



Sviluppare un progetto C# che implementi correttamente la logica della programmazione parallela (Multi Thread).

Si vuole simulare un torneo di bowling dove il totale dei partecipanti viene scelto prima dell'avvio.

La simulazione verrà così svolta:

VERSIONE A (Voto MAX 7)

- Un arbitro controlla l'accesso al campo per un massimo di 4 giocatori per volta
- Scenderanno in campo i successivi 4 giocatori, se disponibili, solo al termine di TUTTE le gare del turno
- Al termine di tutti i turni visualizzare il/i vincitore/i
- ATTENZIONE:** l'implementazione deve funzionare correttamente anche se non viene impostato un numero di partecipanti pari a 4 (o multiplo).

VERSIONE B (Voto MAX 10)

- Un arbitro controlla l'accesso ai campi per un massimo di 4 giocatori per volta
- Ogni giocatore è INDIPENDENTE dall'altro: quando un campo viene liberato (fine gara), un altro giocatore, se disponibile, prende il suo posto
- Al termine di tutti i turni visualizzare il/i vincitore/i

I nomi dei possibili partecipanti sono disponibili all'interno dei file giocatori.txt

I partecipanti al torneo scenderanno in campo con estrazione casuale in base alla disponibilità: ogni giocatore può scendere in campo una sola volta.

MODALITA' DI GIOCO

Il partecipante ha a disposizione 2 lanci. Gestire la modifica del testo delle rispettive label (nome giocatore e punti) ed il relativo spostamento della palla (pictureBox).

Attribuzione del punteggio:

Ogni lancio può generare casualmente un punteggio compreso tra 0 e 10. Sommare il risultato dei 2 tiri a disposizione.

SIMULAZIONE LANCIO

Ogni palla deve spostarsi lungo il campo per un tempo generato casualmente (10, 50). La nuova posizione è determinata, anch'essa casualmente, da un valore X compreso tra (10, 20).

I criteri precedentemente riportati garantiranno l'effetto di "indipendenza" tra i vari concorrenti in gara.

Costanti di spostamento:

```
const int PALLA_INIZIO_X = 400;
const int PALLA_FINE_X = 20;
const int TIRI = 2;
```

Suggerimento per un corretto passaggio di parametri al Thread gestore della gara:

Essendoci componenti grafiche di diverso tipo può essere utilizzato, a differenza delle esercitazioni, un array Control

```
Control[] par = new Control[3];
Label[] lblGioc;
PictureBox pic;
```

e successivamente trasmetterli al thread:

```
par[0] = labelGiocatore;
par[1] = labelPunteggio;
par[2] = pictureBoxPalla;
```

| | Strutture per l'elaborazione dati – esiti | Gestione Accessi al campo | Corretta Implementazione Thread | Controllo accesso gara | Componenti Grafiche | TOT. |
|---------|---|---------------------------------|---------------------------------------|------------------------------|------------------------|-------|
| VERS. A | 1 pt | 1,5 pt | 1,5 pt | 1,5 pt | 1,5 pt | 7 pt |
| VERS. B | 1,5 pt | 2,5 pt | 2 pt | 2,5 pt | 1,5 pt | 10 pt |