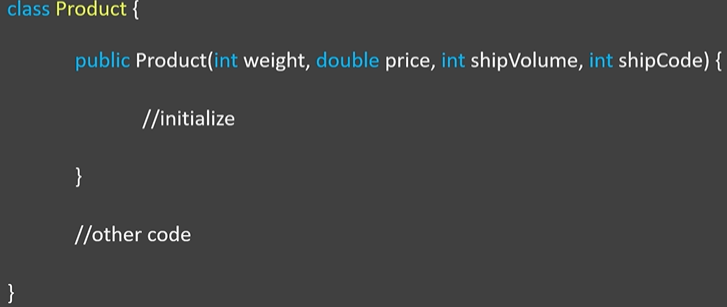
Builder

Creational Design Patterns

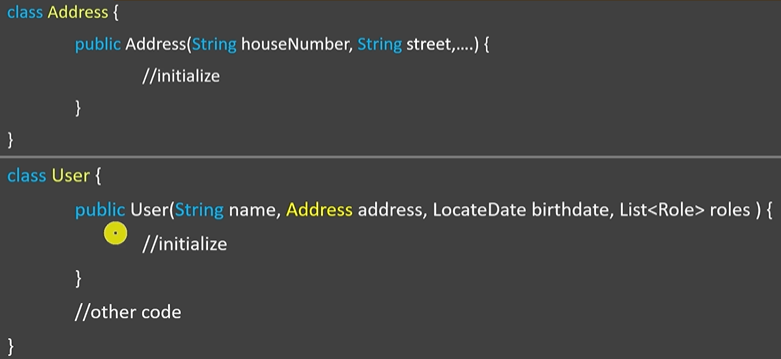
**What problems does Builder solve:**

1. Class Constructor requires a lot of information:



Having a method or constructor that needs a lot of arguments is considered a bad code practice, because anyone who is using such a code has to find out what all those parameters mean so that he can pass the correct values in a correct sequence.

1. Objects that need other objects or parts to construct them:

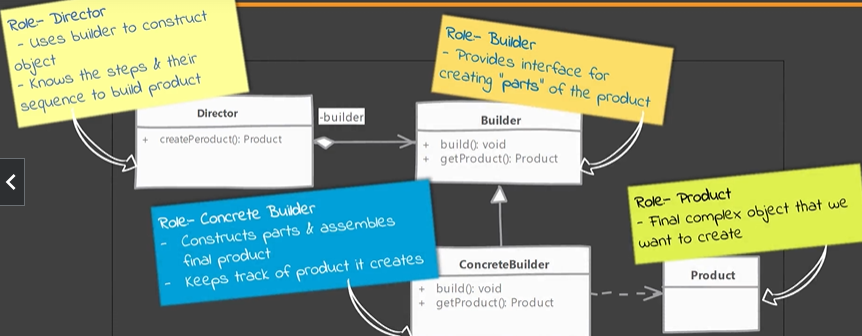


In order to create a User object, you must first create an Address object, multiple Role objects and then put them in a list. Builder design pattern is of great use to use here.

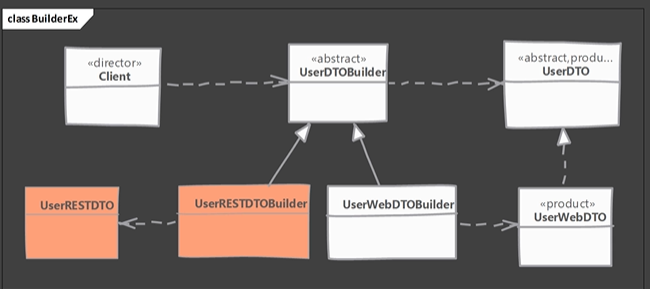
**What is Builder:**

1. Use the Builder Design Pattern whenever we have a complex process to create an object - multiple arguments, objects as arguments; involving multiple steps.
2. The Builder Design Patterns removes the object creation logic from out code and we abstract it out into multiple separate classes so that the user of our object can easily create objects.

**UML:**

****

**Example:**

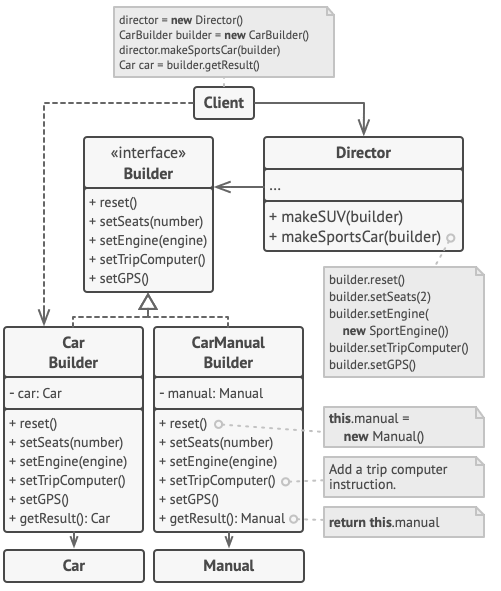
****

Concrete Builder -UserWebDTOBuilder; UserRestDTOBuilder

Builder - UserDTOBuilder

Product - UserWebDTO

Director - Client



**Design Considerations:**

1. Director role is rarely implemented as a separate class, typically the consumer of the object instance or the client handles that role.
2. Abstract builder is not required if the product itself is not part of any inheritance hierarchy.

**Implementation:**

1. Create the Director. (creates the different variants of objects).

Example: Director contains methods makeSUV, makeSportCar or prepareVegMeal, prepareNonVegMeal.

1. Create the Components of the objects: Engine, Seats, etc.
2. Create the final products – Car & ManualCar or Burger & ColdDrink
3. Create a Builder interface, that contains all the methods for constructing an object.

Example: Builder contains methods: setSeats, setEngine, etc. or contains: setPacking, setPrice, setName.

1. Create classes that implement the Builder: Burger & ColdDrink or Car & ManualCar Builders that create the according object.