Kirill Tsibikov.

```h
#pragma once
#include <iostream>
#include <bitset>
#include <map>
#include <time.h>
using namespace std;


class GenEngn {
	map<double, bitset<1000>> popul;
public:
	GenEngn();
	void Run();
	void generateFirstGeneration();
	void selection();
	void crossing();
};
```

```cpp
#include "Header.h"

GenEngn::GenEngn()
{

}

void GenEngn::Run()
{
}

void GenEngn::generateFirstGeneration()
{
	srand(time(NULL));
	map<double, bitset<1000>> pop;
	do{
		bitset<1000> a;
		a.reset();
		for (int j = 0; j < 1000; j++)
			((rand() % 2) == 1) ? a.set(j,true) : a.set(j, false);
		pop[(double)a.count()] = a;
	} while (pop.size() < 100);
	double sum = 0.;
	for (map<double, bitset<1000>>::iterator i = pop.begin(); i != pop.end(); i++)
			sum += i->first;
	for (map<double, bitset<1000>>::iterator i = pop.begin(); i != pop.end(); i++)
		popul[i->first/sum]=i->second;
}

void GenEngn::selection()
{
}

void GenEngn::crossing()
{
}
```
