

Abstract Data Types:

- Circular Array Queue:
 - The Circular Array Queue stores all three players.
 - The Circular Array Queue keeps track of which player's turn it is by shifting all the elements so that the element at the front rotates between the three players.
 - We chose the Circular Array Queue because we will not have to dequeue and enqueue each individual player every time. Instead, the Circular Array Queue will keep track of whose turn it is.
- Stack:
 - Each player has their own stack of cards.
 - Each stack stores the incorrect guesses of the player, and functions as the "notepad."
 - The player can peek at their "notepad" and look at the last item in the stack for reference.
 - We chose this data structure because of the last in first out (LIFO) order. This allows the player to see their most recent incorrect guess.
- Possible Hashtable to report the professor, room, and weapon.

Classes:

- Player(String name)
 - Stores Player object in circular array queue
 - Guess() method
 - Creates an empty notepad
- Deck()
 - 3 arrays of professors, locations, and weapons
 - chooseCrime() method
 - choosePlayerCards() method
- PlayGame()
 - Functions as a driver class
 - Has a main() method
- Panel classes (All should have a private class button listener)
 - Home screen panel: starting screen where players type in their names and start game
 - Game board panel: screen with player tabs where they can see their cards, peek at their notepad, switch rooms, and guess
 - Switch room panel: screen where players can choose which room to switch to
 - switchRoom()
 - Notepad panel: screen where players can peek at their notepad
 - checkNotepad()

Methods:

- Player class will have a guess() method that takes a user's guess (professor, location, and weapon). The guess method will then determine whether they have guessed the correct answer or not. If yes, they win! If no, the guess() method will return one of the incorrect guesses.
- Deck class will have a method chooseCrime() in which a professor, a location, and a weapon are each randomly chosen from the deck of cards and placed in a new array. The array will store the correct answer. (For example: [Lyn, Elevator, keyboard]). The chooseCrime() method will occur once a player clicks the "Start game" button.
- choosePlayerCards() is a helper method within the Deck class that randomly gives each player four cards of the cards that are leftover in the deck.
- Various GUI methods. We have several buttons on our different panels (including "Switch rooms?", "Check notepad?", and "Start game.")
- Notepad class will have a checkNotepad() method that returns the top element from the player's notepad. Therefore, giving them a hint at a previously incorrect answer.
- The SwitchRoom panel class has a switchRoom() method that redirects the user to a new switchRooms() panel, where they will click a new room. This method will also change the label on the Game Board.
- The Notepad class has a checkPanel() method which will return a previous incorrect guess of the player. If they have no incorrect guesses, the program throws an EmptyCollectionException and will tell the user that their notepad is empty.
- The PlayGame class will have a main() method that functions as a driver.