Assignment 3

Exercise 1

Find preconditions and object invariants for the code on page 2. Express them using the syntax of C#'s Code Contracts:

- At the beginning of each method, `Contract.Requires(expr);` can be used to denote a precondition. Here, `expr` should be a pure boolean C# expression referring only to fields and methods with greater or equal visibility as the method.
- Object invariants are denoted in a special contract invariant method. Again, the visibility of all referred fields must be greater or equal to the visibility of the contract invariant method.

```
[ContractInvariantMethod]
private void ObjectInvariant() {
        Contract.Invariant(expr);
        ...
}
```

- Methods can be marked with the `[Pure]` attribute (in the line before the method declaration) to be pure. Only pure methods can be used in contract expressions.

```
public class Bag
{
    private int[] elems;
    private int count;

    public Bag(int[] initialElements) {
        this.count = initialElements.Length;
        int[] e = new int[initialElements.Length];
        initialElements.CopyTo(e, 0);
        this.elems = e;
    }

    public Bag(int[] initialElements, int start, int howMany) {
        this.count = howMany;
        int[] e = new int[howMany];
        Array.Copy(initialElements, start, e, 0, howMany);
        this.elems = e;
    }

    public int Count() {
        return count;
    }

    public int RemoveMin() {
        int m = System.Int32.MaxValue;
        int mindex = 0;
        for (int i = 0; i < count; i++) {
            if (elems[i] < m) {
                mindex = i;
                m = elems[i];
            }
        }
        count--;
        elems[mindex] = elems[count];
        return m;
    }

    public void Add(int x) {
        if (count == elems.Length) {
            int[] b = new int[2*elems.Length];
            Array.Copy(elems, 0, b, 0, elems.Length);
            elems = b;
        }
        elems[count] = x;
        count++;
    }

}
```

Exercise 2

Create an Alloy model for the system described below:

(a) there are undergraduate students and graduate students, no student is both undergraduate and graduate student;

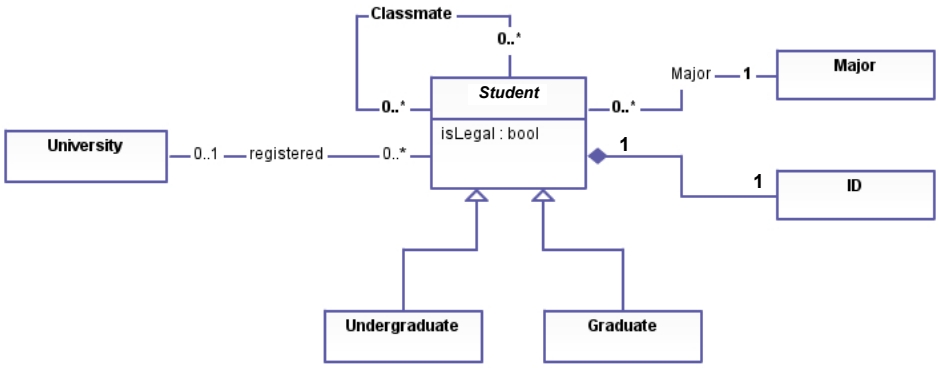
(b) a student should register at a university, and only registered students are legal students;

(c) every student has a unique student ID, and he or she has exactly one major;

(d) students with the same major are regarded as classmates, students can have several classmates.

(e) graduates and undergraduates are never classmates

Try to stick roughly to the UML-design from last weeks exercise:



Exercise 3

Download the .zip-file containing additional files from the course website. Open the files below and answer the questions in the comments:

1. Properties of binary relations. File: properties.als
2. Refactoring navigation expressions. File: distribution.als
3. Doris Day's song. File: everybody.als
4. Barber paradox. File: barber.als
5. Modeling the Tube. File: tube.als