

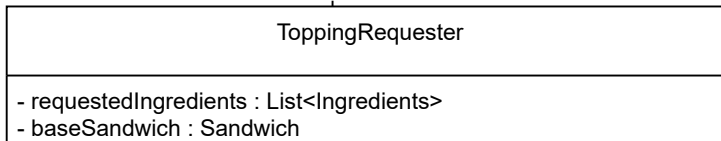
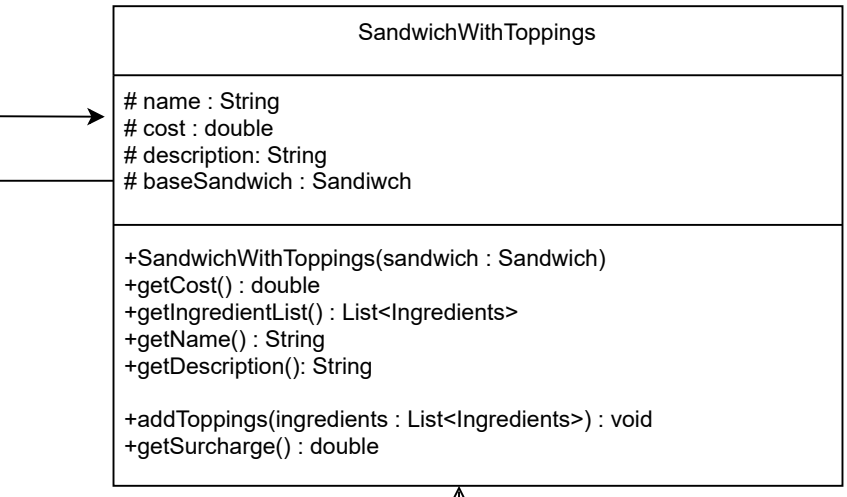
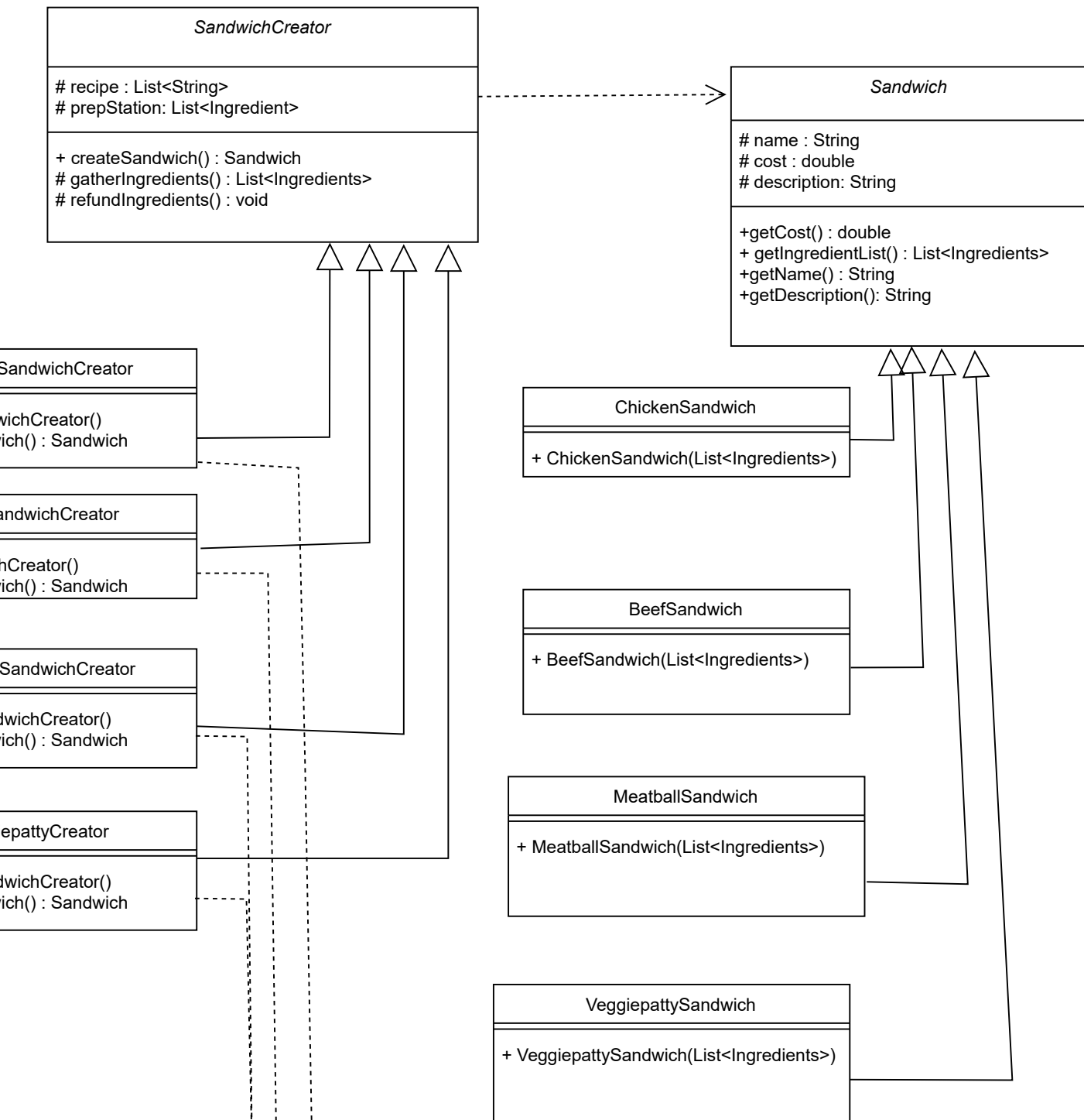
Singleton

We chose to use the Singleton design pattern for the Inventory system. There should only ever be a single instance of the inventory class at a time. Singleton will get that instance or make a new inventory if it doesn't exist.

Factory Design Pattern

Chose to use the Factory design pattern. This allows us to change the properties of a specific sandwiches while maintaining comparability with the rest of the app.

For example: if we wanted to add a +50% price premium on meatball sandwiches, there would be no need to change code anywhere but the MeatballSandwich subclass. The rest of the app will handle this new behavior.



Decorator Design Pattern

Used the decorator design pattern. Decorates the original sandwich object with a toppings. Achieved by taking the original sandwich and applying more ingredients and creating a wrapped sandwich.

We can add toppings to sandwiches regardless of which base sandwich was created, otherwise every base sandwich would need to be modified with toppings.

