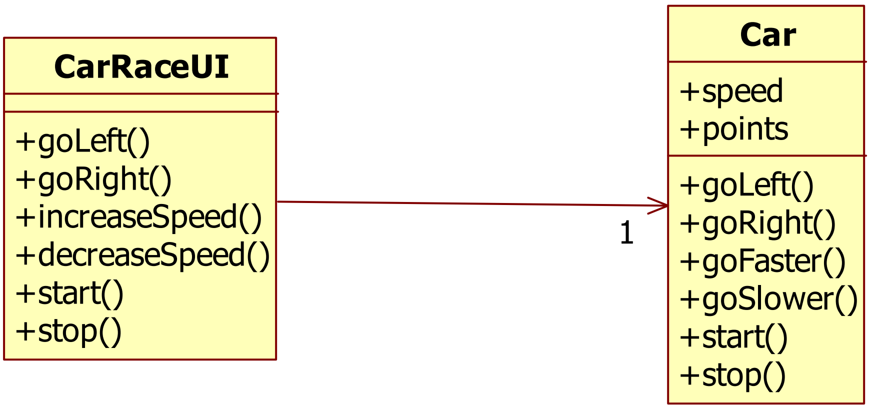
Question 1 [50 points] {60 minutes}

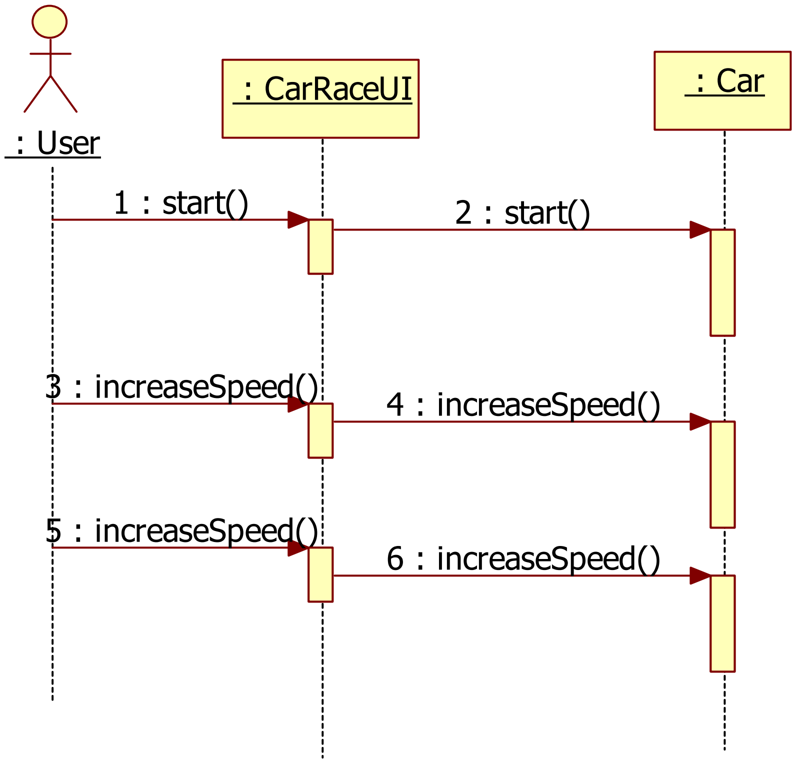
Suppose we are asked to write a simple car racing game with the following requirements:

- The user can do the following actions: go left, go right, increase speed and decrease speed.
- The user gets points for how well it stays on the road, and how fast it can race.

We design and implement this simple car racing game as follows:



The sequence diagram of a possible scenario looks like this:



Now suppose we are asked to redesign our application into a car racing framework. The framework has the following requirements:

1. The framework should support undo and redo functionality for the actions **increaseSpeed**() and **decreaseSpeed**().
2. The framework should support different car modes. The modes supported by the framework are:
   a. Rookie Racer: This means you have less than 1000 points, and if you press the increaseSpeed button, your speed is increased with 1 mile per hour. If you press the decreaseSpeed button, your speed is decreased with 1 mile per hour

   b. Racer: This means you have between 1000 and 5000 points, and if you press the increaseSpeed button, your speed is increased with 2 miles per hour. If you press the decreaseSpeed button, your speed is decreased with 2 miles per hour

   c. Pro Racer: This means you have between more than 5000 points, and if you press the increaseSpeed button, your speed is increased with 3 miles per hour. If you press the decreaseSpeed button, your speed is decreased with 3 miles per hour

   If the car has enough points, it will automatically be upgraded to the next node. The car can only be upgraded to a higher mode when the goFaster() or goSlower() method is called on the Car.

   It should be easy to write racing applications using the framework that support different car modes.
3. When we start the game by pressing the start button on the CarRaceUI, the car speed and number of points is loaded from the database. When we stop the game, the car speed and number of points is stored in the database.
4. In the racing game, we want to show all kind of car information (speed, points, current user action) on the screen in different window frames. The user can then choose which window frames will be shown on the screen. The framework should support the following window frames:
   - A window that shows the current speed of the car
   - A window that shows the number of points
   - A window that shows the last user action

   It should be easy to write racing applications using the framework that support different window frames, for example a window that shows both the speed and the points next to each other.

a. [5 points] Draw the class diagram of your design of the framework. **Only show the classes with their names and the relationships. Do NOT show attributes and methods**.

b. [45 points] Draw the sequence diagram of the following scenario:

1. The user clicks the **start** button, and the start() method is called on the CarRaceUI. On the screen we see the following window frames:
    a. A window that shows the current speed of the car
    b. A window that shows the number of points
    c. A window that shows the last user action

    The car is in **Rooky Racer** mode, and the car has **990 points**.
2. The user clicks the increaseSpeed button, and the speed goes up with 1 mile per hour. At the same time, the user gets 15 more points, and should be promoted to the **Racer mode.**
3. The user clicks the **undo** button.

Make sure you show the given scenario in your sequence diagram. If you show a different scenario, then you will lose points.