PointDistance 🡪

Create class Point with properties X and Y (coordinates on the coordinate system)

Create method Point.GetDistance(Point point)

Input parameters – object of type Point

Output parameters – distance between two points

Create method Point.GetDistanceToCenter()

Input parameters –

Output parameters – the distance between the point and the center of the coordinate system

Test your class in the main method

School 🡪

Create class Student with properties and fields

Fields

Grades – List<double>

Properties

Id – string, length 5 symbols,

Name – string, length at least 1 symbol,

Birthday – DateTime,

Age – int

Create method Student.AddGrade(double grade)

Input parameters – double grade

Output parameters – void

Create method Student.AverageGrade()

Input parameters –

Output parameters – double average (rounded to the second digit after the floating point)

Create method Student.PrintGrades()

Input parameters –

Output parameters – void

Test your class in the main method

VehicleCatalog 🡪

Create class Vehicle (should not be instanced) with properties

HorsePowers – int, at least 1,

Color – string, at least 1 symbol

Id – string between 9 and 10 symbols

Create class Car and inherit Vehicle

Add property

NumberOfSeats – int, at least 1

Override ToString() to return

A {Color} car with {HorsePowers} horse powers, {Id} Id and {numberOfSeats} number of seats

Create class Truck and inherit Vehicle

Add property

Tonnage – int, at least 5

Override ToString() to return

A {Color} truck with {HorsePowers} horse powers, {Id} Id and {Tonnage}t tonnage

Create class Catalog to store Cars and Trucks

Class functionality 🡪

Add only unique by id cars and trucks and store them

Has property that return stored vehicles count

Has property that returns number of cars

Has property that returns number of trucks

Has method that prints all vehicles on the new line

Override ToString to return what print method prints

Has Add(Vehicle v) method that add an unique vehicle into catalog, if a vehicle already exist, throw an exception with an appropriate message

Has Remove() method that remove a vehicle by Id and returns a bool

Sorter 🡪

Create a class InsertionSort

Create method Sort([] arr) that sorts in place, a collection