Assignment 3 - Solution

Exercise 1

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

    [ContractInvariantMethod]
    void ObjectInvariant() {
        Contract.Invariant(elems != null);
        Contract.Invariant(0 < elems.Length);
        Contract.Invariant(0 <= count && count <= elems.Length);
    }
    public Bag(int[] initialElements) {
        Contract.Requires(initialElements != null);
        Contract.Requires(0 < initialElements.Length);
        ...
    }
    public Bag(int[] initialElements, int start, int howMany) {
        Contract.Requires(0 <= start && 0 < howMany);
        Contract.Requires(0 <= start + howMany);
        Contract.Requires(initialElements != null);
        Contract.Requires(start + howMany <= initialElements.Length);
        ....
    }
    public int RemoveMin() {
        Contract.Requires(0 < Count());
        ...
    }
    public void Add(int x) {
        ....
    }
}
```

Exercise 2

Answer files:

1. Properties of binary relations. File: properties_sol.txt
2. Refactoring navigation expressions. File: distribution_sol.txt
3. Modelling the Tube. File: tube_sol.txt
4. Doris Day's song. File: everybody_sol.als
5. Barber paradox. File: barber_sol.als

Exercise 3

Answer file: students.als