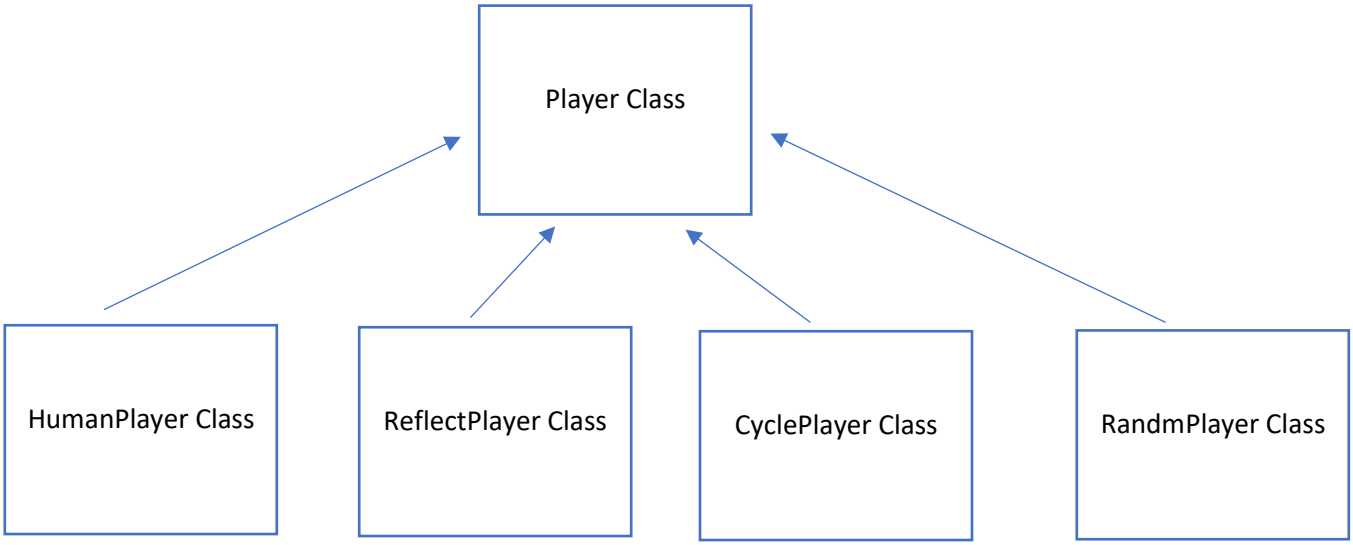
Tips for Project 2 (Rock, Paper, Scissors):

1. You have a parent class (Player), you need to implement sub classes (RandomPlayer, HumanPlayer, CyclePlayer, ReflectPlayer)



Player Class: (Already available in the starter code, and you don't need to modify it)

- Plays always "Rock"
- Function move (always **returns** "Rock")

HumanPlayer Class: (Sub class of Player class):

- Function move (**returns** a user input move)
- Don't forget to validate the input user, the available moves are only (Rock, Paper or scissors)

RandomPlayer Class: (Sub class of Player class):

- Function move (**returns** a random choice from the list moves [Rock, Paper or scissors])

CyclePlayer class: (Sub class of Player class):

- Function move:
  - If this is the first move (**returns** random or any other move)
  - If not the first move (using function **learn** it will remember what move played last round, and cycles through the different moves. (If it played 'rock' this round, it will **return** 'paper' in the next round.)
- Function learn:
  - Will keep track (save) of the moves of both players

ReflectPlayer Class: (Sub class Player class):

- Function move:
  - If this is the first move (**returns** random or any other move)
  - If not the first move (using function **learn** it will remember what the opponent last move and plays that move *this* round) if you play 'paper' on the first round, a ReflectPlayer will play 'paper' on the second round.)
- Function learn:
  - Will keep track (save) of the moves of both players

2. Class Game:

   The players' score and the number of rounds played, should be stored as instance variables.

   - play_round function:
     - Use function beats to check who is the winner.
     - Don't forget to handle the **Tie**
     - Use players' score to keep track of the winner in each round to calculate the winner of the game
     - Announce the winner of this round.
   - play_game function:
     - The number of rounds per game, as well as when to stop, are up to you, you can also use 'quit' to terminate the game
     - Using the players' score announce the winner.

3. if __name__ == '__main__':

    - In this part You should display a welcome message.
    - Input the type of player the user wants to play with [Random, Reflect, Repeat (always plays rock), or Cycle]
    - Don’t forget the input validation.
    - For each choice you will create an if condition to handle the user choice [Random, Reflect, Repeat (always plays rock), or Cycle]
      then and starts the game with Humanplayer and the user choice.

      For example, if the user choose “Random” you will start the game as following
      game = Game (HumanPlayer(), RandomPlayer())