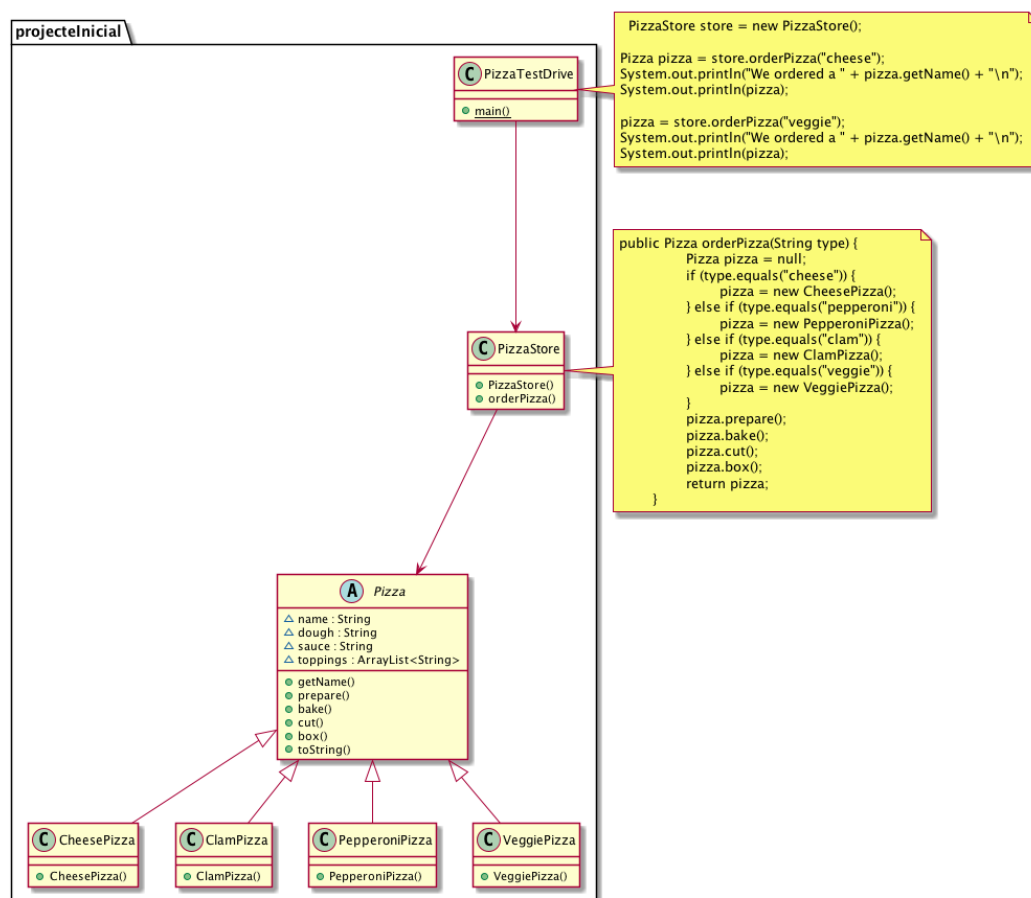


Exercicis de Patrons: Factory (Factory method i Abstract Factory)

Es vol modelar una pizzeria que ofereixen pizzas de formatge, de peperoni i vegetals. Cada pizza es crea, segons el seu tipus, es prepara, es cou, es talla, es posa en una caixa i es lliura la pizza. Un dissenyador novell ha fet el projecte PizzaStore (<https://campusvirtual.ub.edu/mod/resource/view.php?id=2668609>) ha dissenyat una classe Pizza de la qual hereten les classes les diferents classes de pizzas, Totes elles comparteixen els mètodes que estan implementats en la classe pare, però tenen atributs diferents.

- a) **PROJECTE INICIAL:** Quin principi S.O.L.I.D. vulnera aquest codi? Mira el mètode orderPizza() de la classe Pizza. Si hi hagués alguna classe més que volgués crear pizza, seria reutilitzable? Com solucionaríem aquest problema? Quin patró de Factories usaries? Fes una carpeta nova anomenada **Solucio1** i es el refactor del projecte inicial.

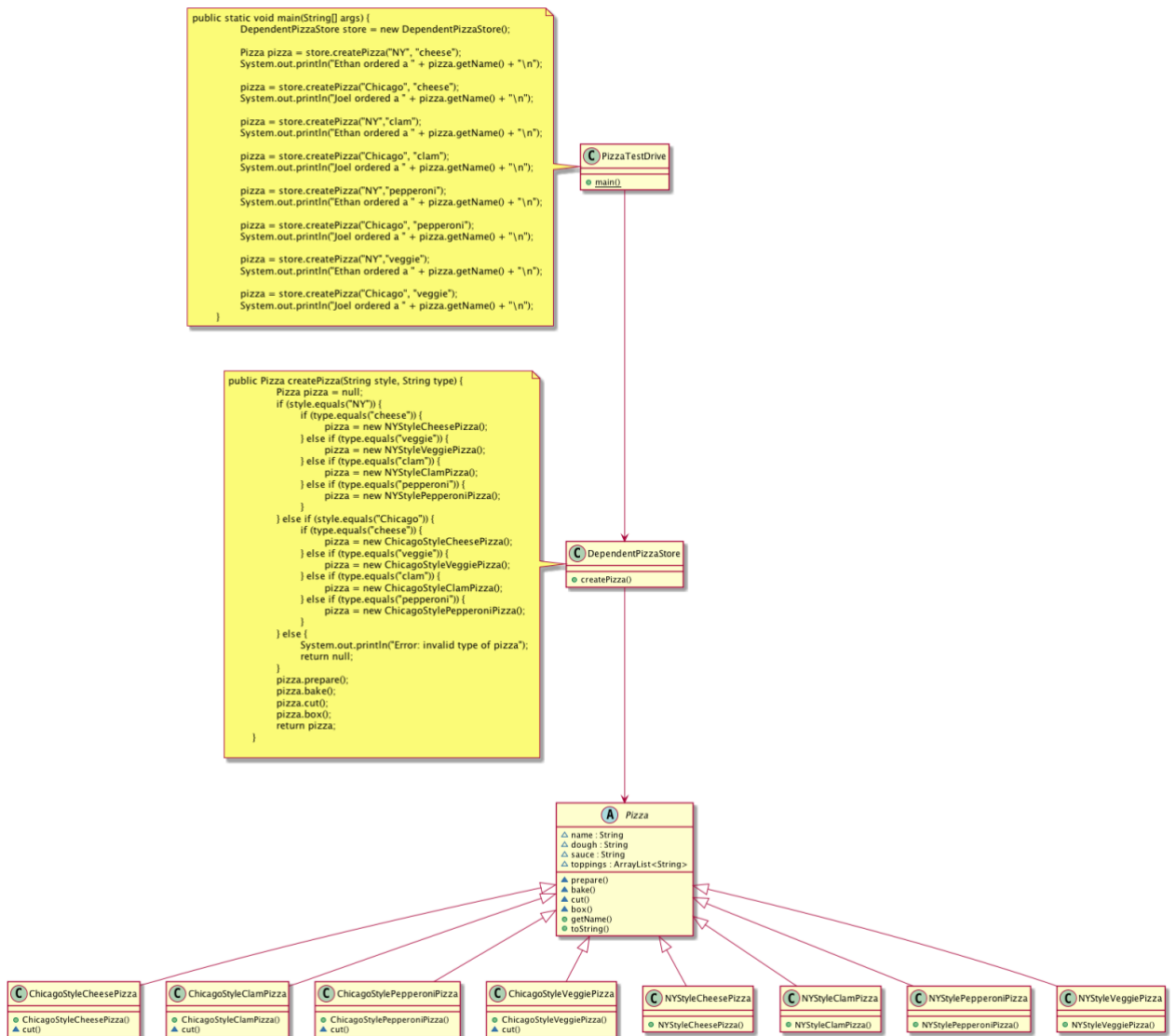
PROJECTEINICIAL's Class Diagram



- b) **PROJECTE FRANQUÍCIES:** En el cas que es volguessin fer varies franquícies de la Pizzeria, on totes les franquícies tenen el mateix tipus de pizzas però cuinades

de formes diferents, com s'hauria de modificar el codi? Imagina que tens franquícies a New York i a Chicago. Quins principis S.O.L.I.D. estaries vulnerant? **Fes una carpeta nova anomenada Solucio2 aplicant el patró de Factories adient per a aquest projecte de franquícies.**

PROJECTEFranquícies's Class Diagram



- c) **PROJECTE INGREDIENTS:** En el cas que es volguessin afegir ingredients a les diferents pizzas, que depenen també del tipus de pizzas i de la franquícia, quin problema es tindria en crear les classes? Cada ingredient de la pizza (massa, salsa, formatge, etc.) es fa de forma diferent segons siguin pizzas de NY o de Chicago. S'ha utilitzat el patró d'**Abstract Factory** per a solucionar aquest problema. Mira la solució plantejada al projecte `projecteIngredients` del campus i intenta fer el seu diagrama de classes. Mira com es correspon al patró **Abstract Factory** de la teoria. Quines classes equivalen al Client, AbstractFactory, ConcreteFactory, Product i ConcreteProduct?