The purpose of this assignment is to learn how to use and introduce generic types in Java classes.

Exercise 1 – Tuples

Open the *S1Es1.java* file in your IDE. The program implements the *Tuple* class that is capable of storing two Object instances and allows to switch them, by returning a new Tuple class with values switched.
Starting from the provided source code implement:

- class Tuple1, that defines a generic type T for the stored values. The swap method should return a new instance of Tuple1. Implement the example test cases with the new class in the *testTuple1Class* method.
- class Tuple2 which defines two generic types (T1, T2) to specify individually the types of the first and second element. The swap method should return a new instance of Tuple2. Implement the example test cases with the new class in the *testTuple2Class* method.

Exercise 2 – MyQueue

Write a program that implements a generic queue developed as a singly linked list of nodes. Each node should be used to store a single value and to link to the next node in the queue. Limit the functionality of the queue to add, remove, sort and print operations:

- *Add* : creates a new node with the given value and appends it at the end of the queue
- *Remove*: removes the head node of the queue and returns the value stored in the node
- *Sort* : sorts the queue using a *Comparator* passed as parameter to the method
- *Print* : prints all node values (using the *toString* method) starting from the head

To test your implementation create at least 2 queues of different types and for each queue:

1. Add elements
2. Print queue
3. Sort queue
4. Print queue
5. Remove some elements
6. Print queue

HINT:
Use the following pseudocode to implement the sorting algorithm. Swap node **values** instead of nodes!

```
start = head;
while (start not null) {
    cur = start.next
    while (cur not null) {
        if (compare(start.value, cur.value))
            swap values
        cur = cur.next
    }
    start = start.next
}
```

Exercise 3 – Matrix (Optional)

Write a program that implements a generic Matrix class that uses a bi-dimensional array to store its elements. The Matrix class constructor must receive the number of rows and columns to create the bi-dimensional array. Make sure your program provides at least the following methods:

- *set(row, col, value)* : stores the given value at the specified row and column in the bi-dimensional array.
- *get(row, col)* : returns the value stored at the given row and column.
- *print()* : prints the content of the whole matrix, using Object's *toString* method.
- *transpose()* : returns a new Matrix instance representing the transpose.

Test your implementation by creating at least two Matrix instances holding different types (e.g Integer and String) and test all methods.