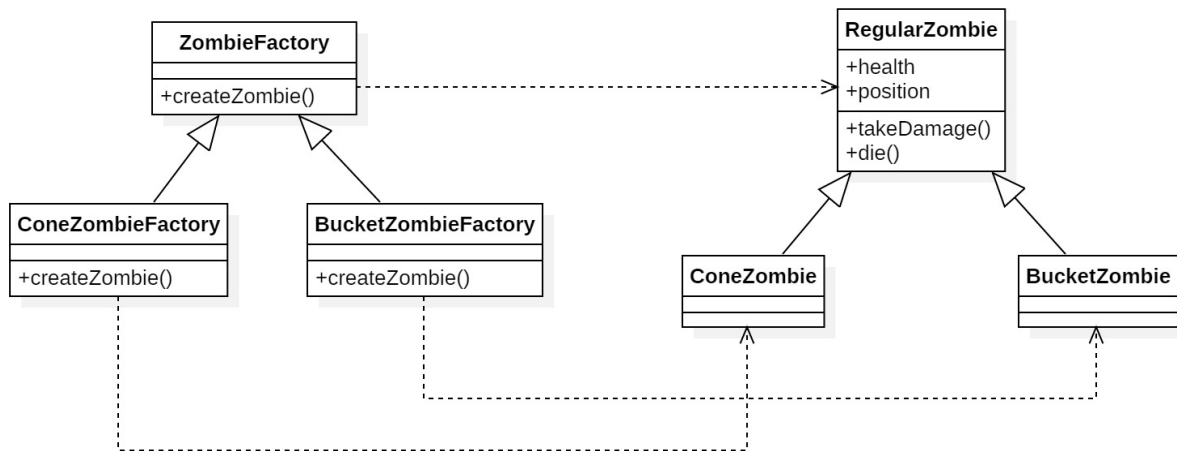


If you choose to go with the simple inheritance, the UML diagram is very straightforward.



In this case, the **RegularZombie** is a “Product” class that other classes inherit from, but also a “ConcreteProduct” itself. It’s almost identical to the Maze example in the lecture.

The create process would involve another class, presumably a “Client” class, that chooses which factory to use when the Client needs to create a zombie.

Another option to make the diagram more like the Pizza store example, is to make an “*Zombie*” abstract class, and add another ConcreteProduct named “**RegularZombie**”. This way, we have 3 concrete products: **Regular Zombie**, **Cone Zombie**, and **Bucket Zombie**. We will need to add another **RegularZombieFactory** as well. This solution will be mostly identical to the **PizzaStore** example, except that each factory only produces one concrete product.

Please note that for this assignment, Factory Method is sufficient as we are not introducing another level of complexity to warrant the usage of Abstract Factory. Even if you choose Abstract Factory, the resulting diagram is still essentially the same as Factory Method.

Meanwhile, in choosing Factory Method, you **must** have one dedicated factory for each individual product. If you have only one factory that has multiple different factory methods, that sort of violates the purpose of the pattern, and the TA will deduct some points.

Finally, whenever you are asked to apply a design pattern, make sure all the participants, and arrows/associations are properly matched.