```
/*
  Hunter Ginther
  HAGD5D
*/

public class Game extends Application {
  private Player player1; // Human/Computer 1
  private Player player2; // Human/Computer 2

  private CheckerBoard checkerBoard;

  private CheckerPiece[] player1_Pieces; // Array of player1's pieces during the game
  private CheckerPiece[] player2_Pieces; // Array of player2's pieces during the game

  public Game(Player player1, Player player2) {
    // Constructor for a new game given player1 and player2
  }

  public void choosePlayer1(String name, int type) {
    // Create a new player with name and type
  }

  public void choosePlayer2(String name, int type) {
    // Create a new player with name and type
  }

  public void startGame() {
    // Initializing a new Game
    // Calls constructor and makes the board and pieces
  }

  public void completeGame() {
    // End current Game
  }

  public void restartGame() {
    // End current Game and start a new one
  }

  @Override
  public void start(Stage stage) throws Exception {
      FXMLLoader loader = new FXMLLoader(getClass().getResource("FXML Document"));
      Parent root = loader.load();
      CheckerBoardFXMLController controller = loader.getController();

      Scene scene = new Scene(root);
      stage.setScene(scene);
      stage.show();
      controller.start(stage);
  }

  /**
   * @param args the command line arguments
   */
  public static void main(String[] args) {
      launch(args);
  }

}



public class Player {
  private int type; // 0 - Human / 1 - Computer
  private Color pColor; // light/dark players
  private String name; // User input
  private String status = "; // Winner/Loser/None

  public Player(int type, Color color) {
    // Construct a new player without a user inputed name
    // This would likely be a computer player
  }

  public Player(int type, Color color, String name) {
    // Construct a new player with a user inputed name
    // This would likely be a human player
  }

  public void winGame() {
    // Change status of player to winner
  }

  public void loseGame() {
    // Change status of player to loser
  }

  public int getType() {
    // Return player Type
  }

  public Color getPlayerColor() {
    // Return player Color
  }

  public String getPlayerName() {
    // Return player name
  }

  public String getPlayerStatus() {
    // Return player status
  }

}



public class CheckerBoard {
  private Rectangle[][] board_spaces; // This will be an array of rectangles that are the board
spaces

  private int numRows;
  private int numCols;
  private double boardWidth;
  private double boardHeight;
  private Color lightColor;
  private Color darkColor;

  private AnchorPane board = null;
  private double rectWidth;
  private double rectHeight;

  public CheckerBoard() {

  }

  public CheckerBoard(int numRows, int numCols, double boardWidth, double boardHeight) {
      this.numRows = numRows;
      this.numCols = numCols;
      this.boardWidth = boardWidth;
      this.boardHeight = boardHeight;
  }

  public CheckerBoard(int numRows, int numCols, double boardWidth, double boardHeight,
Color lightColor, Color darkColor) {
      this(numRows, numCols, boardWidth, boardHeight);
      this.lightColor = lightColor;
      this.darkColor = darkColor;
  }

  public AnchorPane build() {
    // Build out a new game board and returning it
    // Filling out the board_spaces Array
  }

  public AnchorPane getBoard() {
      return this.board;
  }

  public int getNumRows() {
      return this.numRows;
  }

  public int getNumCols() {
      return this.numCols;
  }

  public double getWidth() {
      return this.boardWidth;
  }

  public double getHeight() {
      return this.boardHeight;
  }

  public Color getLigthColor() {
      return this.lightColor;
  }

  public Color getDarkColor() {
      return this.darkColor;
  }

  public double getRectangleWidth() {
      return this.rectWidth;
  }

  public double getRectangleHeight() {
      return this.rectHeight;
  }
}
```

```
public class CheckerPiece {
  private Color color; // light/dark Black or Red pieces??
  private int x_position; // Board Column
  private int y_position; // Board Row
  private int type; // 0 - Regular / 1 - King

  public CheckerPiece(Color color, int x, int y) {
    // Constructor for a checker piece (doesn't need type because they will all start as regular
pieces)
  }

  public void movePiece(int new_x, int new_y) {
    // move piece to a new location
  }

  public void crownPiece() {
    // Upgrade piece to King status when
  }

  public void takePiece() {
    // Destroy a piece when it is overtaken
  }

  public Color getColor() {
    // Return piece Color
  }

  public int getXPosition() {
    // Return piece X position
  }

  public int getYPosition() {
    // Return piece Y position
  }

  public int getType() {
    // Return piece Type
  }
}
```