C# 4. Practice – Inheritance

1. Exercise
The date of Animal Tournament is very near, but the organizers of the competition want changes in our program.

According the original contract our task was:
"Registering an animal, it is necessary to give its *name* and *birth year*. These data can't be changed during competition. Each animal is scored, it gets points for its *beauty* and its *behaviour*. Determining the *resulting score*, the age of an animal is important (now age is only in years). Over a unified *age limit* their score will be 0, under this limit the score is computed. In the case of a young animal the beauty is more important, in the case of an older one the behaviour is significant. (For example if the age limit is 10 years, then the score of a 2 years old animal is: 2*behaviour + (10-2)*beauty.)"
However now the organizers inform as, that only cats and dogs can be participant on this tournament, moreover their valuations are not the same. They ask us for these modifications:
The dogs have a third criteria: they get score for the *relationship with their owner*. This score is added to the score computed according to beauty and behaviour, and it is given before other scoring. If this score is missed, then the full score is 0.
As dogs and cats are together, therefore a cat can't be competitor without a *supplier box*. The existence of box is record during registration, but it can be changed. However if a cat has no supplier box at the beginning of competition, its full score will be 0.
As it is possible that more animals have the same name, therefore an *id-number* is necessary as well. Let it be the serial number of registration.
In the controller, please write the registration (mixed cats and dogs), and make a competition.
The ToString() method of animals lists the data in the next form: id number, dog/cat name, score.
Read the data from the *animals.txt* file.

2. Exercise
Vehicle fleet. A contractor wants to manage a vehicle fleet and we are asked to create a C# application to maintain the data of vehicles. Now he has buses and trucks and a large area, so he can extends the current fleet if it is necessary. Maybe later on he can buy other kind of vehicles as well.
It would be very useful to create a flexible program, so it easily would be modified in the future. Therefore we use inheritance.
Of course, the base class will be the Vehicle class. Vehicles have attributes associated with them that need to be tracked. It is important to know the manufacturing year and the registration number of the vehicle, however the registration number can be changed. So we need a unique identifier as well. This class can be abstract, because there is no vehicle object.
Their fuel consumption is very important as well. For simplicity, now it is an estimate value, but of course, it can be changed. Now the contractor has used vehicles, so he know this value, but maybe later on he can buy very new vehicles as well, without knowing their consumption.
The condition of odometer is an essential value as well. This value increases during transportations. (An odometer or odograph is an instrument that indicates distance traveled by a vehicle, such as a bicycle or automobile. The device may be electronic, mechanical, or a combination of the two.)

The contractor wants to rent these vehicles. Of course, a vehicle can be rented if it is free. The rent fee consists of more components. There will be a unified base fee, plus the cost of current transport increased with a unique profit margin. The cost of current transport can be computed according to the current distance, the fuel consumption of vehicle and the current price of fuel.

Renting a bus, the rent fee increases with a computed value depending on the number of seats. This value is: the number of seats multiplied by a multiplier value. This multiplier is the same for all buses.

Renting a truck the rent fee increases with a computed value   as   well. This value is: the carrying capacity of truck multiplied by other   multiplier. This multiplier is the same for all trucks.

Let us operate the vehicle fleet:
- Read the data from a data file
- Rent vehicles – this means: Repeat randomly many times: Select a random vehicle. If it can travel, then print the data of vehicle (its registration number, the current cost and the rent fee). Compute the sum of costs and the sum of rent fees. At last, select a random vehicle, which finishes its travelling.
- Make statistics, that is compute the average of vehicles' age, and determine which vehicles travelled the most (the longest distance).
- Sort them according to fuel consumption in increasing order.

3. Exercise
Write program to manage different cylindrical shapes.
- There are cylinders which are only geometric shapes. Cylinder has radius r, and height h.

- There are rods (solid (filled) cylinder) which have mass. These kind of shapes have density also.

- There are tubes. (Tube is a hollow cylinder of metal, plastic, glass etc.)

Cylinder

Rod

Tube

Create the classes of the given shapes according to their hierarchic relationships (inheritance).
Every shape can retrieve its base data, volume and mass (if it has a mass).
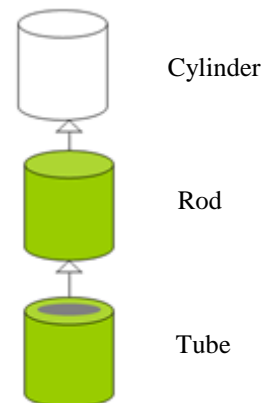
Write a Controlling class to create a cylinder, a rod and two tubes and the program is able to write the data of these objects to the console.

4. **Exercise**
A **book** is characterized by its *author*, *title*, *page number*, *purchase price*, *tax* and the *selling price*. The tax is determined by the *uniform VAT rate*. The selling price is the purchase price plus calculated tax value. The book cover (ToString () method) shows the author, title, page number and the selling price values.

The **foreign-language books** are also characterized by book's data as well as by a headword level indicating the difficulty of the book. When a foreign language book object is created the headword level property will get an integer value, but the book cover will show a text (knowledge level) assigned to headword level. The book cover also shows information about how many foreign word knowledge needs to read it (what is the vocabulary).

For example, the default data:

Headword level	Number of Words	Knowledge level
1	700	Basic level
2	1500	Intermediate level
3	3000	Advanced level (fluent)

To stimulate language learning, the selling price of the book is reduced with a discount % (NumberOfWords/1000), but of course the price cannot be less than a given minimum value.

Create a controller class. Use the attached *books.txt* data file (or a similar data file created by you) to read the data then list the books to the console. The file structure per line: *language*, *author*, *title*, *page number*, *purchase price* (HUF).
Assume that all books will be sold and calculate the total income. Which one (s) is the most expensive and which one (s) the cheapest book (s)? Sort books in descendant order by their selling price.

5. Exercise
Write a personal finance manager program that maintains information on your **bank accounts**. Incorporate these rules:
• For the **savings accounts**, you can make a maximum of three withdrawals in a month without incurring a fee. The bank charges $1.00 for every withdrawal after the third.
• For the **checking accounts**, the bank charges $0.50 for every check you write for the first 20 checks (i.e., withdrawals) in a month. After that, there will be no charge.
You should be able to open and save account information to a file. You should be able to list all transactions of a given account or of all accounts.