# Assignment 3: TicTacToe

A short tutorial

Dr. Ziad Kobti

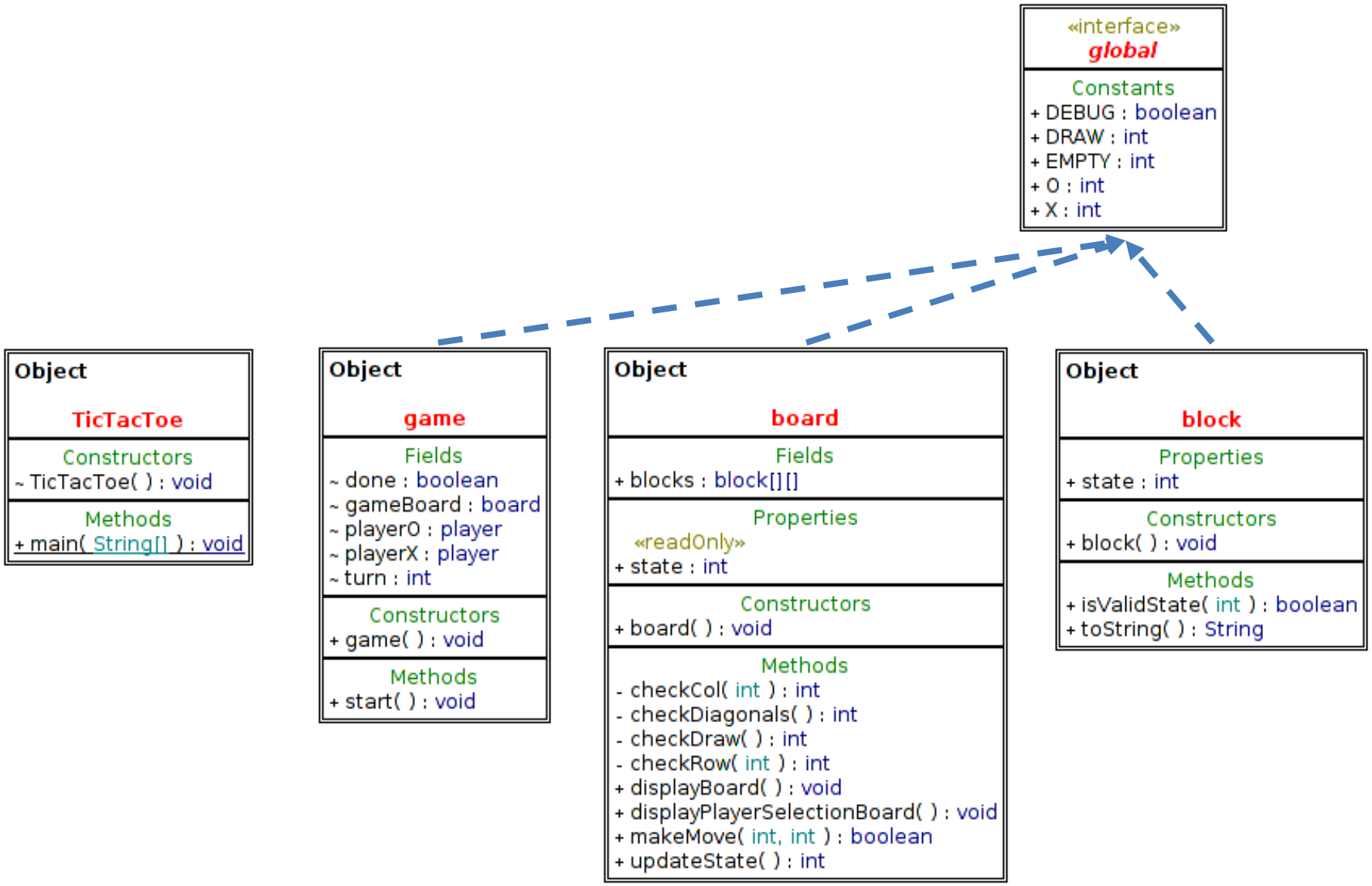
# TicTacToe… The game

- I am sure you have played TicTacToe before, you have experienced it, so we know the following facts:
  - The game requires 2 players
  - The game has a game board
  - The game board starts with 9 empty blocks or cells organized in a 3 by 3 grid
  - Each player takes a symbol, usually X and O
  - The game ends either by one player winning or we fill up the board without a winning move, that would be a draw.
  - Players take turn, no player can take 2 consecutive turns.
  - A player can only move onto an empty block
  - The game is won when a player holds three consecutive blocks, either horizontally, vertically or diagonally.

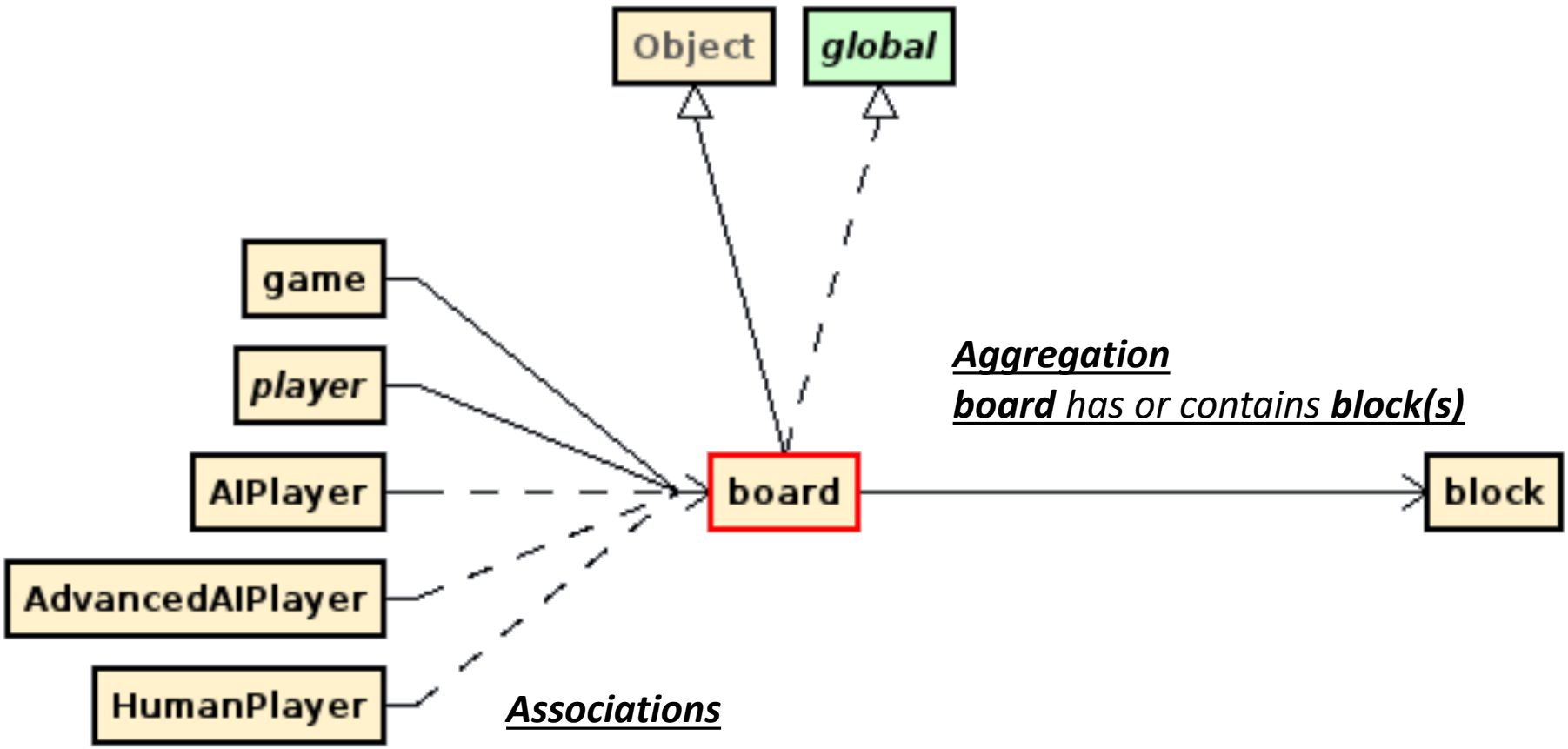
# Can we design a structure and assign responsibilities to objects – who does what?

- Well we need a **board**, and a board has 9 **blocks**
- We need a **game** instance that manages the game play (coin toss to start, turns, declare winner, etc)
- We need 2 players in the game.

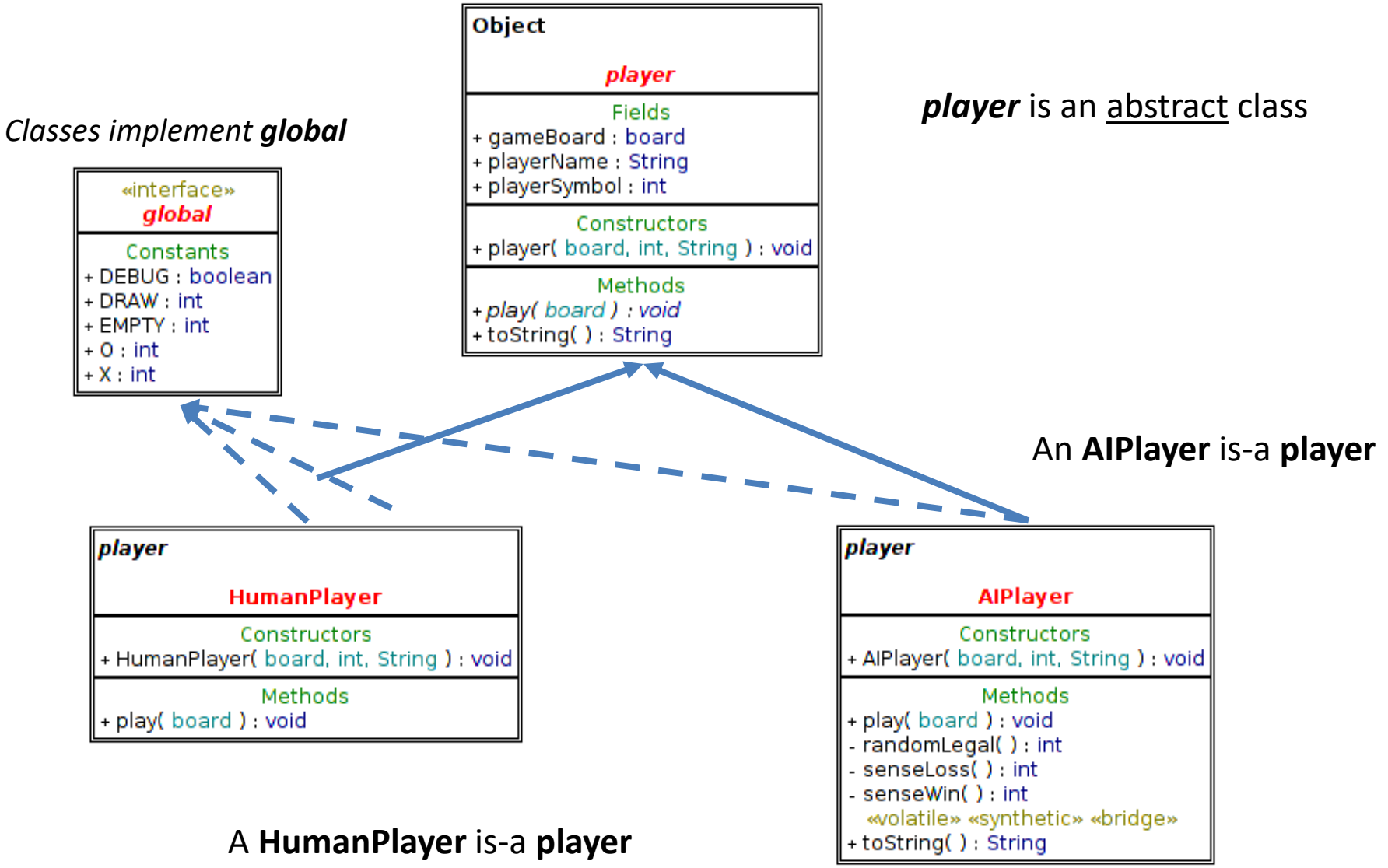
# Sample UML used in the demo



# Association and Aggregation relations



# Sample UML used in the Demo




*Classes implement* ***global***

***player*** is an abstract class

An **AIPlayer** is-a **player**

A **HumanPlayer** is-a **player**

# Design Patterns

- How do you know which object does what?
- Design patterns in Object Oriented Design are common practices by software architects who define these named patterns in terms of problem-solution pairs.

Example:

Problem: Who creates “block” instances?

Solution: “board” because it contains or uses the blocks.

Design pattern name applied here: **creator**

This is why the board class instantiates block instances.

Can you tell what pattern I used to create extensible player types?

Here is a good reference:

https://www.cs.colorado.edu/~kena/classes/5448/f12/presentation-materials/rao.pdf

# main?

```
class TicTacToe
{
        public static void main(String args[])
        {
                game myGame = new game();
                myGame.start();
        }

}
```

Just start the game!

# You can try it…

- I provided the demo in a jar file
- A jar file is simply a java archive containing a bunch of files which make it easy for distribution.
- Follow the instructions given in the link in the assignment folder to run it.

- Good luck!