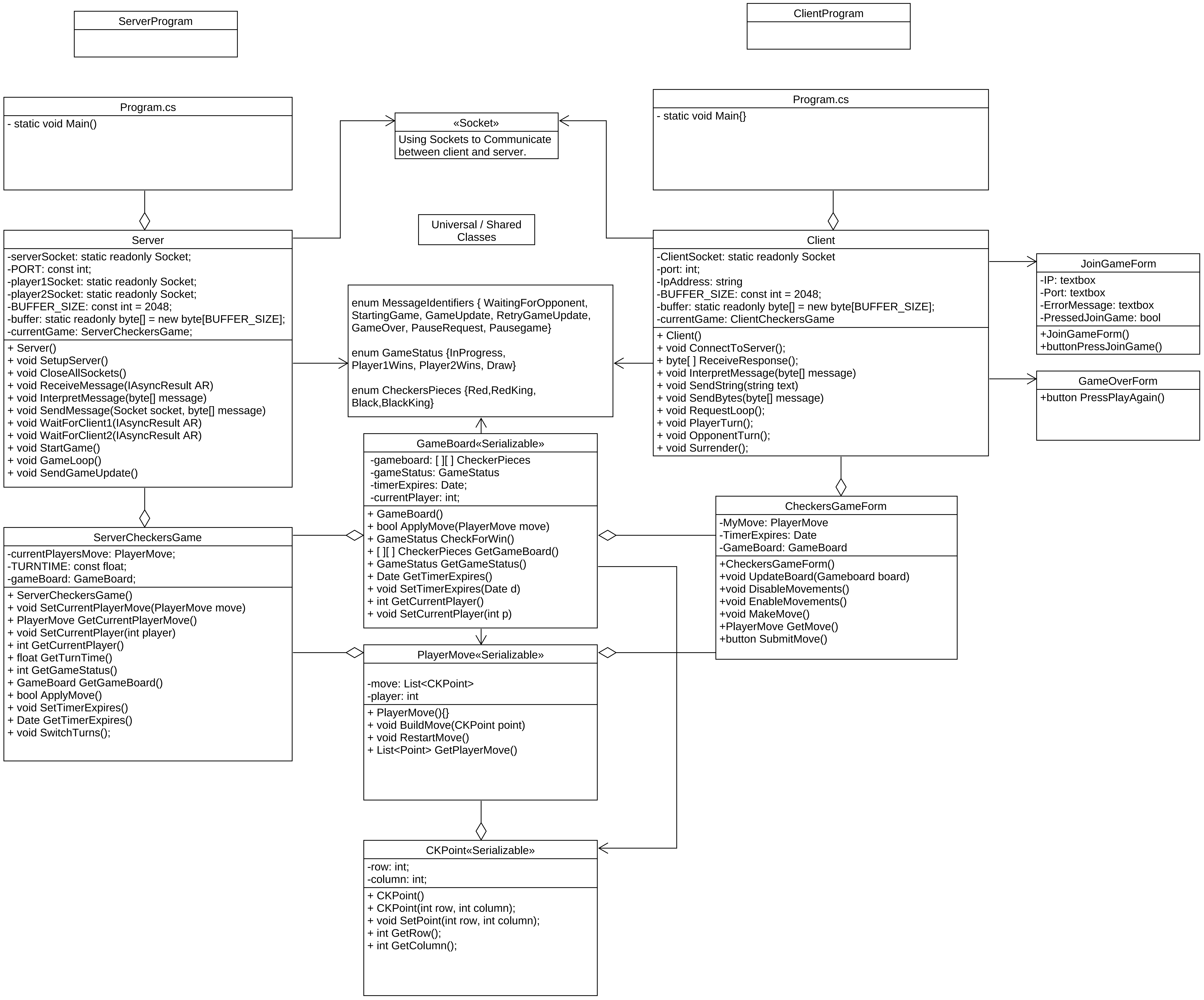

```
ServerProgram
```

```
ClientProgram
```

```
Program.cs
- static void Main()
```

```
Server
-serverSocket: static readonly Socket;
-PORT: const int;
-player1Socket: static readonly Socket;
-player2Socket: static readonly Socket;
-BUFFER_SIZE: const int = 2048;
-buffer: static readonly byte[] = new byte[BUFFER_SIZE];
-currentGame: ServerCheckersGame;
-----
+ Server()
+ void SetupServer()
+ void CloseAllSockets()
+ void ReceiveMessage(IAsyncResult AR)
+ void InterpretMessage(byte[] message)
+ void SendMessage(Socket socket, byte[] message)
+ void WaitForClient1(IAsyncResult AR)
+ void WaitForClient2(IAsyncResult AR)
+ void StartGame()
+ void GameLoop()
+ void SendGameUpdate()
```

```
ServerCheckersGame
-currentPlayersMove: PlayerMove;
-TURNTIME: const float;
-gameBoard: GameBoard;
-----
+ ServerCheckersGame()
+ void SetCurrentPlayerMove(PlayerMove move)
+ PlayerMove GetCurrentPlayerMove()
+ void SetCurrentPlayer(int player)
+ int GetCurrentPlayer()
+ float GetTurnTime()
+ int GetGameStatus()
+ GameBoard GetGameBoard()
+ bool ApplyMove()
+ void SetTimerExpires()
+ Date GetTimerExpires()
+ void SwitchTurns();
```

```
«Socket»
Using Sockets to Communicate
between client and server.
```

```
Universal / Shared
Classes
```

```
enum MessageIdentifiers { WaitingForOpponent,
StartingGame, GameUpdate, RetryGameUpdate,
GameOver, PauseRequest, Pausegame}

enum GameStatus {InProgress,
Player1Wins, Player2Wins, Draw}

enum CheckersPieces {Red,RedKing,
Black,BlackKing}
```

```
GameBoard«Serializable»
-gameboard: [ ][ ] CheckerPieces
-gameStatus: GameStatus
-timerExpires: Date;
-currentPlayer: int;
-----
+ GameBoard()
+ bool ApplyMove(PlayerMove move)
+ GameStatus CheckForWin()
+ [ ][ ] CheckerPieces GetGameBoard()
+ GameStatus GetGameStatus()
+ Date GetTimerExpires()
+ void SetTimerExpires(Date d)
+ int GetCurrentPlayer()
+ void SetCurrentPlayer(int p)
```

```
PlayerMove«Serializable»

-move: List<CKPoint>
-player: int
-----
+ PlayerMove(){}
+ void BuildMove(CKPoint point)
+ void RestartMove()
+ List<Point> GetPlayerMove()
```

```
CKPoint«Serializable»
-row: int;
-column: int;
-----
+ CKPoint()
+ CKPoint(int row, int column);
+ void SetPoint(int row, int column);
+ int GetRow();
+ int GetColumn();
```

```
Program.cs
- static void Main{}
```

```
Client
-ClientSocket: static readonly Socket
-port: int;
-IpAddress: string
-BUFFER_SIZE: const int = 2048;
-buffer: static readonly byte[] = new byte[BUFFER_SIZE];
-currentGame: ClientCheckersGame
-----
+ Client()
+ void ConnectToServer();
+ byte[ ] ReceiveResponse();
+ void InterpretMessage(byte[] message)
+ void SendString(string text)
+ void SendBytes(byte[] message)
+ void RequestLoop();
+ void PlayerTurn();
+ void OpponentTurn();
+ void Surrender();
```

```
JoinGameForm
-IP: textbox
-Port: textbox
-ErrorMessage: textbox
-PressedJoinGame: bool
-----
+JoinGameForm()
+buttonPressJoinGame()
```

```
GameOverForm
+button PressPlayAgain()
```

```
CheckersGameForm
-MyMove: PlayerMove
-TimerExpires: Date
-GameBoard: GameBoard
-----
+CheckersGameForm()
+void UpdateBoard(Gameboard board)
+void DisableMovements()
+void EnableMovements()
+void MakeMove()
+PlayerMove GetMove()
+button SubmitMove()
```