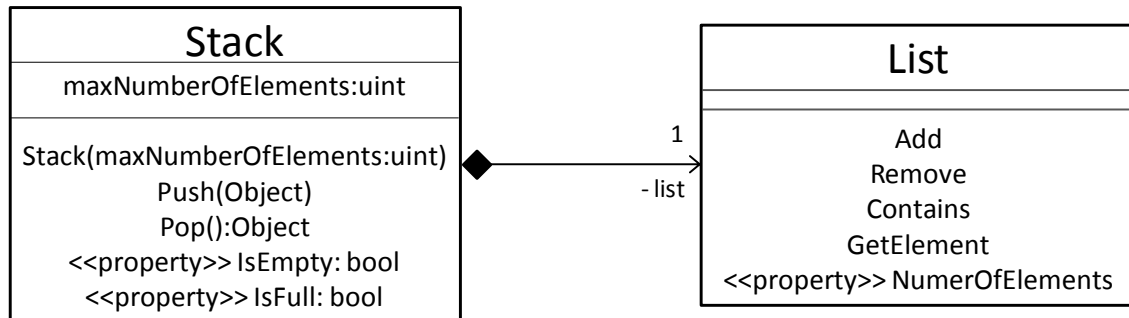# Mandatory Activity. Object Oriented Paradigm. Lab 03.

This activity should be autonomously done by the student. **It must be done prior to the following laboratory class**. It will be used as part of the following laboratory.

## Activity

Reusing the `List` class of the previous laboratory class, implement a polymorphic `Stack` class, following the subsequent design:

```
┌─────────────────────────────────┐          ┌──────────────────────────────────┐
│             Stack               │          │               List               │
├─────────────────────────────────┤          ├──────────────────────────────────┤
│    maxNumberOfElements:uint     │          │                                  │
├─────────────────────────────────┤    1     ├──────────────────────────────────┤
│ Stack(maxNumberOfElements:uint) │◆────────>│              Add                 │
│          Push(Object)           │  - list  │             Remove               │
│          Pop():Object           │          │             Contains             │
│   <<property>> IsEmpty: bool    │          │            GetElement            │
│    <<property>> IsFull: bool    │          │  <<property>> NumerOfElements    │
└─────────────────────────────────┘          └──────────────────────────────────┘
```

The implementation of `Stack` must follow the design by contract. Therefore

- Class invariants must be identified and implemented.
- Preconditions and postconditions of public methods should be identified and implemented.

Use the appropriate mechanisms provided by C# to implement the class contract.

Test its correct behavior by using the testing tool of Visual Studio.

# Optional Activity. Object Oriented Paradigm. Lab 03.

The following activity is not mandatory.

## Activity

Identify the contract (class invariant and preconditions and postconditions of public methods) of the polymorphic `List` class. Add the contract implementation to the existing list.

Test its correct behavior by using the testing tool of Visual Studio.