```
1  /**
2   * Kocsis Barna
3   * 2009
4   *
5   * IIO91110 .NET Programming
6   *
7   * The distance task 1: A pre-task Before Kick Start
8   */
9
10 using System;
11 using System.Collections.Generic;
12
13 namespace JAMK.IT.FinnishLotto
14 {
15   class Lotto
16   {
17     /// <summary>
18     /// The lover bound of that range from the lotto numbers are drwan.
19     /// </summary>
20     private readonly int min;
21
22     /// <summary>
23     /// The upper bound of that range from the lotto numbers are drwan.
24     /// </summary>
25     private readonly int max;
26
27     /// <summary>
28     /// The random number generator
29     /// </summary>
30     Random rand;
31
32     /// <summary>
33     /// Creates a new lotto.
34     /// </summary>
35     /// <param name="minimum">The inclusive lover bound of that range from the lotto numbers are drawn.</param>
36     /// <param name="maximum">The inclusive upper bound of that range from the lotto numbers are drawn.</param>
37     /// <exception cref="ArgumentException">Exception is thrown if minimum is greater than maximum.</exception>
38     public Lotto (int minimum, int maximum)
39     {
40       if (minimum >= maximum)
41         throw new ArgumentException("The minimum should be less than maximum.");
42
43       this.min = minimum;
44       this.max = maximum;
45
46       this.rand = new Random(DateTime.Now.Second + DateTime.Now.Millisecond);
47     }
48
49     /// <summary>
50     /// Draws numbers.
51     /// </summary>
52     /// <param name="count">The count of numbers to draw</param>
53     /// <param name="canBeEqual">A number can be draw more than once</param>
54     /// <returns>The drawn numbers</returns>
55     /// <exception cref="ArgumentException">
56     /// Exception is thrown if the range is less than the count to draw, and a number can
57     /// be drawn only once.
58     /// </exception>
59     public List<int> MakeADraw (int count, bool canBeEqual)
60     {
61       if (max - min < count && !canBeEqual)
62         throw new ArgumentException("There is not enough numbers to draw from.");
63
64       List<int> result = new List<int>(count);
65
66       for (int i = 0; i < count; ++i)
67       {
68         int number = this.rand.Next(this.min, this.max + 1);
69
```

```
70          while (!canBeEqual && result.Contains(number))
71            number = this.rand.Next(this.min, this.max + 1);
72
73          result.Add(number);
74        }
75
76        result.Sort();
77
78        return result;
79      }
80    }
81
82
83    class UI
84    {
85      /// <summary>
86      /// The number of draws displayed by this UI
87      /// </summary>
88      private int drawCounter;
89
90      /// <summary>
91      /// Creates a new UI
92      /// </summary>
93      /// <param name="min">The minimum</param>
94      /// <param name="max">The maximum</param>
95      public UI (int min, int max)
96      {
97        Console.WriteLine("Finnish Lotto");
98        Console.WriteLine("The numbers are drawn from the range [" + min + "," + max + "].↙
    \n");
99      }
100
101     /// <summary>
102     /// Writes the results of a draw to the screen
103     /// </summary>
104     /// <param name="results">A list containing the drawn numbers</param>
105     public void WriteDrawResult (List<int> results)
106     {
107       Console.Write(++drawCounter + " draw (" + results.Count + " numbers) : ");
108
109       for (int i = 0; i < results.Count - 1; ++i)
110         Console.Write(results[i] + ", ");
111
112       Console.WriteLine(results[results.Count - 1]);
113     }
114   }
115
116
117   class Program
118   {
119     /// <summary>
120     /// The entry point of this application
121     /// </summary>
122     /// <param name="args">The command line arguments</param>
123     static void Main (string[] args)
124     {
125       UI ui = new UI(1, 39);
126
127       int numberOfDraws = 1;
128
129       try
130       {
131         if (args.Length > 0)
132           numberOfDraws = Int32.Parse(args[0]);
133       }
134       catch
135       {
136         Console.WriteLine("Invalid argument: ." + args[1] + "\nRestoring number of draws↙
     to default (1).\n");
137
138         numberOfDraws = 1;
139       }
140
```

```
141       Lotto lotto = new Lotto(1, 39);
142
143       while (numberOfDraws-- > 0)
144         ui.WriteDrawResult(lotto.MakeADraw(7, false));
145
146
147       Console.ReadKey();
148     }
149   }
150 }
151
```