

# Assignment 4

Inheritance, interfaces and exceptions

---

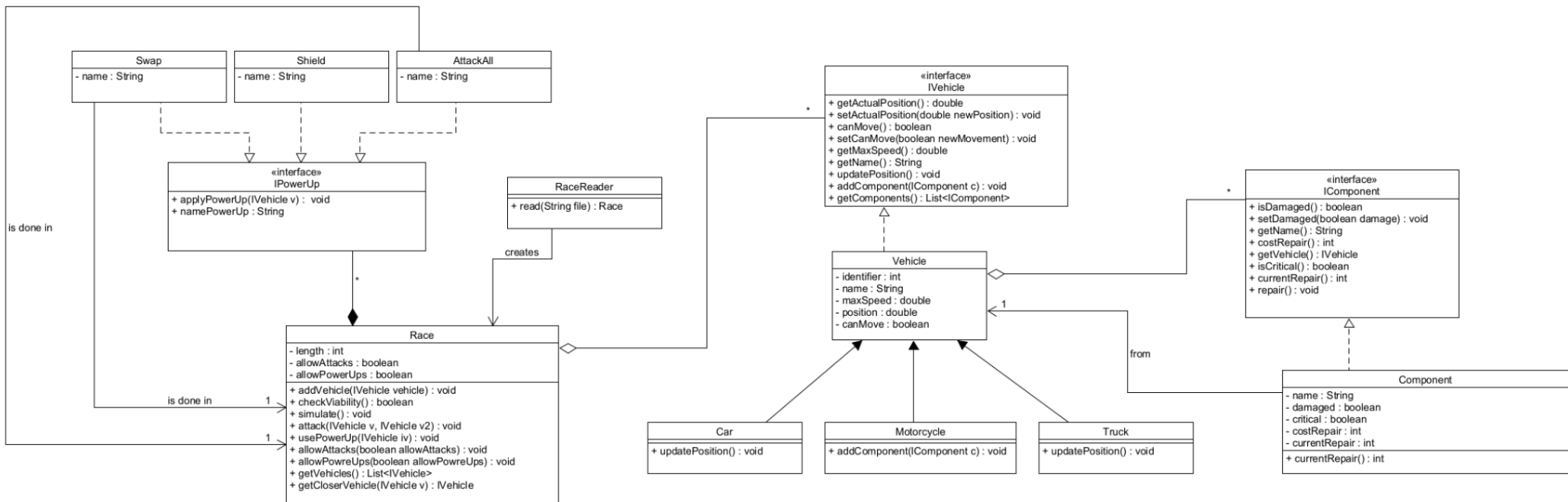
28/04/21

ADSOF 2292

Daniel Varela

Guillermo Martín-Coello

### Class diagram:



## **Decisions and problems through the development:**

The goal of the assignment was to simulate a race between vehicles. To manage that achievement, we first started designing the basic components in a very modular and extensible way, so when we added the new functionality it was easy for us to extend the code.

A big decision that we made was to create a vehicle class to implement the given vehicle interface, because that let us save a lot of time and code by using the same methods for all the vehicles. We took the same decision for the components, but not for the power ups, as each power up method worked in a very different way.

At first we thought about creating different classes for each component but then we realised they don't have any particular method or attribute specific, so we thought the best design choice was to use a unique class implementing the interface, and each component depended on what it was assigned through their constructor method.

For the different power ups, we realised that we needed to access the race to change the different vehicles position in case the power up requested to do so. As we couldn't change the given interface, we decided to give access to the race pointer to all the power up classes that needed so.

Instead of having a very large simulation method in race, we modularized parts of it so it was easier for us to read, and also to access those functionalities from other parts of the program to avoid repeating code.

## **Pending problems:**

We achieved all the expected functionalities for the assignment, but we realised about some features we could add to the program to improve its quality.

The vehicle interface is a good way to group all the vehicles, but for this program all the vehicles worked in a pretty similar way, so the most efficient way of programming it was through an abstract class which had all the shared attributes of the vehicles, and then complementing it with a class for each type.

The power ups of the race are always all the created ones. It would be nice to be able to choose which power ups to be in the race through the input file.

The output is really dense and hard to read, it would be easier to read the results if it was shown in a more intuitive way using colors for example.

## **Part 2:**

Question: As the Morcycle always wins in 12 turns ( as it's speed is 9), the minimum max velocity for the car to win the race must be 8.34, just enough to do the race in 12 turns, in the case he got the chance of going to max speed in all the turns.

## **Power up Shield:**

We added a new power up to the simulation: the Shield power up. The shield is a defensive power up which adds a new component to the vehicle that gets it (the shield component). This component is repaired in only one turn if broken, and is useful to the vehicle because it counts as a non critical component, that results in less chances of getting hit on a critical one. The more components the vehicle has, the less protection the given shield provides, as for a bigger vehicle with more components it would protect less. We made this power up because the motorcycle was in a disadvantage since it couldn't attack on the attacking phase, with this power up, if given to a motorcycle it reduces its chances of being hit on a critical component when hit from 100% to 66% on the first shield which was a needed boost for that vehicle.