Oblig 2

The roller coaster problem

A number of passengers (1:n) wish to repeatedly take rides with a Roller Coaster. There should be one process for each passengers and one for the Roller Coaster. As for the problem of the sleeping barber, the processes must participate in several synchronization stages. The passengers must wait for the car to be present, and the car must wait for C passengers to enter. After the ride, the car must awake all riding passengers.

The solution does not have to prevent sneaking. Make a solution in the setting of monitors with signal and continue discipline for signaling.

Use the await language to solve this exercise.

In addition, try to formulate a reasonable invariant based on your solution, but it is not necessary to prove the invariant using programming logic.

Code:

```
 int C;

monitor Coaster_Monitor {
    # Invariant: ( is_filling = 1 and available_seats > 0) or ( is_filling = 0 and available_seats = 0 )
    cond car_available;     # signaled when is_filling = 0
    cond car_filled;        # signaled when available_seats = 0
    cond car_left;          # signaled when available_seats = 0

    available_seats := C;
    is_filling := 1;

    procedure get_seat() {
        while (available_seats = 0) wait(car_available);
        available_seats := available_seats - 1;
        if (available_seats = 0) {
            signal car_filled;
        }
        wait(car_available); # waiting for the ride to finish
    }

    procedure start_ride() {
        while (available_seats > 0) wait(car_filled);
        is_filling := 0;
        signal car_left;
        # ride starts
    }

    procedure finish_ride() {
        while (is_filling = 1) wait(car_left);
        is_filling := 1;
        available_seats := C;
        signal_all(car_available);
    }
}

process Passenger[id=0 to n] {
    Coaster_Monitor monitor;
    while (true) {
        monitor.get_seat();
    }
}

process Car {
    Coaster_Monitor monitor;
    while (true) {
        monitor.start_ride();
        # ride is happening here
        monitor.finish_ride();
    }
}
```