DENIS RAMSKIY
Group - II-11

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace GenApp
{
    public class Gen
    {
        public List<int> hromosom;
        public int fitnes;
        public double propability;
        static Random rnd;

        public Gen()
        {
            rnd = new Random((int)DateTime.Now.TimeOfDay.Milliseconds);

            hromosom = new List<int>();
            for (int i = 0; i < 1000; i++)
            {
                int nimber=(rnd.Next(i))%2;
                hromosom.Add(nimber);
                if (nimber == 1)
                     fitnes++;
            }
          }
        public static Gen operator +(Gen arg1, Gen arg2)
        {
            for (int i = 0; i < 1000; i++)
                arg1.hromosom[i] = arg2.hromosom[i];
            return arg1;
        }
    }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace GenApp
{
    class Generation
    {
        List<Gen> temp_generation;
        public int general_fitnes;

        public Generation()
        {
            temp_generation = new List<Gen>();
            for (int i = 0; i < 100; i++)
            {
                Gen temp = new Gen();
                temp_generation.Add(temp);
                general_fitnes += temp.fitnes;
            }
        }

        public void sort()
```

```
        {
            for(int i=0;i<100;i++)
                for(int j=0;j<99;j++)
                    if (temp_generation[j].fitnes < temp_generation[j + 1].fitnes)
                    {
                        Gen temp = new Gen();
                        temp += temp_generation[j];
                        temp_generation[j] += temp_generation[j + 1];
                        temp_generation[j + 1] += temp;
                    }
        }

        public void prop_gen()
        {
            for (int i = 0; i < 100; i++)
                temp_generation[i].propability =(double) temp_generation[i].fitnes /
general_fitnes;
        }

        public void next_gen()
        {
            List<Gen> new_generation=new List<Gen>();
            Random rnd = new Random((int)DateTime.Now.Ticks);
            List<Gen> per=new List<Gen>();

            for (int i = 0; i < 100; i++)
            {
                for (int j = 0; j < 50; j++)
                {
                    if (temp_generation[j].fitnes >= rnd.NextDouble())
                        per.Add(temp_generation[j]);
                    if (per.Count == 2)
                        break;
                }

                Gen rebonok = new Gen();
                for (int j = 0; j < 1000; j++)
                    if (rnd.Next()%2 == 1)
                        rebonok.hromosom[j] = per[0].hromosom[j];
                    else
                        rebonok.hromosom[j] = per[1].hromosom[j];

                new_generation.Add(rebonok);
            }

            for (int i = 0; i < 100; i++)
                temp_generation[i] += new_generation[i];
        }
    }
}
```

MAIN CLASS

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace GenApp
{
    class Program
    {
        static void Main(string[] args)
        {
```

```
            Generation mu_generation = new Generation();

            while (true)
            {
                mu_generation.prop_gen();
                mu_generation.sort();
                mu_generation.next_gen();
                Console.WriteLine(mu_generation.general_fitnes);
             }
        }
    }
}
```