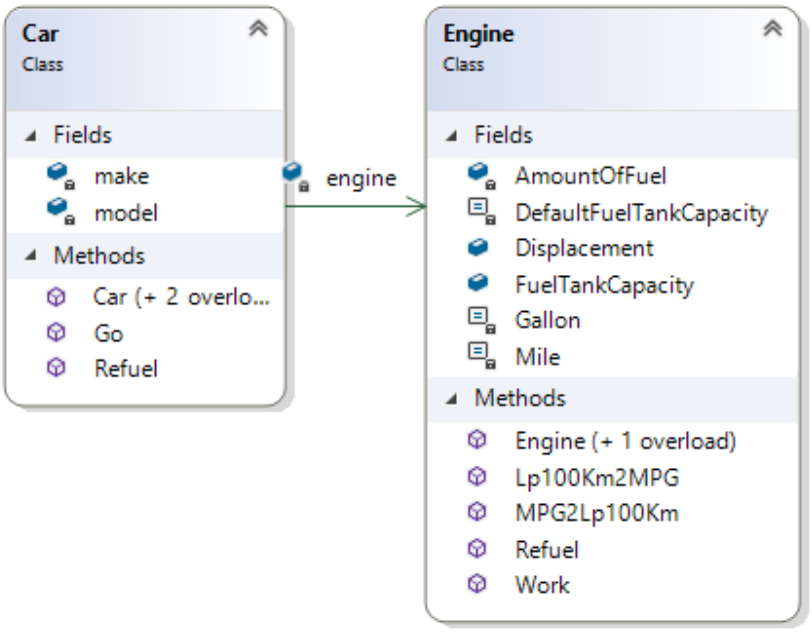
Object-oriented programming

Laboratory classes

Task 2: Car simulator

The goal of the task is to create a simple car driving simulator. While implementing subsequent points at every step, make sure that your model is consistent with reality!

1. Create a new console application, name it *Task2* and then add two new classes representing a *Car* and an *Engine*.
2. Add two methods in the *Engine* class to convert the combustion from litre/100 km to miles per gallon (mpg) and vice versa. The methods are to receive combustion expressed in one quantity and return a converted value. Use the appropriate method in the main function to check what is your car fuel consumption in mpg.
3. The engine is characterised by the capacity and amount of fuel, as well as the capacity of the fuel tank. Make sure that the value for the tank capacity cannot be modified after the engine instance has been created.
4. Add a field in the engine definition that will store the standard (default) fuel tank capacity for each engine you create.
5. Add a constructor that will allow you to set the displacement and amount of fuel. The constructor also sets the fuel tank capacity to the default value.
6. Use the engine constructor in the *Main()* function of the program to create an example engine.
7. Add a second constructor that, in addition to displacement and fuel amount, will also allow you to set the fuel tank capacity. Use the previously created constructor.
8. Use the second constructor in the *Main()* function.
9. The car is characterised by make, model and engine. It also has the *Go()* method, which receives as an argument the distance to go expressed In km. The method works as follows:
   - prints the message “I'm riding”;
   - waits the period of time calculated with the assumption that the car travels 1km per 100ms (*Thread.Sleep ()*);
   - calls the *Work()* method of the engine, which reduces the amount of gasoline in the tank (assume that the engine burns four times the engine capacity per 100km);
   - prints the message “Here I am!” on the screen.
10. The car is to have three constructors that allow you to enter the following values
    - make, model, engine capacity, fuel volume, fuel tank capacity;
    - make, model, engine capacity, fuel volume;
    - make, model, engine.
11. Create a new car in the *Main()* function of the program and test its operation.
12. Add the *Refuel()* method to the car (and to the engine). Handle errors.
13. Implement a car driving simulator. At the beginning of the program, the user inputs all necessary parameters of his vehicle, and then one can enter distance and start driving. Between trips, the user can also refuel the car with the indicated amount of fuel.
14. Change the implementation of the *Go()* method to display an animation of the moving car on the screen.


Car
Class
Fields
make
model
Methods
Car (+ 2 overlo...
Go
Refuel
engine
Engine
Class
Fields
AmountOfFuel
DefaultFuelTankCapacity
Displacement
FuelTankCapacity
Gallon
Mile
Methods
Engine (+ 1 overload)
Lp100Km2MPG
MPG2Lp100Km
Refuel
Work