Running the Program

Clone the github repository and navigate to the src directory. Once in the src directory, compile and run the code by running the following commands in terminal:

javac Game.java
java Game

This will allow you to play one round of blackjack. To play additional rounds, run the command again:

java Game

Design Choices and Data Structures

Card objects represent playing cards, and they hold a Suit enum denoting the suit (hearts, clubs etc.) and a CardValue enum denoting the value of the card (2, 3, jack, queen etc.). Suit and CardValue are enums instead of strings because this ensures that each Card represents a valid playing card in a 52 card deck. A Deck object represents a standard 52 card deck, and stores Card objects in an array list. They are stored in an array list for easy and efficient removal, since the main purpose of the deck will be to remove cards to deal into the game.

A Hand object contains a list of cards that comprise a hand and contains a running total of the value of those cards combined. The Player object contains multiple Hand objects since a player can split their cards on their first move and thus play multiple hands. Although the game is a single player game, having a Player object allows for easy refactoring to make this game a multiplayer game.

Lastly, the Game object runs the game, moderating moves between a Player's different possible hands and the house's hand.

Limitations

There are a few limitations to this program that users should be aware of.

1. The program only supports one player versus the house.
2. A player may only split their cards once, whereas some versions of the game allow a player may split up to 3 times
3. There is no digital currency involved to actually "bet" with
4. If you wish to play the game multiple times, you must re-run the program

Tooling

The language used is Java. I chose to use Java because it has a very simple and easy to use implementation of an Array List, which I wanted to use to store cards in a deck for reasons explained above.

No external libraries were used, however I utilized java.util.Scanner to read input from the user, java.util.Random to generate random numbers to randomly deal a card from a deck, and java.util.ArrayList to access Java's Array List.