

Симулация на състезание

Да се напише програма за (проста) симулация на състезание.

1. Да се дефинира класът `Contestant`, който трябва да съдържа:

- скрити членове за неговото име(`string`), скорост(`double`) и изминато разстояние(`double`)
- метода `getName()`, който връща неговото име
- метода `getSpeed()`, който връща неговата скорост
- метода `getDistance()`, който връща изминатото разстояние
- метода `calcDistance(int iteration_count)`, който изчислява изминатото разстояние и го присвоява на атрибута на класа. Разстоянието се изчислява като `distance = speed * iterations`

2. Да се дефинира класът `Race`, който трябва да съдържа:

- скрит член с колекция(от стандартната библиотека) с участниците (`Contestant`)
- метода `getContestants()`, който връща колекцията с участници
- метода `addContestant(Contestant new_contestant)`, който добавя нов участник към колекцията
- метода `resetContestants()`, който връща изминатото разстояние на всички участници на 0

3. Към класа `Contestant` да се добавят конструктори:

- конструктор по подразбиране(без аргументи), който задава празни стойности за името, скоростта и разстоянието
- конструктор, който като аргументи получава стринг с името на участника и неговата скорост
- независимо от конструктора изминатото разстояние винаги започва от 0

4. Към класа `Race` да се добавят конструктори:

- конструктор по подразбиране(без аргументи), който задава празна стойност за колекцията с участници
- конструктор, който като аргумент получава колекция с участниците

5. Към класа `Race` да се добавят методите

- `simulateRace(int iteration_count)`, който:
 - връща изминатото разстояние на всички участници на 0
 - симулира `iteration_count` на брой итерации
 - на всяка итерация обновява изминатото разстояние на всички участници
 - след обновлението подрежда участниците според изминатото разстояние
- `string getContestantStanding()`, който:
 - използва поток от тип `stringstream` за да генерира стринг с информацията за производителя и моделите
 - връща генерирания стринг (не потока!)
 - генерираният стринг има формата

```
<участник 1>: <разстояние>(<скорост>)  
<участник 2>: <разстояние>(<скорост>)  
...  
<участник N>: <разстояние>(<скорост>)
```

- в `simulateRace` да се използва `getContestantStanding` за да се изведе реда на участниците в началото, след всяка итерация и в края на състезанието