David Šiška School of Mathematics Sciences, University of Edinburgh

2018/19 Semester 1

Object Oriented Programming with Applications

Problem Sheet 2 - Wednesday 10th October 2018[1]

Exercise 2.1. Use generics to write a method that will print the contents of any array to the console. That is, fill in the dots in the code below.

```
static void PrintArray<T>(T[] array)
{
    //....
}
```

Solution: Here is the code:

```
static void PrintArray<Type>(Type[] a)
{
    for (int i = 0; i < a.Length; i++)
        Console.Write ("{0}, ", a [i]);
}
```

Exercise 2.2. Modify the sorting method from the lecture / lab to use generics. You will need a method declaration that looks something like

```
static void MySort<T>(T[] numbers) where T : System.IComparable<T>
```

Moreover, instead of using the usual comparison operator for numbers (i.e. $<$ or $>$) directly you will need to use the `CompareTo` method provided by the `IComparable` interface.
Solution: Here is the code:

```
static void MySort<Type>(Type[] numbers, ref int countComparisons) where Type : System.IComparable<Type>
{
    bool swapped;
    countComparisons = 0;
    do {
        swapped = false;
        for (int i = 0; i < numbers.Length - 1; i++)
        {
            countComparisons++;
            if (numbers[i].CompareTo(numbers[i + 1])>0)
            {
                Type tmp = numbers[i];
                numbers[i] = numbers[i + 1];
                numbers[i + 1] = tmp;
                swapped = true;
            }
        }
    } while (swapped);
}
```

The one thing to observe is that `numbers[i].CompareTo(numbers[i + 1])` returns 0 if the elements are equal, positive if `numbers[i]` is larger that `numbers[i+1]` and negative otherwise.
The algorithm is called “Bubble sort” if you would like to learn more about it.

[1]Last updated 25th October 2018