

ADA End term Assignment – Group 2

Background

You will be implementing an F1 race using ADA's concurrency features. There will be 20 drivers in total in the race. The race will be 45 laps. Each driver is associated with a Car (Task) and each car will enter the pit lane (Task) and pit once every 20 laps. Each car will generate a random lap time between 1.14-1.30 seconds every lap. The lap times along with the driver name and current lap of the car will be submitted to Race Control (Task) which will coordinate the track positions of the cars along with a protected task and print out all necessary details. In the end the cars will be sorted in ascending order of their total race time

Implementation

Driver_Pkg

Create a driver package with the following private record called Driver:

- Name of driver: Unbounded_String
- Team name: Unbounded_String
- Total Race Time: Duration – initially 0.0

The Driver package will have the following functions and procedures

- Function to initialize the driver – takes in the driver name and team name and returns a new driver record.
- Getters for the Name, Team Name and Total Race Time.
- To_String function which returns a string in the following format: **“Driver_Name (Team_Name) – total race time: Total_Race_Time”**
- Procedure which increases the total race time by a Duration.

Main Unit

Create an array type of Drivers with index from 1..20 and create an array called **Driver_Arr** object.

Create a protected task **Rand_Lap_Time** which generates a random lap time between **1.14-1.30**. Use the Ada.Numerics.Float_Random Package (This package is not generic). It has an **Init** procedure, **Generate** function which returns a Duration and a private Generator.

Create a protected task **Printer** which will be responsible for all Printing during in the Main Unit. It has one procedure called Print and takes a normal String as input. **Do not print using Put_Line anywhere else but in the Printer!**

Create a protected task **Race_Position** which will keep track of all the driver's positions. It should have a private array of **Drivers** using the same type created above and it should have a private procedure which sorts this drivers array in ascending order of lap time. Next create the following procedures:

- Init with an input parameter of type **Driver_Array** – this procedure assigns the input parameter to the drivers variable in the private of the protected.
- **Increase_Race_Time_Of_Driver** with an input of a driver and a lap time. This procedure then finds the driver in the drivers array and increases their total race time by the lap time.
- **Print_Array** – this procedure calls the sort on the drivers array first and then builds a string. Go through each driver and append their details to the string followed by a new line (**Use ASCII.LF**) before appending the next driver. Once the entire string is built, send it to the printer.

Create a basic task **Pit_Lane** which has one entry called Pit. The entry has the parameters of the Driver and their current lap number that they are on in the race. Print the following string when a car pits: **“***** Driver_Name has pitted on lap number: lap_number*****”**. Then delay by 0.5 seconds. Call the **Race_Position** task and increase the race time of the driver by 0.5 seconds. The Pit entry should be able to be called an arbitrary number of times.

Create a task of **Car**. Each driver will have a car. The car has an entry Init which takes in the total number of laps in the race and a Driver. After the car has been initialized, the car will start running laps until the total number of laps has been reached. Every lap, the car gets generates a random lap time from the relevant protected task and then delays by that amount of time. It also calls the **Race_Control** task and gives it the driver, the current lap of the car and the lap time that was just completed. Every 20 laps, the car will pit.

Create a pointer of the Car.

The **Race_Control** task has 2 entries

- **Init** with parameters Driver Array and total number of laps.
- **Lap_Completed** which takes in the driver, current lap of the car and the lap time just set by the car.

When the Init entry is called, initialize the **Rand_Lap** task, the **Race_Position** Task and initialize a new Car task using the Car pointer with the driver and the total number of laps for all 20 cars in the driver array.

If the **Lap_Completed** entry is called, increase the race time of the driver with the appropriate protected operation. The print out the following string: **“Driver_Name completed lap Lap_Number: Lap_Time”**. After every 20th Car that calls the **Lap_Completed** entry, call the **Print_Array** procedure of **Race_Position**.

The Init and Lap_Completed entries should be able to be called an arbitrary amount of times and if they are not called, there should be a waiting delay of 5.0 seconds before calling the **Print_Array** of **Race_Position** and then exiting the loop.

Add an exception to all tasks and task types (Not protected tasks) which checks for an exception and then if there is, print the string **“Tasking error name_of_task”**.

There should be no tasking errors and all tasks should terminate automatically.

Use the following code for the **Driver_Arr** declared at the beginning of the main and the code for the Race.Control initialization. **This code should be the only code in the statement section of the Main unit (after begin keyword).**

begin

```
Driver_Arr (1) := Init_Driver ("Max Verstappen", "Red Bull Racing");
Driver_Arr (2) := Init_Driver ("Lando Norris", "McLaren");
Driver_Arr (3) := Init_Driver ("Charles Leclerc", "Ferrari");
Driver_Arr (4) := Init_Driver ("Oscar Piastri", "McLaren");
Driver_Arr (5) := Init_Driver ("Carlos Sainz", "Ferrari");
Driver_Arr (6) := Init_Driver ("George Russel", "Mercedes");
Driver_Arr (7) := Init_Driver ("Lewis Hamilton", "Mercedes");
Driver_Arr (8) := Init_Driver ("Sergio Perez", "Red Bull Racing");
Driver_Arr (9) := Init_Driver ("Fernando Alonso", "Aston Martin");
Driver_Arr (10) := Init_Driver ("Nico Hulkenberg", "Haas");
Driver_Arr (11) := Init_Driver ("Yuki Tsunoda", "RB");
Driver_Arr (12) := Init_Driver ("Pierre Gasly", "Alpine");
```

```
Driver_Arr (13) := Init_Driver ("Lance Stroll", "Aston Martin");  
Driver_Arr (14) := Init_Driver ("Esteban Ocon", "Alpine");  
Driver_Arr (15) := Init_Driver ("Kevin Magnussen", "Haas");  
Driver_Arr (16) := Init_Driver ("Alexander Albon", "Williams");  
Driver_Arr (17) := Init_Driver ("Franco Collapinto", "Williams");  
Driver_Arr (18) := Init_Driver ("Liam Lawson", "RB");  
Driver_Arr (19) := Init_Driver ("Zhou Guanyu", "Kick Sauber");  
Driver_Arr (20) := Init_Driver ("Valtteri Bottas", "Kick Sauber");
```

```
    Race_Control.Init (Driver_Arr, 45);  
end Main;
```

Sample output of the last lap done before the termination of the simulation:

Franco Collapinto completed lap 42: 1.189043880
Kevin Magnussen completed lap 43: 1.161654711
Esteban Ocon completed lap 44: 1.240493774
Valtteri Bottas completed lap 45: 1.228247523
Nico Hulkenberg completed lap 43: 1.193350315
Alexander Albon completed lap 44: 1.275409579
Zhou Guanyu completed lap 45: 1.221964002
Franco Collapinto completed lap 43: 1.256044507
Kevin Magnussen completed lap 44: 1.252793193
Esteban Ocon completed lap 45: 1.266109824
Nico Hulkenberg completed lap 44: 1.178814530
Alexander Albon completed lap 45: 1.253445029
Franco Collapinto completed lap 44: 1.198227644
Kevin Magnussen completed lap 45: 1.252242804
Nico Hulkenberg completed lap 45: 1.252255917
Franco Collapinto completed lap 45: 1.299041390

- 1: Charles Leclerc (Ferrari) - total race time: 55.228047729
- 2: Yuki Tsunoda (RB) - total race time: 55.486525299
- 3: Max Verstappen (Red Bull Racing) - total race time: 55.585995080
- 4: Lance Stroll (Aston Martin) - total race time: 55.668961281
- 5: Oscar Piastri (McLaren) - total race time: 55.676101688
- 6: Sergio Perez (Red Bull Racing) - total race time: 55.696600438
- 7: Kevin Magnussen (Haas) - total race time: 55.714203119
- 8: Zhou Guanyu (Kick Sauber) - total race time: 55.841094972
- 9: Valtteri Bottas (Kick Sauber) - total race time: 55.868083715
- 10: Liam Lawson (RB) - total race time: 55.876236440
- 11: Fernando Alonso (Aston Martin) - total race time: 55.957355619
- 12: Pierre Gasly (Alpine) - total race time: 56.015985009
- 13: Esteban Ocon (Alpine) - total race time: 56.175427082
- 14: Nico Hulkenberg (Haas) - total race time: 56.204640749
- 15: Franco Collapinto (Williams) - total race time: 56.214151145
- 16: Lando Norris (McLaren) - total race time: 56.214171410
- 17: Carlos Sainz (Ferrari) - total race time: 56.240019678
- 18: George Russel (Mercedes) - total race time: 56.265474678
- 19: Lewis Hamilton (Mercedes) - total race time: 56.296877145
- 20: Alexander Albon (Williams) - total race time: 56.353639722

1: Charles Leclerc (Ferrari) - total race time: 55.228047729
2: Yuki Tsunoda (RB) - total race time: 55.486525299
3: Max Verstappen (Red Bull Racing) - total race time: 55.585995080
4: Lance Stroll (Aston Martin) - total race time: 55.668961281
5: Oscar Piastri (McLaren) - total race time: 55.676101688
6: Sergio Perez (Red Bull Racing) - total race time: 55.696600438
7: Kevin Magnussen (Haas) - total race time: 55.714203119
8: Zhou Guanyu (Kick Sauber) - total race time: 55.841094972
9: Valtteri Bottas (Kick Sauber) - total race time: 55.868083715
10: Liam Lawson (RB) - total race time: 55.876236440
11: Fernando Alonso (Aston Martin) - total race time: 55.957355619
12: Pierre Gasly (Alpine) - total race time: 56.015985009
13: Esteban Ocon (Alpine) - total race time: 56.175427082
14: Nico Hulkenberg (Haas) - total race time: 56.204640749
15: Franco Collapinto (Williams) - total race time: 56.214151145
16: Lando Norris (McLaren) - total race time: 56.214171410
17: Carlos Sainz (Ferrari) - total race time: 56.240019678
18: George Russel (Mercedes) - total race time: 56.265474678
19: Lewis Hamilton (Mercedes) - total race time: 56.296877145
20: Alexander Albon (Williams) - total race time: 56.353639722