# 定义

引言:

在建造一艘航空母舰时，我们通常会将航母分成若干个子模块，各个子模块各自独立构建，然后由航母设计师将各个模块按照航母的设计图有序的拼接起来，这样航母就造好了。当然如果我们要建造一艘驱逐舰，那么驱逐舰设计师会按照驱逐舰的设计图，将组成航母的模块按照驱逐舰设计图有序的拼接起来，这样驱逐舰就造好了。

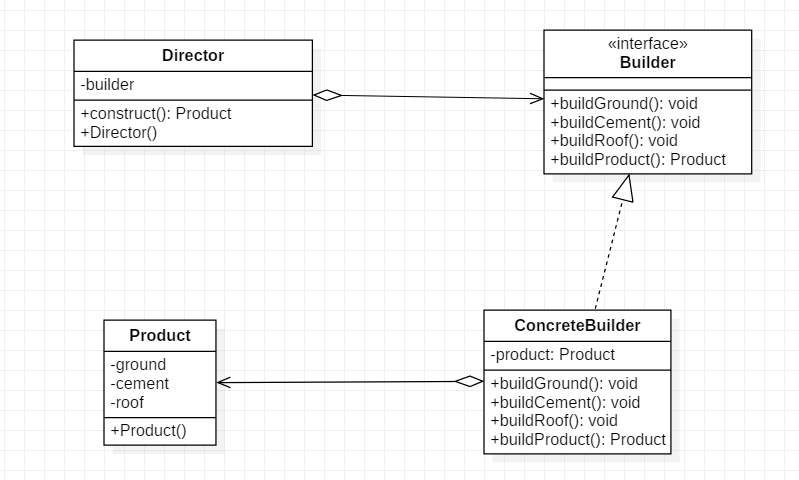
概念:

将一个复杂的对象的构建与它的表示相分离，使得同样的构建过程可以创建出不同的表示，这种方式叫做构建者模式或生成器模式。

# **组成角色**

1. 建造者(builder):Builder角色负责定义用来生成实例的接口api
2. 具体的建造者(concreateBuilder):ConcreateBuilder角色是负责实现Builfer角色定义的接口实现类。针对不同的商业逻辑，具体化复杂对象的各部分的创建，在具体化后提供产品的实例;
3. 设计师(director):负责使用builder角色的接口api来生成实例。内部不涉及具体的产品信息，只负责保证对象各部分完整创建或者按照某种顺序进行创建。即director是负责指挥如何build的，只要负责调度，具体实施交给具体的建造者;
4. 产品(product):即要创建的复杂对象；
5. 使用者(client):实际使用Builder模式的角色

# **建造者模式UML类图**



1. **优缺点**
2. 封装性：

客户端不必知道产品内部组合细节，组需要关心我们要生成某个对象，具体对象产生细节不必要置校。Main类不知道Builder类，它只是调用了Director类的construct方法完成对象的获取;

1. 建造者独立，易于扩展：

上面我们只举例ConcreteBuilder建造者类，如果需要其他建造者新建类即可。建造者彼此之间独立，系统扩展性好，符合开闭原则;

1. 便于控制细节风险:

由于具体建造者是独立的，因此可以对具体的建造过程逐步细化，不会对其他模块产生影响。

1. **应用场景**

建造者模式的典型应用场景如下:

1. 产品类非常复杂，不同的调度产生不同的结果时，使用建造者模式比较合适;
2. 相同的组件或配件都可以装配到一个对象，但是产生的结果又不相同。
3. **建造者模式和工厂方法模式的对比**

建造者模式关注零件类型和装配顺序，建造者模式和工厂方法模式都是创建型模式，重点不同。工厂模式只有一个建造方法，而创建者模式呦多个建造零部件的方法并强调 顺序，而工厂模式并没有顺序的概念。