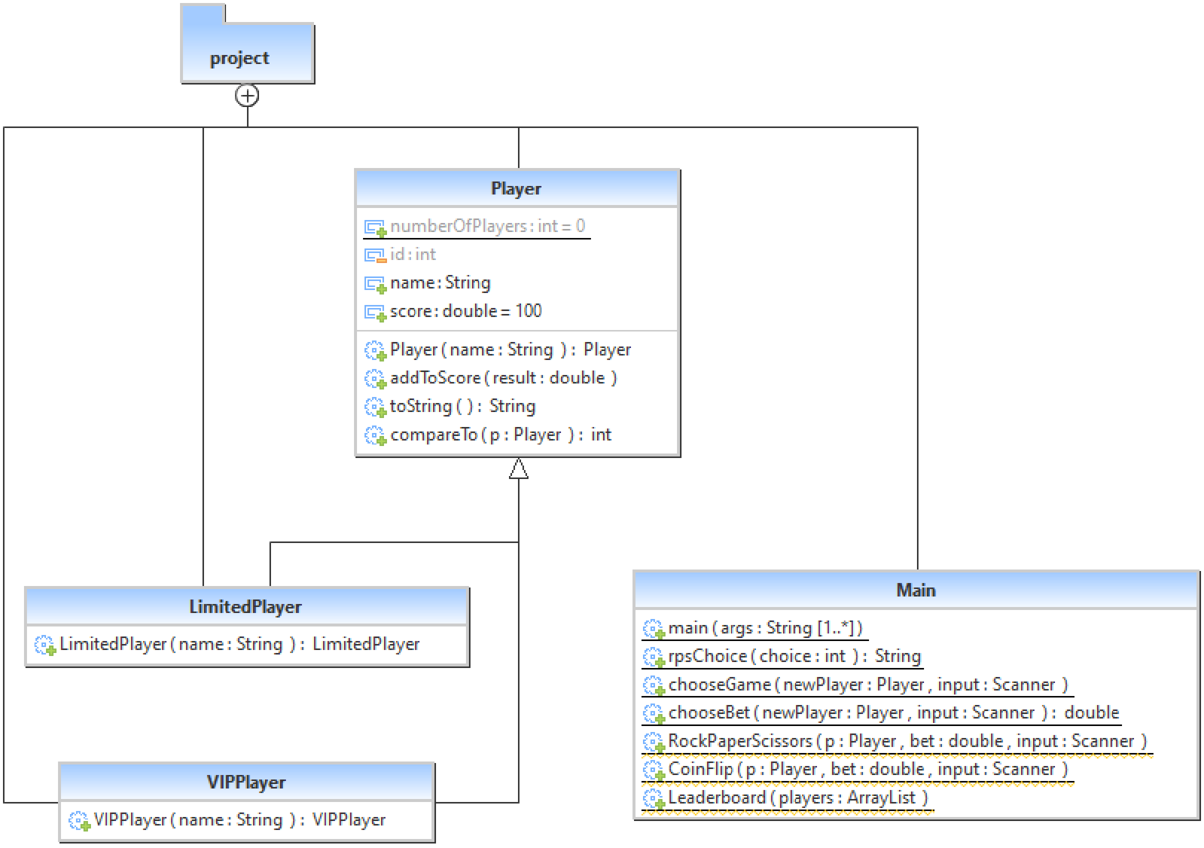
Java Project Report:

- There are 4 classes - Player, VIPPlayer, LimitedPlayer and Main. VIPPlayer and LimitedPlayer extend Player.
- VIPs get (VIP) added to their name and LimitedPlayers get (Limited) added to their name.
- Each Player has a private double called score. This is set to 100 for Players, 500 for VIP and 50 for Limited.
- Players can bet their score after selecting a game. VIPs get a 10% bonus if they win the bet, and LimitedPlayers can only bet up to 20% of their score.
- When the program starts, the user is prompted to enter 1 to create a new player or 2 to quit.
- If they enter 1, they are prompted for a name. After entering their name, they are asked what type of player they are, (VIP, Limited, None of the above). A new instance of the option they choose is made and added to the ArrayList<Player> players.
- The player then chooses to play Rock, Paper, Scissors or CoinFlip, or to quit to the main menu.
- When they choose a game, they are prompted to enter an amount to bet. If the number entered is too high or negative, an error message displays and they are prompted again.
- For Rock, Paper, Scissors, you play 3 rounds. If you win more rounds than the opponent, your bet wins, if you win the same amount of rounds the bet is returned and if you lose more rounds the bet is lost.
- For CoinFlip, if you guess 3 out of 5 coin flips correctly you win your bet, if you guess less than 3 correctly, you lose your bet.
- After a game is played, the bet is added to the Player's score using the addToScore() method. If the result was a draw, the bet amount is set to 0. If the bet was lost, the bet amount is multiplied by -1.
- When they exit the games menu they return to the main menu.
- If they quit, the leaderboard is displayed and stored in a file called "leaderboard.txt". If the file already exists, the leaderboard is appended to the file. This means the leaderboard is only saved to the file when the user quits, not continually as the user plays, so if the program terminates incorrectly, the leaderboard will not be saved. The Player class implements Comparable<Player> using the private double score, which is used to sort the ArrayList<Player> players by score for the leaderboard.
- I used a while(true) loop for the main menu and games menu, with an if statement to break out of the loop when the number to quit, -1 for the games or 2 for the main menu, is entered.


project
Player
numberOfPlayers : int = 0
id : int
name : String
score : double = 100
Player ( name : String ) : Player
addToScore ( result : double )
toString ( ) : String
compareTo ( p : Player ) : int
LimitedPlayer
LimitedPlayer ( name : String ) : LimitedPlayer
VIPPlayer
VIPPlayer ( name : String ) : VIPPlayer
Main
main ( args : String [1..*] )
rpsChoice ( choice : int ) : String
chooseGame ( newPlayer : Player , input : Scanner )
chooseBet ( newPlayer : Player , input : Scanner ) : double
RockPaperScissors ( p : Player , bet : double , input : Scanner )
CoinFlip ( p : Player , bet : double , input : Scanner )
Leaderboard ( players : ArrayList )