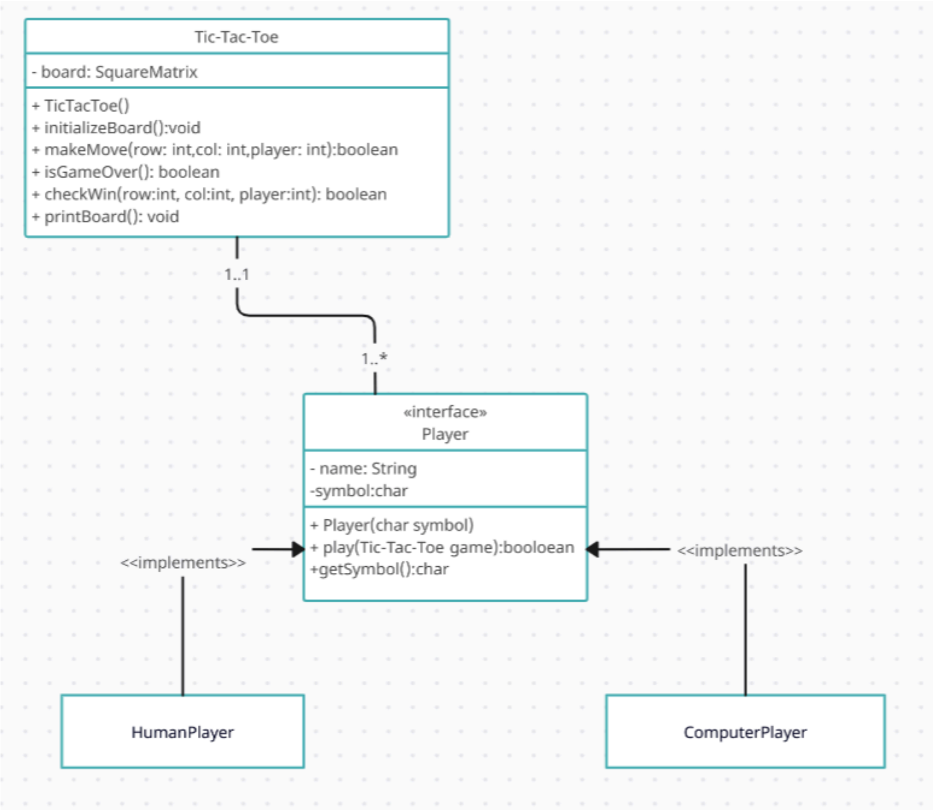
23-813-0207: Lab 4 - Java Programming Lab

Lab exercise - 3

Date: 21/2/2024



1. Modify the makeMove() method in the Tic-Tac-Toc class to handle conditions where the values of rows and columns are not within the limit. If a player tries an invalid move outside the board boundaries, the method should throw an **IllegalArgumentException** with a clear error message. The Main class should catch this exception and handle it by printing an error message and continuing the game with the correct turn.
2. Write your own Exception class **InvalidMoveException** and use it instead of **IllegalArgumentException** in the above example.
3. Create an interface *Player* as shown in the figure. The play() method for HumanPlayer should display the board and allow the user to enter his move. The *play*() method in ComputerPlayer shall scan the board for empty cells and randomly choose one to play. The play() method should use the already existing makeMove() method to make a move. *[Optionally you can create a subclass AIPlayer to the class ComputerPlayer and implement any intelligent play() method]*

4. Write a Java program to create an **abstract** class *Employee* with abstract methods *calculateSalary*() and *displayInfo*(). Create subclasses Manager and Programmer that extend the *Employee* class and implement the respective methods to calculate salary and display information for each role.
    a. Demonstrate these methods by creating one object of each subclass.
    b. Also, create a listEmployees(Employee[] emp) method in your Main class, that displays the Name and Salary of every employee as a table. Is it possible to pass an array containing both Managers and Programmers to this method?
5. Create an **interface** called *PaymentMethod* with abstract methods *pay*() and *cancel*(). Create classes CreditCard and PayPal that implement the *PaymentMethod* interface and provide their implementations of the pay() and cancel() methods. Demonstrate their working with appropriate data.