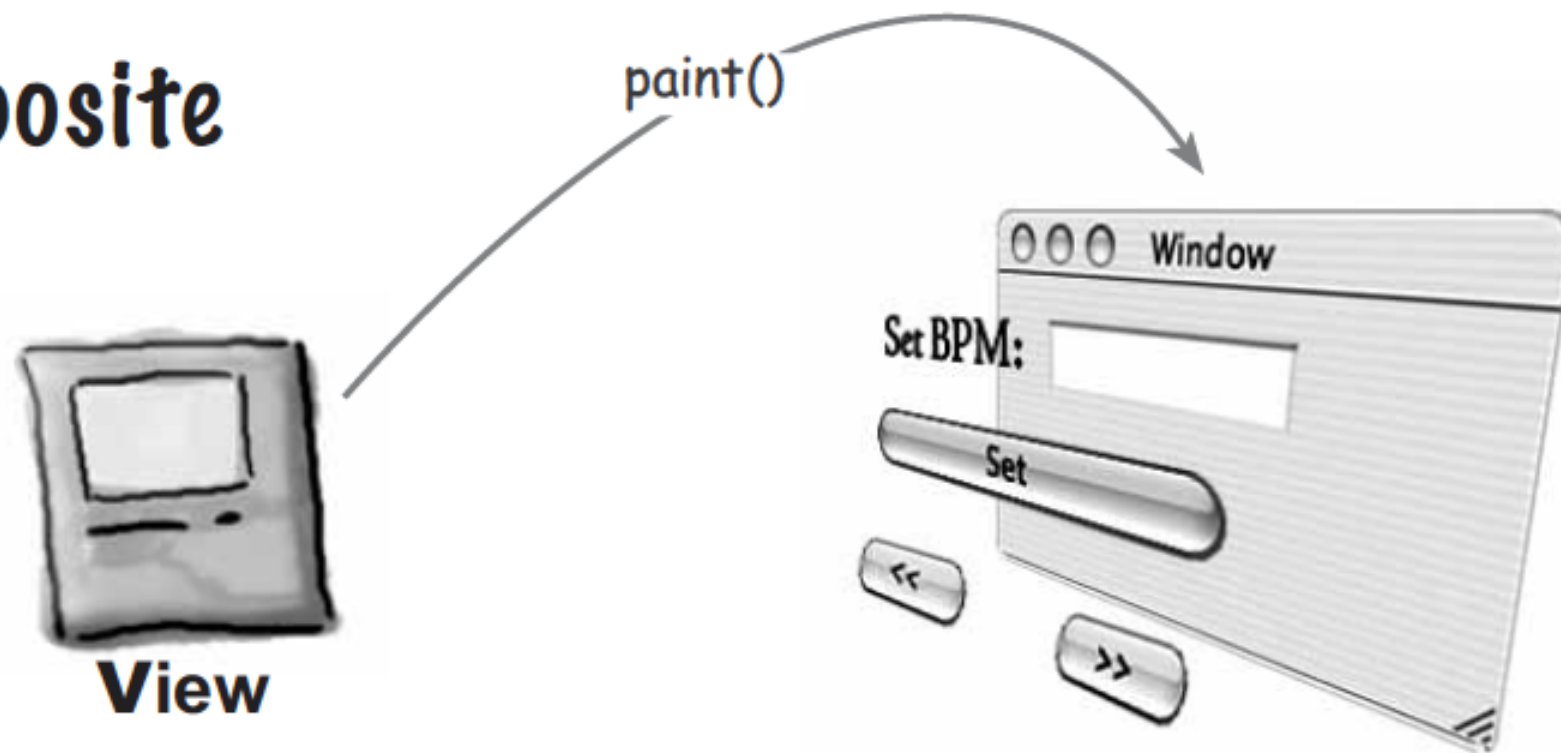
```
* Menus
  * Comidas
    * Plato Fuerte
      # Crispy Chicken,100.89
    * Postres
      # Apple Pie,15.59
      # Cheesecake,19.99
  * Cenas
    # Hotdogs,6.05
    # Spaghetti (v),30.89
```

Model-Vista-Controlador

Patrón en la Vista:

- Composite

Composite



The view is a composite of GUI components (labels, buttons, text entry, etc.). The top level component contains other components, which contain other components and so on until you get to the leaf nodes.