

```

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

namespace labushka1
{
    class Program
    {
        class Chromosome
        {
            public int[,] gena = new int[100, 20];
            public Chromosome()
            {
                Random rand = new Random();

                for (int g = 0; g < 100; g++)
                {
                    for (int i = 0; i < 20; i++)
                    {
                        gena[g, i] = rand.Next(0, 2);
                        //Console.Write(gena[g, i]);
                    }
                    //Console.WriteLine();
                }
            }
            public int getFitness() //кол-во единиц в строке
            {
                int count = 0;
                foreach(int i in gena)
                {
                    if(i==1)
                        count++;
                }
                return count;
            }
        }

        public class Generation
        {
            Chromosome[] chromosomes = new Chromosome[100];
            Boolean isSorted = false;
            Generation()
            {
                for (int i = 0; i < chromosomes.Length; i++)
                    chromosomes[i] = new Chromosome();
            }

            int getAverageFitness ()
            {
                int cnt = 0;
                foreach (Chromosome i in chromosomes)
                    cnt += i.getFitness();
                return cnt / chromosomes.Length;
            }
            /* void sort()
            {
                sort(chromosomes, (o1, o2) ->
                {
                    int o1fit = o1.getFitness();

```

```

        int o2fit = o2fit.getFitness();
        if (o1fit < o2fit)
            return 1;
        else if (o1fit > o2fit)
            return -1;
        else
            return 0;
    });
    isSorted = true;
}
*/
Chromosome[] get2RandomChromosomes()
{
    if (!isSorted)
        sort();
    Chromosome[] random2 = new Chromosome[2];

    random2[0] = chromosomes[random2=50];
    random2[1] = chromosomes[random2=50];

    return random2;
}

Chromosome getBestChromosome()
{
    if (!isSorted)
        sort();
    return chromosomes[0];
}
}
static Random random;
/*static ArrayList<Generation> generations;
static ArrayList<Ineger> fitnessHistory;

static Chromosome crossover(Chromosome first, Chromosome second)
{
    int crossoverPoint = random.(first.gena.Length);
    Chromosome child = new Chromosome();

    System.arraycopy(first.gena, 0, child.gena, 0, crossoverPoint);
    System.arraycopy(second.gena, crossoverPoint, child.gena,
crossoverPoint, second.gena.Length - crossoverPoint);
}
*/

static void Main(string[] args)
{
    //запуск алгоритма
    random = new Random();
    /*fitness = new ArrayList<>();
    generations.Add(new Generation());
    */
    Console.WriteLine("Best chromo: ");

}
}
}

```