

Description:

This is a smart house builder application that allows the user to choose a house they want to build including specific customizations they want it to have. For example, the user can choose anything from the outside color of the house, having a basement, a pool, and even the number of garages.

Design Pattern Used:

The design pattern used for this application is the Builder design pattern which makes the user experience easier in selecting the customizations for their house since they don't have to enter them all together in the constructor of the House class. Instead, the client can choose one customization at a time and enter what they would like to have for that customization and stop whenever they are done and the rest of the available customizations don't have to be selected.

Classes:

In order to implement the builder design pattern there had to be two classes, one parent class called House and a nested class called HouseBuilder. The House class has a private constructor that can be accessed by the HouseBuilder class in order to simplify the process of constructing a House object while adding the complex customizations that the clients need.

UML Diagram:

Please refer to the file `uml_diagram.pdf` in order to see the detailed UML diagram of the application code and how the builder pattern was implemented.

Future improvements:

In the future of this application I am thinking of making the different customizations of the house into classes that will get into more detail about each customization. For example, instead of having the houseColor customization as a String, I will make a class called HouseColor that will have different popular colors and can make the color darker or lighter etc. I will do this to all of the customizations and this way there will be more customizing within each category.