**ACKNOWLEDGEMENTS**

We would like to express our heartfelt appreciation to Ms. Swati Sheoran, our Project Supervisor at SRM Institute of Science and Technology, Delhi-NCR Campus, Modinagar, for her invaluable insights and expertise in the subject matter, which motivated us to work diligently.

We would like to extend our sincere thanks to Dr. S. Vishwanathan, Director of SRM Institute of Science and Technology, Delhi-NCR Campus, Modinagar, for his unwavering support that enabled us to undertake and complete our project work. Our special thanks go to Dr. R. P. Mahapatra, Dean Computer Science and Engineering at SRM Institute of Science and Technology, Delhi-NCR Campus, Modinagar, for his valuable guidance and unconditional support.

We would like to express our gratitude to **Avneesh Vashisht**, HOD CSE 2nd year at SRM Institute of Science and Technology, Delhi-NCR Campus, Modinagar, for his suggestions and encouragement in completing this project.

Finally, we would like to express our sincere appreciation to our parents, family members, and friends for their unwavering support and encouragement, and to all our well-wishers.

Lakshya Chaudhary ---> RA2311026030064

Vidhan Veerbhan --> RA2311026030066

Sarthak Aggarwal --> RA2311026030118

Nirupam Saini --> RA2311026030119

**DECLARATION**

We,

Lakshya Chaudhary [Reg.No.:RA2311026030064]

Vidhan Veerbhan[Reg.No.:RA2311026030066]

Sarthak Aggarwal [Reg.No.:RA2311026030118]

Nirupam Saini [Reg.No.:RA2311026030119]

hereby declare that the work which is being presented in the project report” Tic-Tac-Toe Game” is the record of authentic work carried out by us during the period in October’24 and submitted by us in partial fulfilment for the award of the degree “Bachelor of Technology in Computer Science and Engineering” to SRM IST, NCR Campus, Ghaziabad (U.P.). This work has not been submitted to another University or Institute for the award of any Degree/Diploma.

**INTRODUCTION**

A two-player game (Player 1 and Player 2).

Players take turns to place their mark (Player 1 uses 'X', Player 2 uses 'O') on a 3x3 grid.

The objective is to align 3 marks horizontally, vertically, or diagonally.

Game Setup:

A 3x3 grid represented as a 2D array or a list.

Players choose positions (1-9 or coordinate-based) to place their marks.

Win Condition:

A player wins by forming a straight line (horizontal, vertical, or diagonal).

If all 9 positions are filled without a winner, the game is a draw.

Turn-based Gameplay:

Players take turns sequentially.

Input validation ensures the chosen cell is empty before placing the mark.

Game Flow:

Initialize an empty grid.

Display the grid after every move.

Check for a win or a draw after each turn.

Announce the winner or if it's a draw.

Main Components in Java:

A 2D array to store the grid state.

Loops for input and validation.

Conditional checks for win or draw.

Methods to handle grid display, move validation, and result checking.

Key Methods to Implement:

displayBoard(): to print the current grid.

makeMove(int player, int row, int col): to allow a player to place their mark.

checkWin(): to check for a winning condition.

checkDraw(): to check if the grid is full and no winner is found.

This outline gives you the basic structure to implement a Tic-Tac-Toe game in Java.

**HARDWARE REQUIREMENTS**

• Processor: Any modern processor with at least 1 GHz speed.

• RAM: 512 MB RAM or higher

• Graphics: Integrated graphics card or dedicated graphics card with basic capabilities.

• Storage: A few megabytes of available storage space for the game installation.

• Operating System: Compatible with Windows

• Display: Any monitor or screen with a resolution of 800x600 pixels or higher.

• Sound: Basic sound card for in-game sounds and effects (optional).

• Input Devices: Keyboard for user input during gameplay and development.

**SOFTWARE REQUIREMENTS**

* Java Programming Language
* Database Management System
* Web Browser.

**ADVANTAGES**

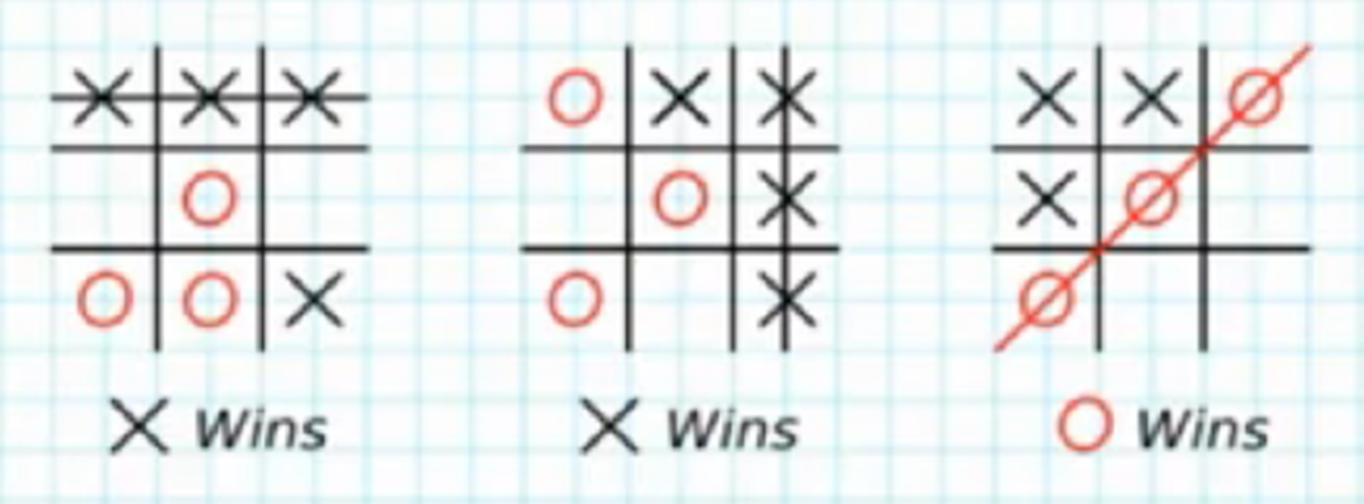
* Here are some advantages of the Java programming language presented in point form:
* Platform Independence: Java's "write once, run anywhere" capability allows applications to run on any device with a Java Virtual Machine (JVM), promoting portability.
* Object-Oriented: Java supports object-oriented programming principles, making code reusable, modular, and easier to maintain.
* Rich Standard Library: Java offers a comprehensive standard library that provides a wide range of built-in classes and methods for various tasks, facilitating faster development.
* Automatic Memory Management: Java's garbage collection process automatically manages memory allocation and deallocation, reducing the risk of memory leaks and other related issues.
* Strongly Typed Language: Java enforces strict type-checking at both compile-time and runtime, which helps catch errors early in the development process.
* Multithreading Support: Java provides built-in support for multithreading, enabling concurrent execution of tasks and improving the performance of applications.
* Robust Security Features: Java offers a secure environment through its extensive API and security features, such as the Java security manager and bytecode verification.
* Large Community and Ecosystem: A vast community of developers supports Java, ensuring a wealth of resources, libraries, and frameworks that can expedite development.
* Strong IDE Support: Java is compatible with various Integrated Development Environments (IDEs) like Eclipse and IntelliJ IDEA, which enhance productivity with tools for debugging, testing, and code management.
* Wide Adoption in Enterprises: Java is extensively used in enterprise environments for building large-scale applications, making it a valuable skill for developers.

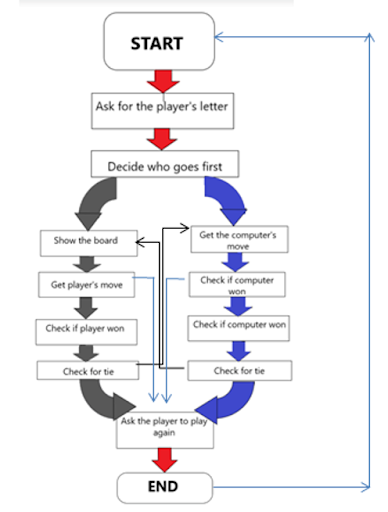
**FLOWCHART**

Here’s a flowchart representing the basic structure of a Tic-Tac-Toe game in Java. The flowchart outlines key steps such as initializing the board, checking for a winner, and alternating turns between two players.

**Explanation of Flowchart:**

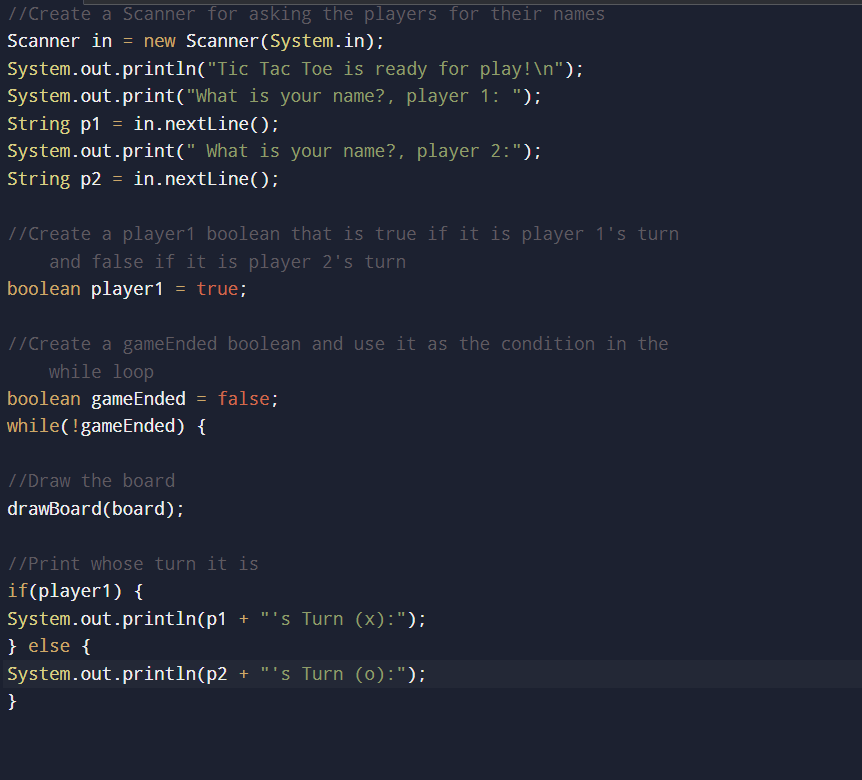
1. **Start the Game**: Initiate the game setup.
2. **Initialize the Board**: Create a 3x3 board to represent the Tic-Tac-Toe grid.
3. **Display the Board**: Show the current state of the board.
4. **Player 1 Turn**: Player 1 chooses their move.
5. **Check if Move is Valid**: Ensure the chosen cell is empty and valid.
6. **Update Board**: Place the player’s move on the board if valid.
7. **Check for Winner or Tie**: After each move, check if there's a winner or if the board is full (tie).
8. **Declare Winner**: If Player 1 or Player 2 wins, announce the winner and end the game.
9. **Alternate Player Turns**: If no winner, continue the game, alternating between Player 1 and Player 2.

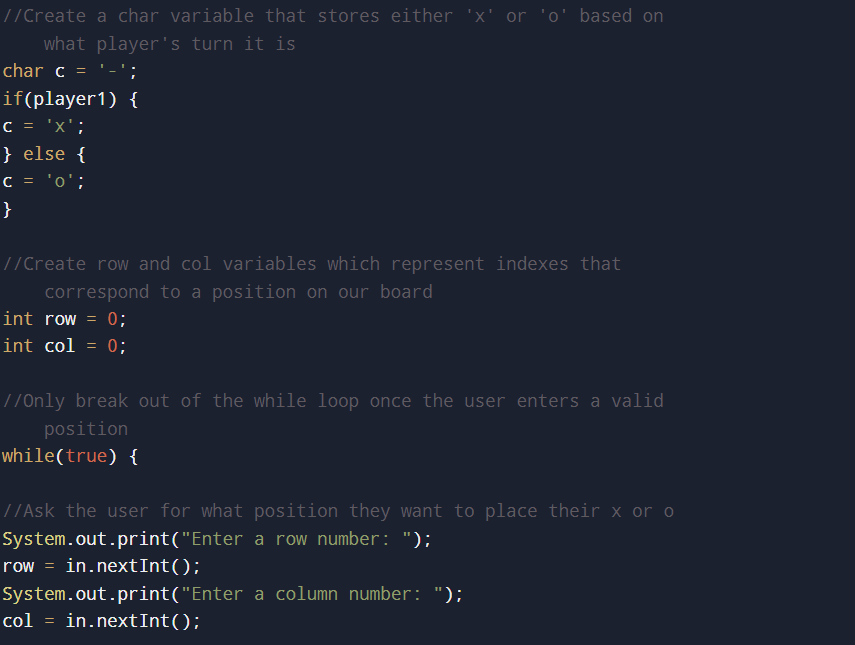


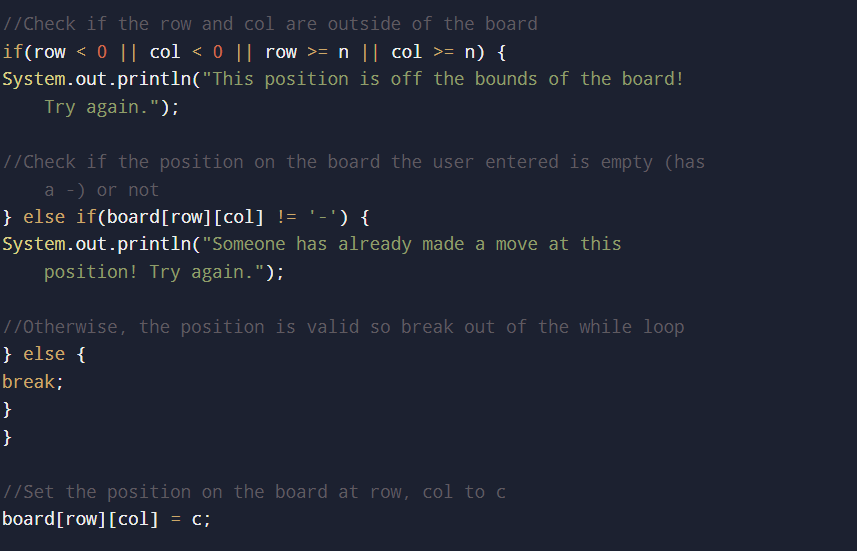


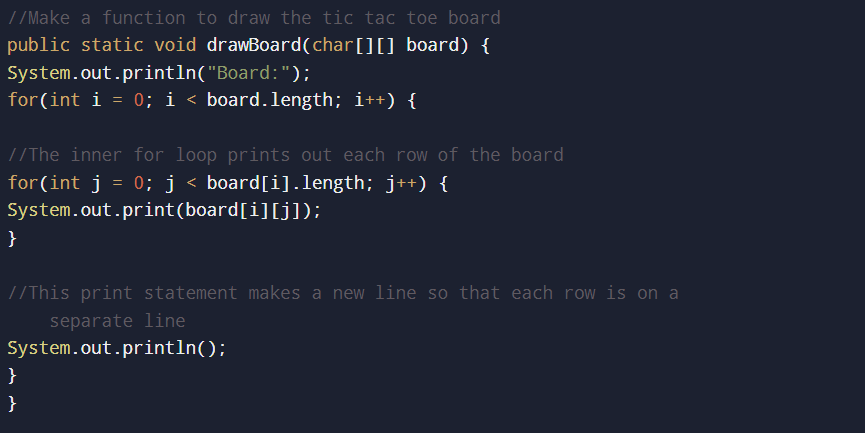
**CODE FOR TIC-TAC-TOE GAME IN JAVA**

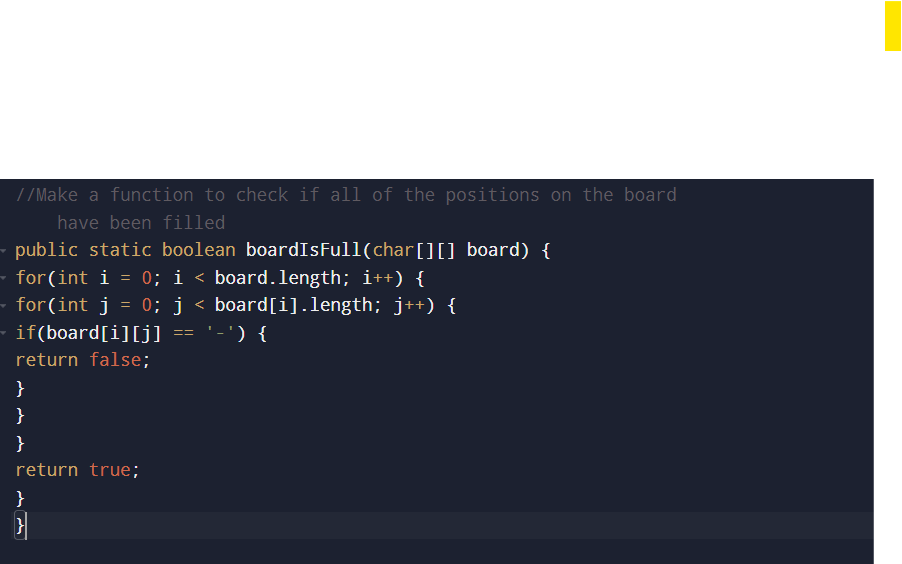












**OUTPUT:-**

