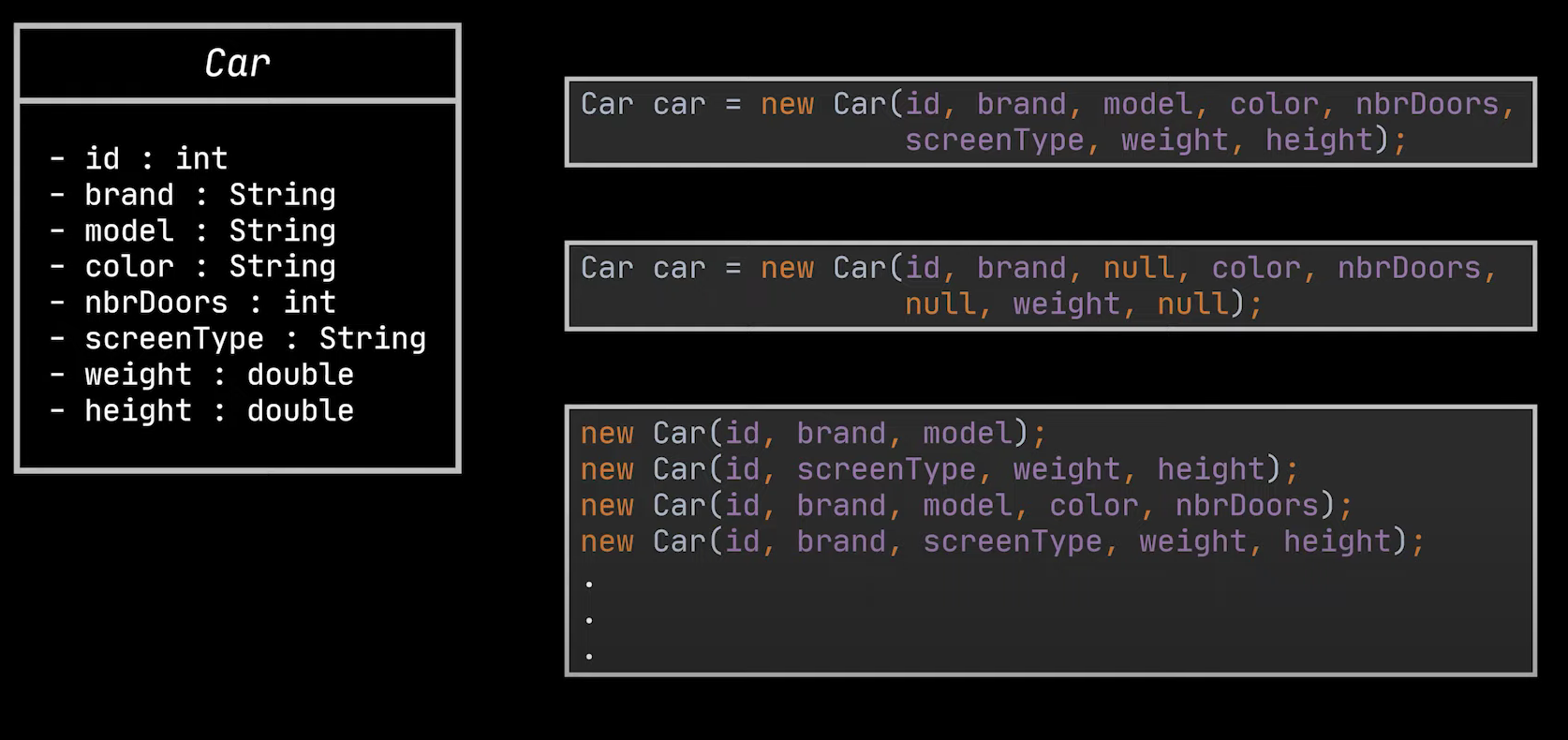
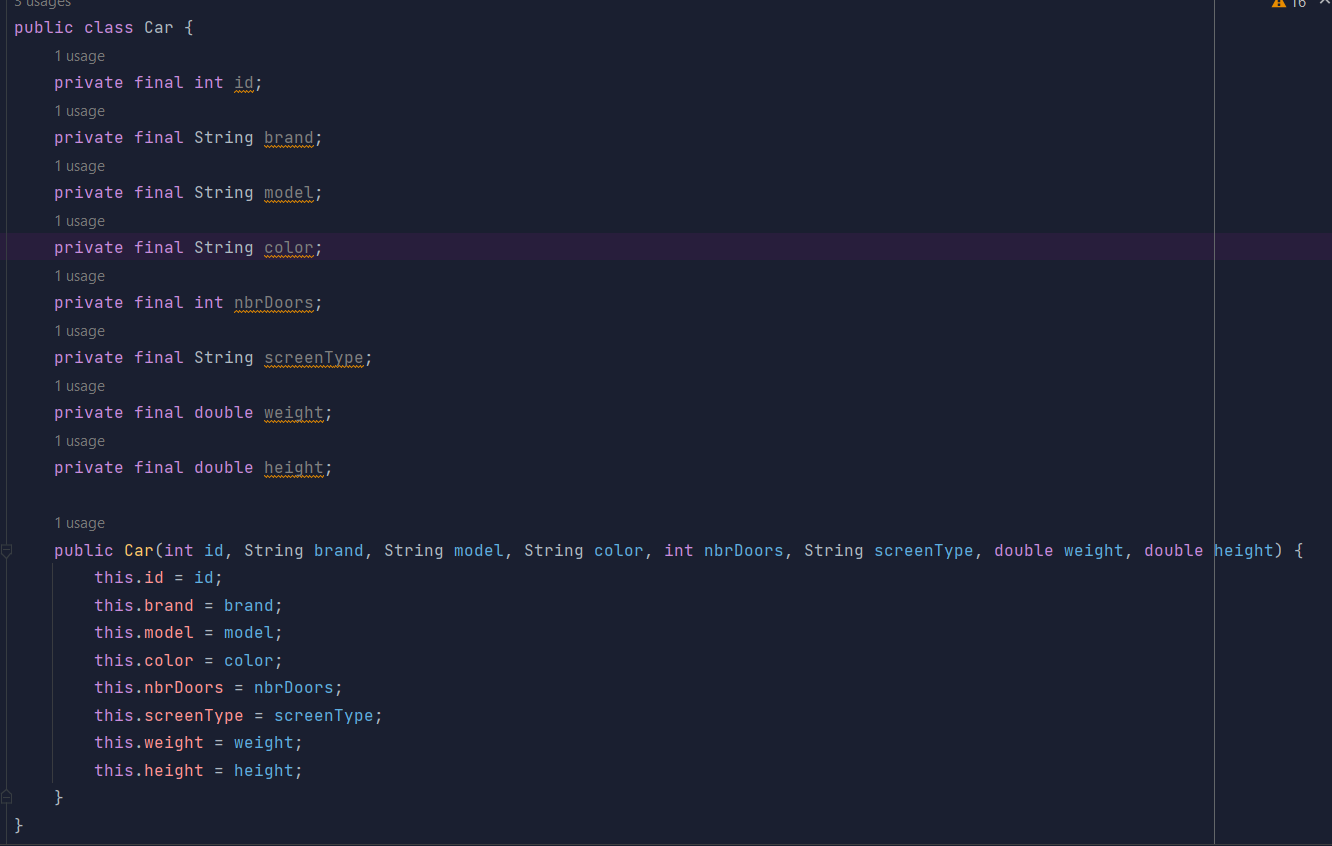
Demo Builder Pattern:

Ví dụ ta có class Car với một constructor khổng lồ chứa nhiều tham số, khi tạo ra một instance sẽ phải nhập đi nhập lại những tham số, rất dễ gây nhầm lẫn và khó để nhớ thứ tự của từng tham số:

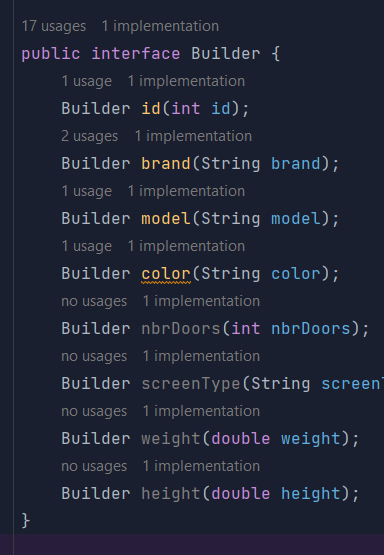


Vậy hãy cùng áp dụng Bulder Pattern:

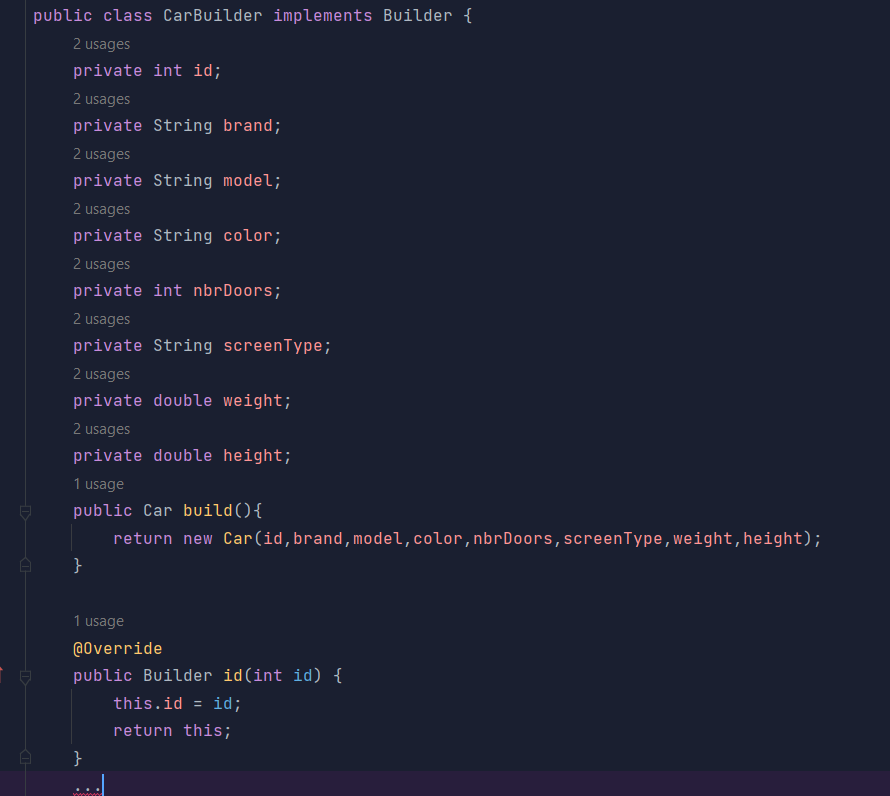
Class Car:



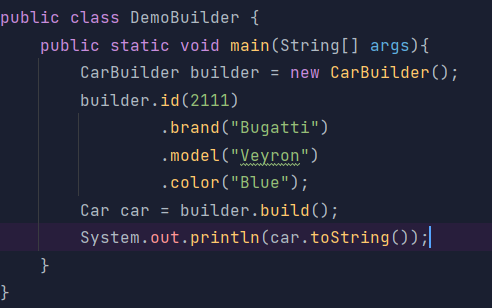
Interface Builder là thành phần AbstractBuilder sẽ chứa các method build xe:

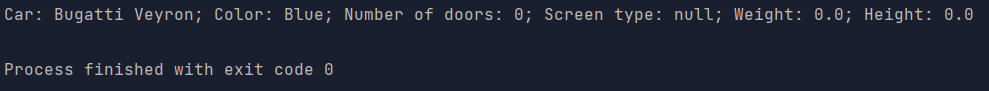


Class CarBuilder là thành phần ConcreteBuilder, implement interface Builder, override lại các method của Builder và chứa method build() trả về một instance của một object Car:



Vậy là chúng ta đã có thể dễ dàng khởi tạo một object Car bằng cách tạo một object CarBuilder và nhờ nó build một chiếc xe:





Trong thực tế, việc build những object có chung thuộc tính xảy ra rất nhiều, vì thế nên cần có một class Director, đóng vai trò như một người chỉ hướng, tạo ra các preset có thể được sử dụng lại nhiều lần, ví dụ:



Method buildBugatti(…) sẽ tự động build một chiếc xe có brand là Bugatti màu xanh, 2 cửa và cao 115cm, tuy nhiên ta vẫn có thể thay đổi và thêm thuộc tính, điều này rất tiện và tiết kiệm thời gian:

