

# The PHP Race Game

You are asked to build a race simulator in PHP 7.4. The game must simulate cars racing on a track made of curves and straights. A simulation is a series of rounds. On each round, a car occupies a position on the track according to the rules. The goal is to provide the position of each car on each game round at the end of the simulation. The first car to cross the finish line wins and the simulation ends.

The rules:

1. A track is a random list of straights and curves called elements. Each element has the same length, regardless if it's a straight or a curve.
2. A track is made up of approximately 50% curves and approximately 50% straights.
3. A track has exactly 2000 elements in total.
4. The elements on the track come in multiples of 40 of the same type. So the minimum length of a series of elements is 40. For example, if the first element of a track is a curve, 39 curves must follow it. If element 41 is again a curve, then again 39 curve elements must follow. If element 81 is a straight, then 39 straight elements must follow.
5. Each car has two types of speeds:
  - a. speed on straight, and
  - b. speed on curve.
6. The speed is the number of elements a car can advance per round on a particular element type.
7. Each car has a total speed of 22. The minimum speed of each type, curve and straight, is 4. This means that if a car has a curve speed of 10, then it must have a straight speed equal to 12.
  - a. Curve and straight speeds are chosen randomly, but according to the rule as per point 7.
8. The outcome of a race is represented by the class **RaceResult**, which in turn contains a list of **RoundResult** objects.
  - a. A RoundResult is an object with two elements, a round number and a cars position array. The cars position array represents the position on the track of each car on a given round.
9. If a car starts a round on an element type, it can only end the round on the same element type, or on the first element of the next sequence of elements, if it has enough speed to reach it.
  - a. So, for example, let's assume that car1 speed on straight is 18, and that the track starts with straights. Then at round 0, the car is on element 0. At round 1, the car is on element 18, at round 2 the car is on element 36. If element 40 is a straight, then on round 3 the car is on element 54. If element 40 is a curve, then on round 3 the car is on element 40.
10. The first car that arrives at the last element wins. If two or more cars arrive at the last element at the same time, it's a draw.
11. There are 5 cars in total.

12. The ultimate goal of this work is to provide a clean and solid solution for the task. You won't need to store data onto a Database. Frameworks are not allowed, only vanilla PHP 7.x. The attached folder includes a list of mandatory classes and functions that must be used as part of the solution, but you are free to create additional files to achieve the best result. Just note that the result must be of type `RaceResult`. This will be used as input of a test function which will validate your solution.