Scrivere un software che per gestire una cucina, il software ha le seguenti classi:

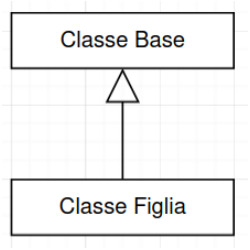
- Cook (Cuoco)
- Dish (Piatto)
- Pizza
- Pasta
- Dessert
- IceCream
- Cake

Concetto di Ereditarietá:
Due classi sono legate dall'ereditarietà quando il verbo che le lega è il verbo **essere**. La classe che eredita si dice classe figlia e la classe da cui si eredita classe base. Ad esempio, la classe guerriero é classe figlia della classe personaggio in quanto un guerriero **é** personaggio.

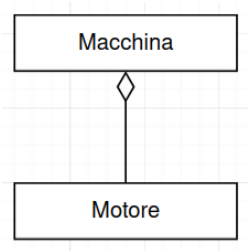
Concetto di Composizione:
Due classi sono legate dalla composizione quando il verbo che le lega è il verbo **avere**. Ad esempio la classe macchina **ha** un oggetto motore.

Simboli:

- Ereditarietá



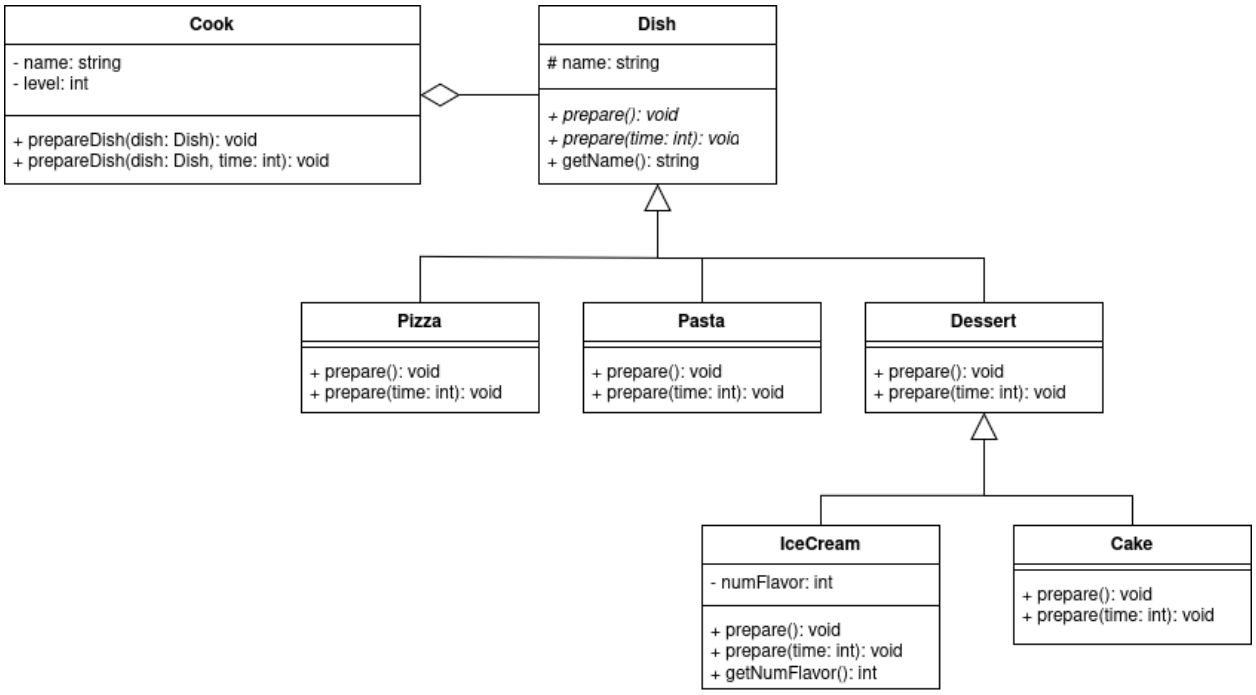
- Composizione



In questo task mostreremo per la prima volta un’architettura software interamente con il diagramma UML. Che mette in evidenza le classi con i loro attributi e metodi e i loro legami.

Task

L’architettura SW da realizzare é la seguente



Notiamo:

- La classe cuoco **HA** come attributi i piatti che puó cucinare e vengono passati nei metodi prepareDish. Sono due perché viene effettuato overload del metodo inserendo la tempistica con cui preparare il piatto.

- La classe Dish é una classe con dei metodi virtuali, con l’obiettivo di fare override dalle classi figlie.

- Le figlie specializzano la classe madre. Notiamo che l’attributo name viene ereditato da tutte le figlie perché é protected.

- Il cuoco ha a che fare solo ed esclusivamente con la classe base Dish che funge da interfaccia.

- La classe dolce (dessert) viene ulteriormente specializzata in due classi figlie. Quindi sia il gelato che la torta **SONO** dolci e anche piatti (il che ha senso se ci pensi).

Goal:

Realizzare l’architettura software che é stata descritta in questo file in modo tale che il main istanzi quanti piatti vuole e li fa preparare ad un cuoco.