# Exercises about Generics

* Solve them in Visual Studio.

## Exercise 18.01

|  |  |
| --- | --- |
|  |  |
|  | In the documentation, a basic generic stack is created.  It has the functionalities Push() and Pop().  This exercise resembles on the example, but we will create a Queue.  A Queue has the principle of first in, first out. |
|  |  |

* Create a generic class Queue.
  + The queue has a maximum of 10 items in it.
* Create several methods:
  + Enqueue (an item).
    - This adds an item to the queue.
    - Make sure that you can’t exceed the maximum numbers of items.
  + Dequeue (an item).
    - This removes an item from the queue.
    - Make sure that the routine works if no items are in the queue.
  + Print.
    - Loop thru the items of the queue.
    - Print their value (use ToString()).
* Create a test routine.
  + Make 2 queues.
    - One with strings.
    - One with integers.
* Prove that everything works.

## Exercise 18.02

* Create an class (or interface) for a vehicle.
  + Properties.
    - Number of wheels. (4, 6, 3, …).
    - Maximum speed. (250, 100, 60, …).
    - Type (BMW, Mercedes, Bugatti, …).
    - Subtype (318, A1, 911, Phanthom, …).
    - Price (be realistic 😊).

|  |  |
| --- | --- |
|  |  |
|  | What is the best option here?  A Class or an Interface? |
|  |  |

* Create a class for Cars, that inherits from vehicle.
* Create a class for Motorcycles, that inherits from vehicle.
* Create a class for Trucks, that inherits form vehicle.
  + Add a property: Maximum load.
* Create a class for a Vehicle collection.
  + Only data types that are implementations from the class or interface Vehicles are allowed.
  + Create a list of vehicles.
  + Create an Add functionality.
  + Create a property that counts the number of vehicles in the list.
  + Create a property that calculates the average price of the collection.
  + Create a method that overviews the items with all their properties.
* Test the complete functionality.
  + Prove that you can’t create a list of items that aren’t vehicles.
  + Prove that an item that does not inherits from Vehicle can’t be added to the Vehicle collection.
  + I will try to break your code by just typing stuff in your test routine.
  + I will surely test if the Maximum Load of the trucks I’ve added are shown.

## Exercise 18.03

* You create a list of companies.
* A company has a tax number and company name.
* Some of the companies are clients.
* Some of the companies are suppliers.
* Some of the companies are both.
* Create a functionality to add a client.
* Create a functionality to add a supplier.
* The tax number should be an unique key.
* Changing the name of a company corresponding to the tax number is possible. Pay attention on the companies that are a client and a supplier.
* Create a method that shows the clients.
* Create a method that shows the suppliers.

|  |  |
| --- | --- |
|  |  |
|  | There are several techniques to solve this.  Remember that this is an exercise of generics.  Give me 2 different solutions to solve this, one must be with generics. The other, you can choose what technique you use. |
|  |  |

## Exercise 18.04

* Creates a class that defines a generic array of 100 elements.
* This class must have a constructor.
* That constructor fills the array with the default value of the data type used.
* If the datatype is a reference type based on a class you created, the default value is null.

### Variant 1

|  |  |
| --- | --- |
|  |  |
|  | Recreate the exercise but make sure that you run the constructor of the data type you use.  e.g. You create a Panda class, that class has a default constructor (the one with no parameters) |
|  |  |