**1.2. Open / Close Principle:**

**1.** The classes should be open for extension, but closed for modifications.

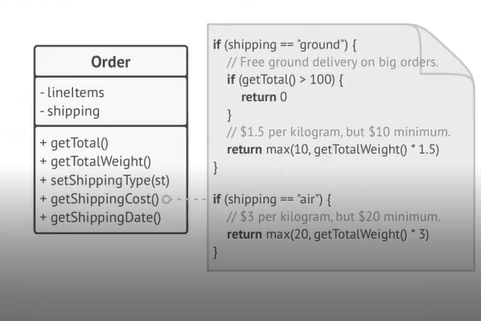
**2.** The main goal of this principle is to avoid breaking existing code when new features needs to be implemented.

**3.** A class is open if you can extend it, for instance, by producing a subclass and do extend the superclass behavior (add new methods or fields, override base behavior) or by adding fields to the data structures, it contains.

**4.** A class is closed if it is ready for being used by other classes (its interface is clearly defined and will not be changed in the future).

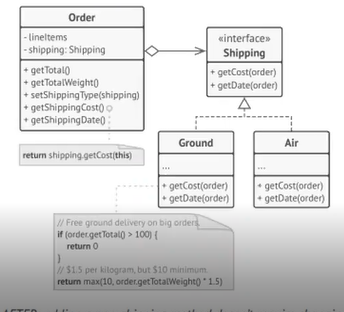
**Note:** issues on the base classes must be fixed in the base class. In this principle, we should not create sub classes to fix issues in the base class.

**- Example:**



In this example, we can see that the “Order” class contains a method to calculate the Shipping Cost based on the Shipping type. The problem of the current implementation is that the getShippingCost() method (so the “Order” class), will need to be modified every time that a new Shipping type is added. Those changes could break the existing code.

**- Solution:**



Create an interface defining properties and behavior for every shipping type. Then the “Order” class will contain an attribute shipping, and every shipping type instance will know how to get its own shipping cost. If a new shipping type is added, we will need to create a new Shipping class, implements the interface, and implement the new way to calculate the shipping cost. Following this rule, we will avoid to break the existing code.