



Online Java Compiler IDE

For Multiple Files, Custom Library and File Read/Write, use our new - [Advanced Java IDE](#)

```

1  import java.util.Arrays;
2  import java.util.Scanner;
3  import java.util.stream.Collectors;
4  import java.util.stream.IntStream;
5
6  // we are going to create a simple 2-players Connect Four implementation in Java 8
7  public class A0Star_ConnectFour
8  {
9
10     // we define characters for players (R for Red, Y for Yellow)
11     private static final char[] PLAYERS = {'R', 'Y'};
12     // dimensions for our board
13     private final int width, height;
14     // grid for the board
15     private final char[][] grid;
16     // we store last move made by a player
17     private int lastCol = -1, lastTop = -1;
18
19     public A0Star_ConnectFour(int w, int h) {
20         width = w;
21         height = h;
22         grid = new char[h][];
23
24         // init the grid will blank cell
25         for (int i = 0; i < h; i++) {
26             Arrays.fill(grid[i] = new char[w], '.');
27         }
28     }
29
30     // we use Streams to make a more concise method
31     // for representing the board
32     @Override
33     public String toString() {
34         return IntStream.range(0, width).
35             mapToObj(Integer::toString).
36             collect(Collectors.joining()) +
37             "\n" +
38             Arrays.stream(grid).
39             map(String::new).
40             collect(Collectors.joining("\n"));
41     }
42
43     // get string representation of the row containing
44     // the last play of the user
45     public String horizontal() {
46         return new String(grid[lastTop]);
47     }
48
49     // get string representation fo the col containing
50     // the last play of the user
51     public String vertical() {
52         StringBuilder sb = new StringBuilder(height);
53
54         for (int h = 0; h < height; h++) {
55             sb.append(grid[h][lastCol]);
56         }
57
58         return sb.toString();
59     }
60

```

```

61 // get string representation of the "/" diagonal
62 // containing the last play of the user
63 public String slashDiagonal() {
64     StringBuilder sb = new StringBuilder(height);
65
66     for (int h = 0; h < height; h++) {
67         int w = lastCol + lastTop - h;
68
69         if (0 <= w && w < width) {
70             sb.append(grid[h][w]);
71         }
72     }
73
74     return sb.toString();
75 }
76
77 // get string representation of the "\"
78 // diagonal containing the last play of the user
79 public String backslashDiagonal() {
80     StringBuilder sb = new StringBuilder(height);
81
82     for (int h = 0; h < height; h++) {
83         int w = lastCol - lastTop + h;
84
85         if (0 <= w && w < width) {
86             sb.append(grid[h][w]);
87         }
88     }
89
90     return sb.toString();
91 }
92
93 // static method checking if a substring is in str
94 public static boolean contains(String str, String substring) {
95     return str.indexOf(substring) >= 0;
96 }
97
98 // now, we create a method checking if last play is a winning play
99 public boolean isWinningPlay() {
100     if (lastCol == -1) {
101         System.err.println("No move has been made yet");
102         return false;
103     }
104
105     char sym = grid[lastTop][lastCol];
106     // winning streak with the last play symbol
107     String streak = String.format("%c%c%c%c", sym, sym, sym, sym);
108
109     // check if streak is in row, col,
110     // diagonal or backslash diagonal
111     return contains(horizontal(), streak) ||
112            contains(vertical(), streak) ||
113            contains(slashDiagonal(), streak) ||
114            contains(backslashDiagonal(), streak);
115 }
116
117 // prompts the user for a column, repeating until a valid choice is made
118 public void chooseAndDrop(char symbol, Scanner input) {
119     do {
120         System.out.println("\nPlayer " + symbol + " turn: ");
121         int col = input.nextInt();
122
123         // check if column is ok
124         if (!(0 <= col && col < width)) {
125             System.out.println("Column must be between 0 and " + (width - 1));
126             continue;
127         }
128
129         // now we can place the symbol to the first
130         // available row in the asked column

```

```

130 // available row in the given column
131 for (int h = height - 1; h >= 0; h--) {
132     if (grid[h][col] == '.') {
133         grid[lastTop = h][lastCol = col] = symbol;
134         return;
135     }
136 }
137
138 // if column is full ==> we need to ask for a new input
139 System.out.println("Column " + col + " is full.");
140 } while (true);
141 }
142
143 public static void main(String[] args) {
144     // we assemble all the pieces of the puzzle for
145     // building our Connect Four Game
146     try (Scanner input = new Scanner(System.in)) {
147         // we define some variables for our game like
148         // dimensions and nb max of moves
149         int height = 6; int width = 8; int moves = height * width;
150
151         // we create the AOStar_ConnectFour instance
152         AOStar_ConnectFour board = new AOStar_ConnectFour(width, height);
153
154         // we explain users how to enter their choices
155         System.out.println("Use 0-" + (width - 1) + " to choose a column");
156         // we display initial board
157         System.out.println(board);
158
159         // we iterate until max nb moves be reached
160         // simple trick to change player turn at each iteration
161         for (int player = 0; moves-- > 0; player = 1 - player) {
162             // symbol for current player
163             char symbol = PLAYERS[player];
164
165             // we ask user to choose a column
166             board.chooseAndDrop(symbol, input);
167
168             // we display the board
169             System.out.println(board);
170
171             // we need to check if a player won. If not,
172             // we continue, otherwise, we display a message
173             if (board.isWinningPlay()) {
174                 System.out.println("\nPlayer " + symbol + " wins!");
175                 return;
176             }
177         }
178
179         System.out.println("Game over. No winner. Try again!");
180     }
181 }
182 }

```

Execute Mode, Version, Inputs & Arguments

CommandLine Arguments

Result

compiled and executed in 84.404 sec(s)

```

Use 0-7 to choose a column
01234567
.....
.....
.....

```

```
.....
.....
.....
.....

Player R turn:
4
01234567
.....
.....
.....
.....
.....
.....
....R...
```

```
Player Y turn:
3
01234567
.....
.....
.....
.....
.....
....YR...
```

```
Player R turn:
4
01234567
.....
.....
.....
.....
....R...
...YR...
```

```
Player Y turn:
2
01234567
.....
.....
.....
.....
....R...
..YYR...
```

```
Player R turn:
4
01234567
.....
.....
.....
....R...
....R...
..YYR...
```


```
Player Y turn:
2
01234567
.....
.....
.....
....R...
..Y.R...
..YYR...
```

```
Player R turn:
4
01234567
.....
.....
....R...
```

```
.....  
....R...  
..Y.R...  
..YYR...
```

Player R wins!

Note:

1. For file operations - upload files using upload button . Files will be upload to /uploads folder. You can read those files in program from /uploads folder. To write a file from your program, write files to '/myfiles' folder. Please note the uploaded files stored in the server only for the current session.
2. For detailed documentation check - [Our Documentation](#), or check our [Youtube channel](#).

Thanks for using our
Online Java Compiler IDE
to execute your program



Know Your JDoodle

- JDoodle Supports 76+ Languages with Multiple Versions and 2 DBs. [Click here](#) to see all.
- Fullscreen - side-by-side code and output is available. click the "☐" icon near execute button to switch.
- Dark Theme available. Click on "⋮" icon near execute button and select dark theme.
- You can embed code from JDoodle directly into your website/blog. [Click here](#) to know more.
- JDoodle offers an API service. You can execute programs just by calling our API.

JDoodle For Your Organisation

- Do you have any specific compiler requirements?
- Do you want to integrate compilers with your website, webapp, mobile app, courses?
- Do you need more than our [Embed](#) and [API](#) features?
- Looking for Multiple Files, Connecting to DB, Debugging, etc.?
- Are you building any innovative solution for your students or recruitment?
- Want to run JDoodle in-house?
- Custom Domain, White labelled pages for your institute?

[Click here](#) to know more.

● If you like JDoodle, Please share us in Social Media. [Click here](#) to share.

● Check our [Documentation Page](#) for more info.

JDoodle is serving the programming community since 2013

Contact us - We are happy to help!