

# Programmazione 2 e Lab. di programmazione 2

*Corso di Laurea in Informatica - Anno Accademico 2022-23*

## **Docenti**

Prof. Angelo Ciaramella

`[angelo.ciaramella@uniparthenope.it]`

Prof. Luigi Catuogno

`[luigi.catuogno@uniparthenope.it]`

## **Tutor**

Dott. Antonio Vanzanella

`[antonio.vanzanella@studenti.uniparthenope.it]`

1

Esercizi svolti

2

## Esercizio: *tic-tac-toe* #1

Scrivere una classe `tttPlayGround` che abbia, tra gli altri, i seguenti attributi privati:

```
int playground[3][3];
int prossimoG;
```

3

## Esercizio: *tic-tac-toe* #1

La classe `tttPlayGround` presenta la seguente interfaccia:

```
tttPlayGround()
```

il costruttore di default. Azzera il contenuto di `playground` e imposta `prossimoG=1`;

```
void show()
```

visualizza il contenuto di `playground` in modo che, le celle contenenti 0 appaiano vuote, quelle a 1 riportino il carattere `O` e quelle con 2 il carattere `X`

4

## Esercizio: *tic-tac-toe* #1

```
char prossimo()
```

restituisce il il carattere **O** se **prossimoG** vale 1, oppure **X** se **prossimoG** vale 2;

```
bool muovi(int riga, int col)
```

assegna il valore corrente di **prossimoG** alla cella indicata da **riga** e **col** . Quindi aggiorna il valore di **prossimoG** per dare il turno all'altro giocatore (se è 1 passa a 2, se è 2 passa a 1). restituisce false se non è possibile fare la mossa (true se è ok).

```
void reset()
```

5

## Esercizio: *tic-tac-toe* #1

```

5 class tttPlayGround {
6 private:
7     int playground[3][3];
8     int prossimoG;
9
10    int vincitore;
11    int counter;
12    const char icons[3]={' ', 'O', 'X'};
13    bool check();

```

6

## Esercizio: *tic-tac-toe* #1

```

14 public:
15     tttPlayGround()
16     {
17         reset();
18     }
19     void reset()
20     {
21         for(int i=0;i<3;i++)
22             for(int j=0;j<3;j++)
23                 playground[i][j]=0;
24         prossimoG=1;
25         vincitore=0;
26         counter=0;
27     };
28

```

7

## Esercizio: *tic-tac-toe* #1

```

29     bool finita()
30     {
31         return (check() || counter==9);
32     };
33
34     char prossimo()
35     {
36         return icons[prossimoG];
37     };
38     char vince()
39     {
40         return icons[vincitore];
41     }
42     bool muovi(int i,int j);
43     void show();
44 };
45

```

8

## Esercizio: *tic-tac-toe* #1

```

46 bool tttPlayGround::check()
47 {
48     int pr,pc,pd1=0,pd2=0;
49     int v=0;
50
51     for (int i=0;i<3;i++){
52         pr=pc=0;
53         for (int j=0;j<3;j++){
54             if (playground[i][j]==1)
55                 pr+=1;
56             if (playground[i][j]==2)
57                 pc+=1;
58             if (playground[j][i]==1)
59                 pd1+=1;
60             if (playground[j][i]==2)
61                 pd2+=1;
62         } // fine for j

```

9

## Esercizio: *tic-tac-toe* #1

```

64         if (playground[i][i]==1)
65             pd1+=1;
66         if (playground[i][2-i]==1)
67             pd2+=1;
68         if (playground[i][i]==2)
69             pc+=1;
70         if (playground[i][2-i]==2)
71             pc+=1;
72
73         if ((pr==3) || (pc==3) || (pd1==3) || (pd2==3))
74             v=1;
75         if ((pr==30) || (pc==30) || (pd1==30) || (pd2==30))
76             v=2;

```

10

## Esercizio: *tic-tac-toe* #1

```

78
79         if (v>0)
80             break;
81     } // fine for i
82
83     vincitore=v;
84     return v!=0; // restituisce true se la partita è stata vinta.
85 }

```

11

## Esercizio: *tic-tac-toe* #1

```

86 void tttPlayGround::show()
87 {
88     int iconNumber=0;
89     cout << "+---+"<<endl;
90     for (int i=0;i<3;i++){
91         cout<<"|";
92         for(int j=0;j<3;j++){
93             iconNumber=playground[i][j];
94             cout <<icons[iconNumber];
95         }
96         cout <<"|"<<endl;
97     }
98     cout << "+---+"<<endl;
99 }

```

12

## Esercizio: *tic-tac-toe* #1

```

101 bool tttPlayGround::muovi(int i, int j)
102 {
103     if (playground[i][j]!=0||counter==9)
104         return false;
105     playground[i][j]=prossimoG;
106     counter++;
107     prossimoG=3-prossimoG;
108     check();
109     return true;
110 }

```

13

## Esercizio: *tic-tac-toe* #1

```

112 int main()
113 {
114     tttPlayGround match;
115     int riga,col;
116     bool mossaOk;
117     do {
118         match.show();
119         cout << "Giocatore " << match.prossimo() <<": ";
120         cin >> riga >> col;
121         mossaOk=match.muovi(riga,col);
122         if (!mossaOk)
123             cout << "Mossa sbagliata!"<<endl;
124     } while(!match.finita());
125     match.show();
126     cout <<"Vince:"<<match.vince()<<endl;
127 }

```

14