Builder Design Pattern

What is this particular design pattern?

The builder design pattern allows us to make different objects and types, step by step whilst following the same guidelines as far as coding goes. By breaking our coding into manageable chunks that are more interchangeable/fluid.

What is the problem we are trying to solve?

Sometimes objects have many fields that may overlap, for example (from the YouTube video) with animals, we may have a duck base class that has a field saying it can fly. But when we keep adding more and more classes of different ducks, that cannot fly and fly in a different way, we would have to keep creating sub classes and branching off. This situation creates a lot more work and by creating so many layers just makes the project more confusing and prone to mistakes and errors.

What is the solution?

I think from the solutions, creating a large constructor in the base house class would work but again creating many sub classes can get messy. By creating separate objects for the traits of our ducks (fly() , eat(), size()) we can call upon them individually and when they are used. I was thinking that these are essentially our bricks, and once put together we *build* our class. By keeping the classes and constructors organized you can simply go to one place and update your code from there, minimalizing room for error.