

## Folder Q2

3 printable files

(file list disabled)

Q2\Game.java

```
1 package quest6.Q2;
2
3 public enum Game {
4     X,O,EMPTY
5 }
6
```

Q2\GameTest.java

```
1 package quest6.Q2;
2
3 import java.util.Scanner;
4
5 public class GameTest {
6     public static void main(String[] args) {
7         Scanner entrada = new Scanner(System.in);
8         TicTacToe jogo = new TicTacToe();
9         int linha, coluna;
10
11         while (!jogo.acabou) {
12             System.out.println("Vez do " + jogo.jogador + "° jogador");
13             System.out.println("Escolha a linha que você quer jogar: ");
14             linha = entrada.nextInt();
15             System.out.println("Escolha a coluna que você quer jogar: ");
16             coluna = entrada.nextInt();
17             if(jogo.fazerJogada(linha - 1, coluna - 1)) {
18                 System.out.println(jogo.printaJogo());
19             } else {
20                 System.out.println("Espaço já jogado tente outro");
21             }
22         }
23         if(jogo.vencedor != 0) {
24             System.out.println("Parabéns jogador nº " + jogo.vencedor + " você venceu!!!");
25         }
26     }
27 }
28
29
```

Q2\TicTacToe.java

```
1 package quest6.Q2;
2
3 import java.util.Arrays;
4
5 public class TicTacToe {
```

```
6      public Game game[][] = new Game[3][3];
7      public int jogador = 1;
8      public int vencedor = 0;
9      public boolean velha;
10     boolean acabou = false;
11
12     public TicTacToe() {
13         for(int i = 0; i < 3; i++){
14             for(int j = 0; j < 3; j++) {
15                 game[i][j] = Game.EMPTY;
16             }
17         }
18     }
19
20     public boolean espacoOcupado(int linha, int coluna) {
21         if(game[linha][coluna] != Game.EMPTY) {
22             return true;
23         } else {
24             return false;
25         }
26     }
27
28     public boolean fazerJogada(int linha, int coluna) {
29         if(espacoOcupado(linha, coluna)) {
30             return false;
31         }
32         if(jogador == 1) {
33             game[linha][coluna] = Game.X;
34             jogador = 2;
35         } else {
36             game[linha][coluna] = Game.O;
37             jogador = 1;
38         }
39         acabou = verificaAcabou();
40         return true;
41     }
42
43     public String printaJogo() {
44         String jogoAtual = "";
45         for(int i = 0; i < 3; i++) {
46             for(int j = 0; j < 3; j++) {
47                 if(game[i][j] == Game.EMPTY) {
48                     jogoAtual += "_";
49                 } else {
50                     jogoAtual += game[i][j];
51                 }
52                 if(j == 0 || j == 1) {
53                     jogoAtual += '|';
54                 }
55             }
56             jogoAtual += '\n';
57         }
58         return jogoAtual;
59     }
60
61     public boolean verificaAcabou() {
62         boolean velha = false;
63         Game teste[] = new Game[3];
64
65     }
```

```
66 //verifica linhas
67 for(int i = 0; i < 3; i++) {
68     if(Arrays.stream(game[i]).allMatch(esp -> (esp == Game.X))) {
69         vencedor = 1;
70         return true;
71     } else if(Arrays.stream(game[i]).allMatch(esp -> (esp == Game.O))) {
72         vencedor = 2;
73         return true;
74     }
75 }
76
77 //verifica colunas
78 for(int i = 0; i < 3; i++) {
79     for(int j = 0; j < 3; j++) {
80         teste[j] = game[j][i];
81     }
82     if(Arrays.stream(teste).allMatch(esp -> (esp == Game.X)) ||
Arrays.stream(teste).allMatch(esp -> (esp == Game.O))) {
83         return true;
84     }
85 }
86
87 //verifica diagonal principal
88 for(int i = 0; i < 3; i++) {
89     teste[i] = game[i][i];
90 }
91 if(Arrays.stream(teste).allMatch(esp -> (esp == Game.X)) || Arrays.stream(teste)
.allMatch(esp -> (esp == Game.O))) {
92     return true;
93 }
94
95 //verifica diagonal secundaria
96 for(int i = 0; i < 3; i++) {
97     int j = 2;
98     switch(i) {
99         case 0:
100             j = 2;
101             break;
102         case 1:
103             j = 1;
104             break;
105         case 2:
106             j = 0;
107             break;
108     }
109     teste[i] = game[i][j];
110 }
111 if(Arrays.stream(teste).allMatch(esp -> (esp == Game.X)) || Arrays.stream(teste)
.allMatch(esp -> (esp == Game.O))) {
112     return true;
113 }
114
115 //verifica se deu velha
116 velha = true;
117 for(int i = 0; i < 3; i++) {
118     for(int j = 0; j < 3; j++) {
119         if(game[i][j] == Game.EMPTY) {
120             velha = false;
121         }
122     }
123 }
```

```
124     if(velha) {  
125         System.out.println("Deu VELHA!");  
126         return true;  
127     }  
128  
129     return false;  
130 }  
131  
132  
133 }  
134
```