

## F1 simulátor



Firmu, ve které pracujete, si najal nejmenovaný tým F1. Vaší úlohou bude tomuto týmu vytvořit simulátor závodů Formule 1, který budou používat pro vlastní analýzy. Programovat nebudete úplně od začátku, ale musíte navázat na kód, který vám zanechal předcházející vývojář. Jde o značně zjednodušenou verzi simulace, která je založena na paralelním a asynchronním programování.

### Úloha

Kostra se skládá z několika předpřipravených tříd, které budete mít za úkol dokončit. Níže jsou popsány, jaký je jejich význam a jak s nimi pracovat.

### TrackPoints

Jde o třídy, které implementují rozhraní `ITrackPoint`. Reprezentují jednotlivé části dráhy - rovinky, zatáčky nebo boxovou uličku. Podrobnější popis těchto tříd naleznete níže.

Formule prochází daným `ITrackPoint` em určitý čas a ne vždy může do úseku hned vjet. Občas musí počkat do doby, než se daný `ITrackPoint` uvolní. Samotný průjezd `ITrackPoint` em také zabere nějaký čas.

Každý `ITrackPoint` vrací pomocí metody `PassAsync` objekt `TrackPointPass`, který obsahuje:

- referenci na daný `ITrackPoint`
- čas, který řidič čekal před průjezdem `ITrackPoint` u
- čas, který trval průjezd `ITrackPoint` em

### Straight

Reprezentuje rovinku. Rovinka nemá žádné omezení na počet současně projíždějících formulí - všechny formule, které se na ní nachází, nemusí na nic čekat a hned ní projedou. Výsledný čas průjezdu rovinkou je průměrný čas průjezdu rovinky vynásobený rychlostí aktuální pneumatiky auta a rychlostí auta na rovince.

### Turn

Reprezentuje zatáčku. Vjezd do zatáčky obsahuje vlastnost `DriveInTime` (v milisekundách) a ve stejný čas může do zatáčky jet pouze omezený počet formulí. Když auto projede vjezdem do zatáčky, průjezd

touto zatáčkou je roven součinu `AverageTime` , rychlosti aktuálních pneumatik na formuli a rychlostí auta v zatáčkách.

## PitLane

Reprezentujte boxovou uličku. Každý tým F1 má právě jeden box. Není tedy možné, aby více jezdců stejného týmu bylo v boxové uličce ve stejný čas. Pokud je tedy box obsazený, musí další jezdec počkat, dokud se box neuvolní. Průjezd boxem také znamená výměnu pneumatik na formuli:

- všechny 4 pneumatiky se mění najednou a paralelně
- výměna jedné pneumatiky trvá v intervalu od 50 do 100 ms
- výměna pneumatik znamená, že od průjezdu má auto nasazené nové pneumatiky
- `TireStrategy` je seznam toho, v jakém pořadí bude mít auto pneumatiky měněno.

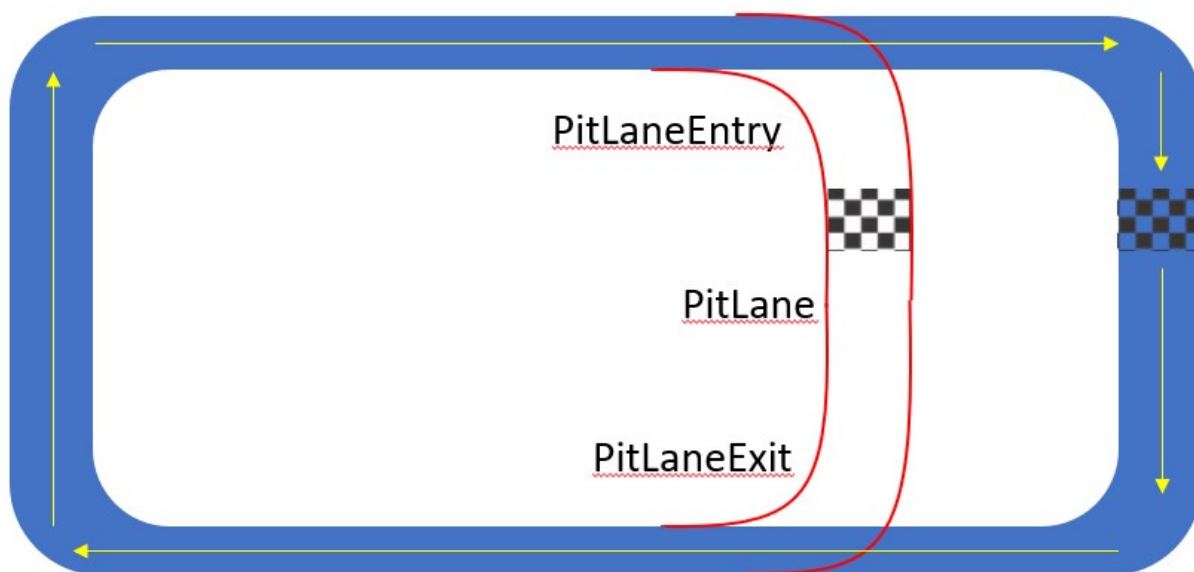
## Track

Reprezentuje celou závodní trať. Tato třída už obsahuje několik metod, kterými můžete přidávat jednotlivé `ITrackPoint` y do tratě.

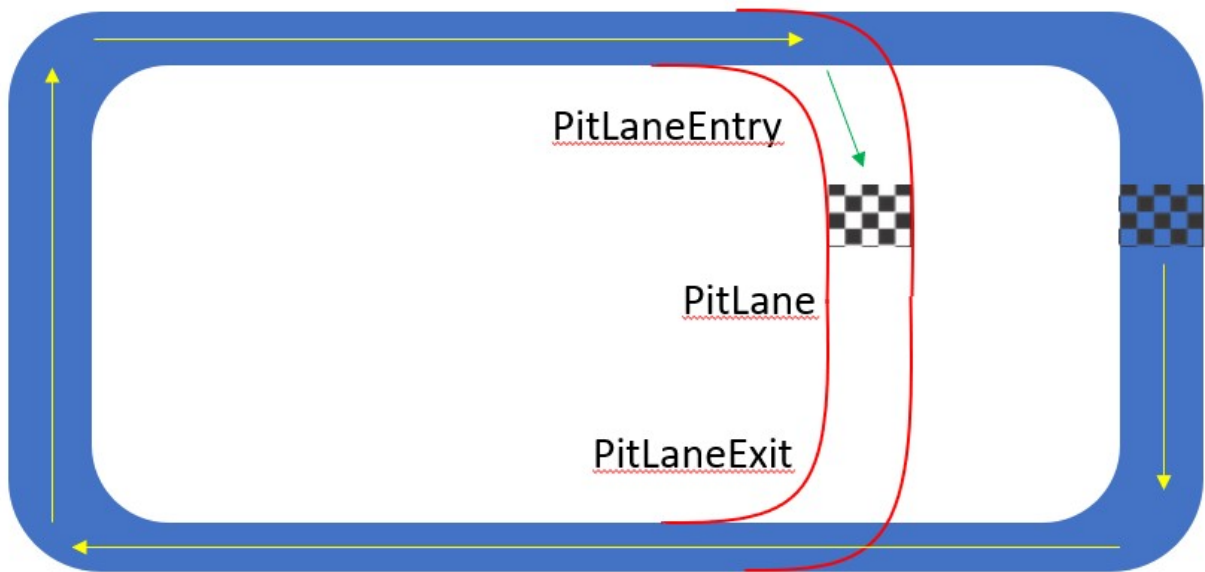
Vaší úlohou bude implementovat logiku metody `GetLap` , která vrací seznam `IEnumerable<ITrackPoint>` pro dané auto. Seznam se mění podle toho, zda formule potřebuje vyměnit pneumatiky, což znamená průjezd přes `PitLane` . Nastat mohou tři situace:

1. Při běžném kole bez výměny pneumatik formule jezdí normálně po okruhu.
2. Pokud je potřeba vyměnit pneumatiky, auto musí projet přes `PitLane`. Do `PitLane` se jede přes `_pitLaneEntry` . Průjezdem `_pitLaneEntry` kolo závodníka končí. Všechny `ITrackPoint` y, které se nachází za tímto průjezdem, jezdec vynechá.
3. Pokud minulé kolo končilo průjezdem `_pitLaneEntry` , toto kolo bude začínat v `PitLane` , po kterém projede formule přes `_pitLaneExit` a pokračuje po dráze dále. Znamená to, že vynechá některé `ITrackPoint` y na začátku dráhy.

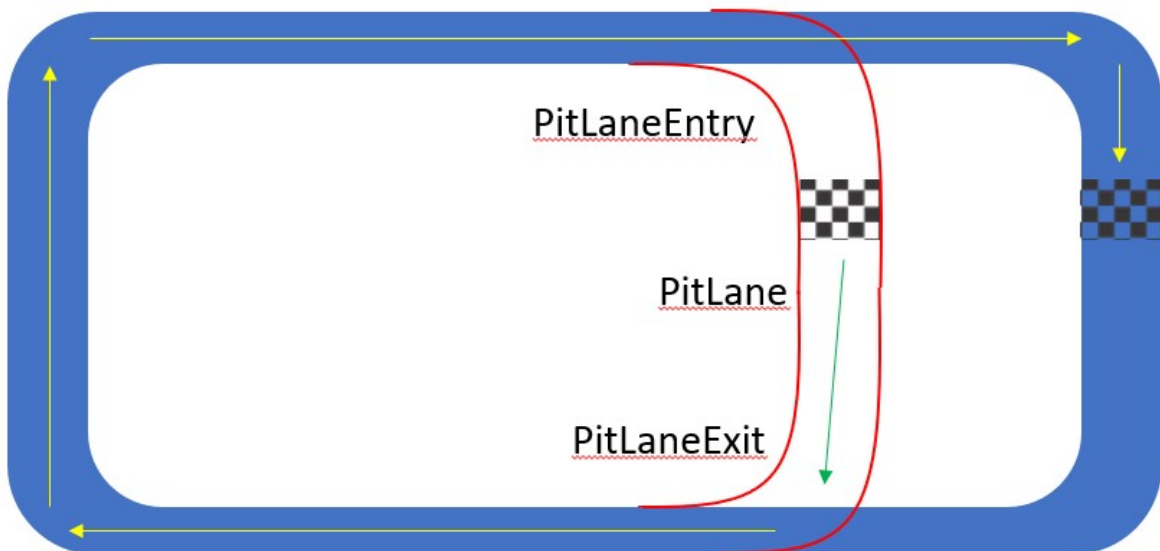
### Normální kolo:



Kolo před výměnou:



Kolo po výměně:



## Race

Vaší úlohou bude implementovat metodu `StartAsync`, která reprezentuje logiku závodu. Konkrétně to znamená:

- Spustí všechny auta v jeden čas na okruh
- Závod trvá určitý počet kol, který je uložen ve fieldu `numberOfLaps`.
- Po průjezdu prvního auta cílem (po dojetí specifického počtu kol) závod končí a formule, které jsou ještě na dráze dokončí svoje rozjeté kolo, ale další kolo už nezačínají.

Další úlohou bude po každém kole vypisovat průběžný čas závodu s následujícími specifikacemi:

- První formule, která dojede kolo bude mít vedle sebe čas, ve kterém kolo projela.
- Ostatní budou mít u sebe zobrazen rozdíl svého času vs času prvního, který kolo dojel

```
Lap: 1
Hulkenberg: 00:03.62
Zhou: +00:00.03
Bottas: +00:00.04
Alonso: +00:00.04
Perez: +00:00.07
Sargeant: +00:00.09
Magnussen: +00:00.10
```

## Simulation

Tato třída reprezentuje simulaci a dovoluje simulovat buď celý závod nebo zadaný počet kol pro jedno konkrétní auto.

## Strategy

Třída obsahuje předpřipravené strategie výměny pneumatik, které můžete použít, případně si můžete vytvořit vlastní.

## CurrentF1

Třída `CurrentF1` obsahuje jednu dráhu - Silverstone, týmy a jezdce, kteří na daném okruhu závodí. Tyto týmy a jezdce můžete upravovat podle sebe. Je to třída, která vám má usnadnit testování vašeho řešení.

## Analytics

Vaší úlohou bude vytvořit několik metod, které je možné použít na analýzu závodu.

- metoda, která vypíše závěrečné pořadí jezdů a jejich čas
- metoda, která vypíše nejrychlejší kolo pro každého jezdce (jezdec + číslo kola)
- metoda, která vypíše průběžné pořadí po zadaném kole Například pro vstup 5 vypíše průběžné pořadí po 5. kole závodu.
- metoda, která vypíše pro každý `ITrackPoint` nejrychlejší průjezd (jezdec + čas + číslo kola) a nejdelší dobu čekání (jezdec + čas + číslo kola)

## Důležité

Celý závod **musí probíhat paralelně**, s tím, že nikdy od začátku závodu (odstartování všech aut najednou) **nedojde k synchronnímu čekání nebo blokování vláken**.

V `Program.cs` je předchystaná ukázka závodu, kterou můžete spustit a zkusit, zda vám vaše řešení funguje. Vaší úlohou není vytvářet žádné uživatelské rozhraní, stačí pouze demonstrovat spuštění závodu.

Tato úloha je zaměřena na paralelní programování a synchronizaci vláken. Dávejte si proto pozor na správnou práci s vlákny a synchronizaci mezi nimi. Všechny metody související se závodem musí fungovat dle zadání a musí být bezpečné je používat z více vláken (thread-safe).