

Assignment 3

Exercise 1

Find preconditions, postconditions and object invariants for the code on page 2. Express them using C# syntax.

Exercise 2

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. Modelling the Tube. File: tube.als
4. Doris Day's song. File: everybody.als
5. Barber paradox. File: barber.als

Exercise 3

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.

```

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

    [ContractInvariantMethod]
    private void Invariant() {
    }

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

```