report for INFO-6205 FINAL PROJECT

1. **Problem Statement**

**1.1 Game of life**

Game of Life is an infinite, two-dimensional orthogonal grid of square cells, each of which is in one of two possible states, alive or dead, (or populated and unpopulated, respectively).Every cell interacts with its eight neighbours, which are the cells that are horizontally, vertically, or diagonally adjacent. At each step in time, the following transitions occur:

Any live cell with two or three neighbors survives.

Any dead cell with three live neighbors becomes a live cell.

All other live cells die in the next generation. Similarly, all other dead cells stay dead.

The game is a zero-player game, meaning that its evolution is determined by its initial state, requiring no further input. One interacts with the Game of Life by creating an initial configuration and observing how it evolves.

**1.2 Genetic algorithm**

Genetic algorithm is a method of searching for the optimal solution by simulating the process of natural evolution. The algorithms used by programs using genetic algorithms are basically the same. The key is to establish a suitable model for the current problem.

Some corresponding concepts in GA:

Genotype: the set of replicable and heritable information which is a property of a candidate solution (“organism”)—the genes;

Expression: the “mapping” of genotype to phenotype—the mapping between gene and trait does not have to be 1:1;

Phenotype: the traits of the candidate which affect how good a solution it is;

Fitness: how well the candidate solution solves the problem

1. **Solution to the problem**

**2.1 Approach**

For this problem, our basic steps are as follows:

1. Create initial population.

2. Calculate fitness.

3. Increase the algebra of population by 1.

4. Choose survivors and offspring instead.

5. Mutation of selected offspring.

6. Remove the expectation of death, combine the survivor's expectation and the mutation of the offspring to form a new population.

7.Repeat step 3 and step 6 until a certain end criterion is met.

**2.2 Key parameter design**

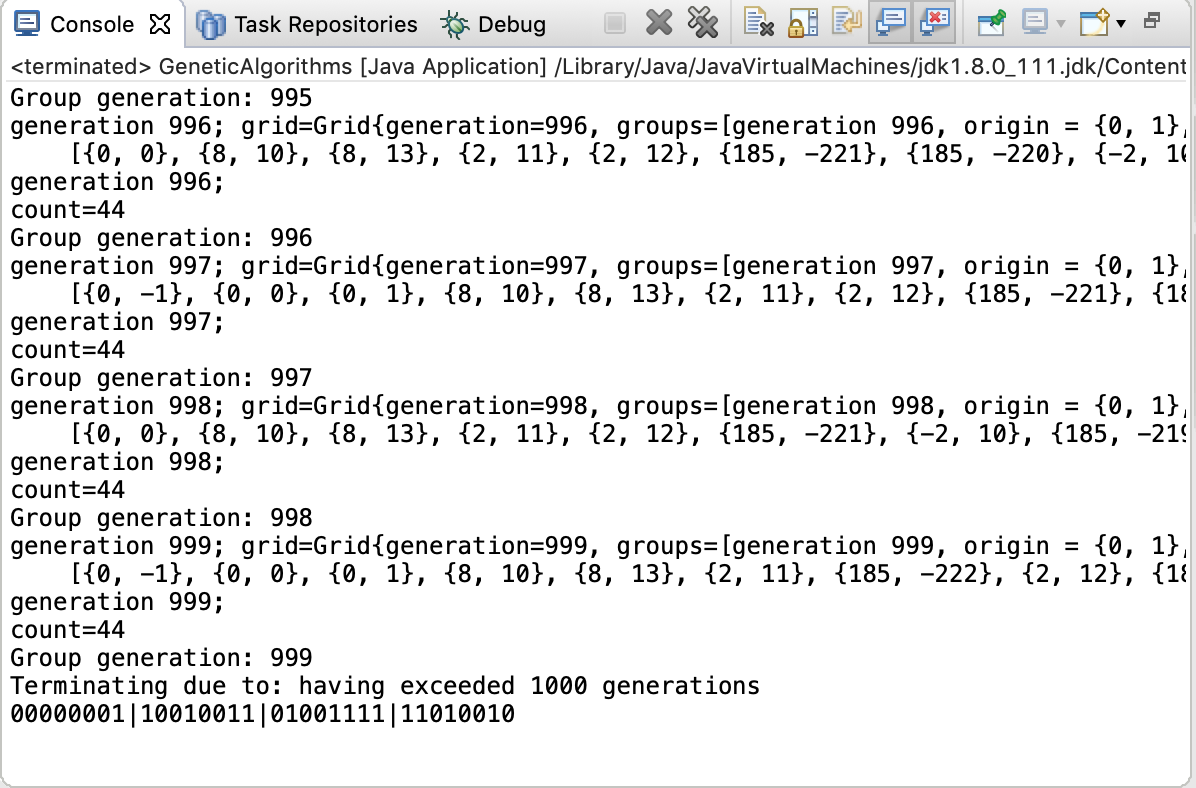
The genetic code and a random generator/mutator of such codes.

Gene expression: how do individual genes code for particular traits—ain the case of the game of life problem,we need to translate our bits into a set of points.

The fitness function--this is essentially a measure of how good a candidate (organism) solution is for the problem ; The survival function—to select the survivors according to their fitness.

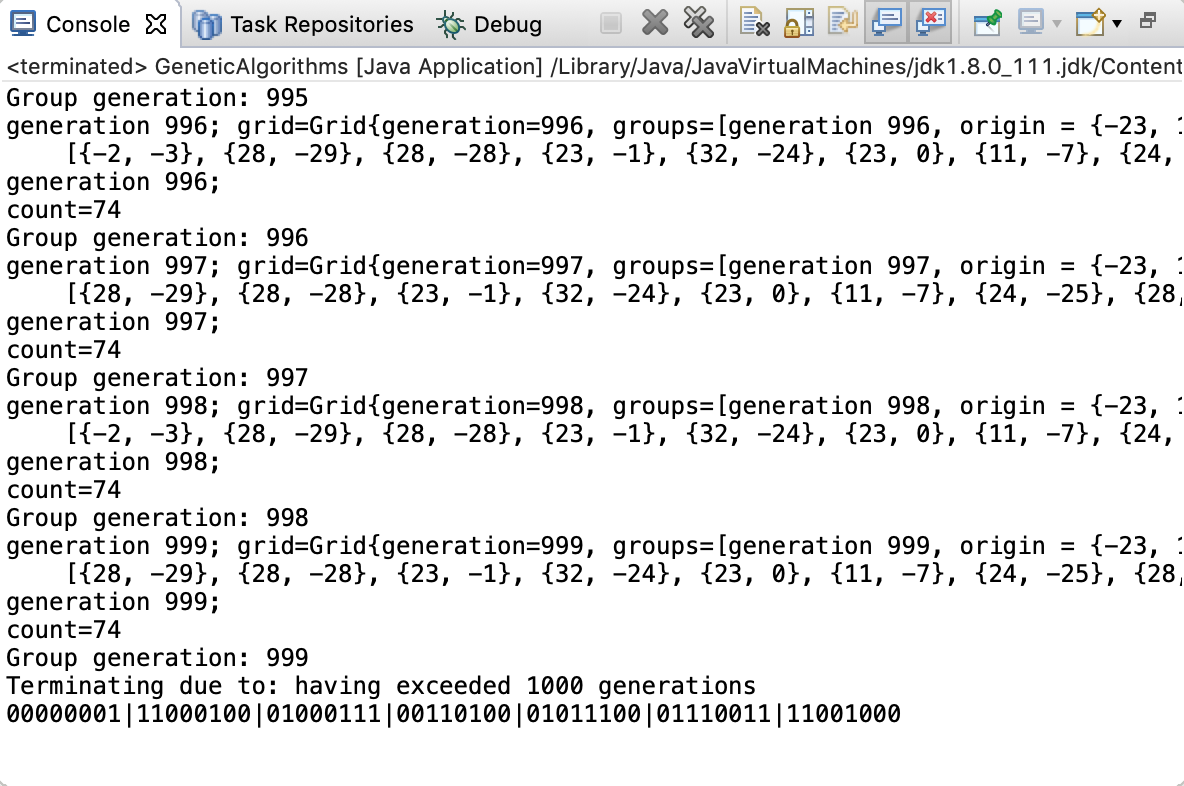
1. **Results**

Tablelength=5：

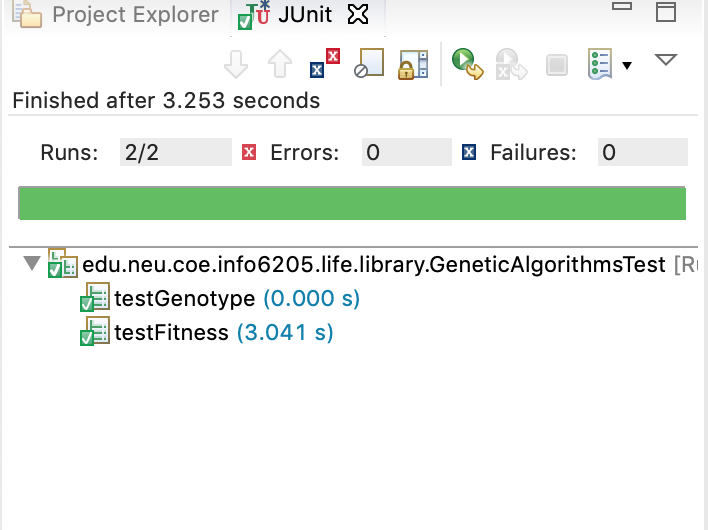
****

**Figure1 Running program screen shoot**

**Tablelength=7：**

****

**Figure2 Running program screen shoot**

****

**Figure2 unit tests screen shoot**

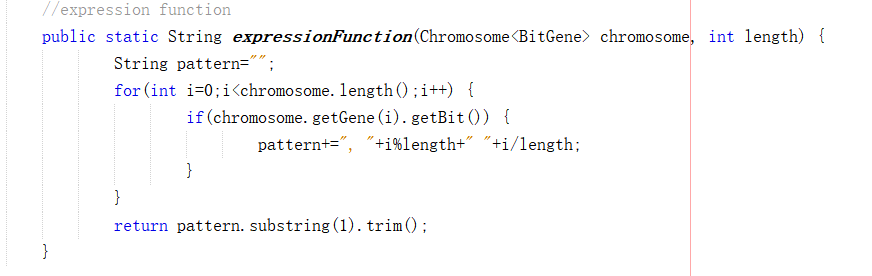
We can use a 4\*4 table to express this.Each square represents a Gene.In this problem,we uesd BitGene.

Genotype<BitGene> genotype=Genotype.of(BitChromosome.of(tableLength\*tableLength));

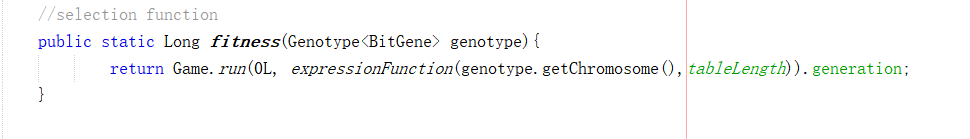
So 0 represents that the cell is dead and 1 represents that the cell is alive.The funtion is to generate bits, generate the entire genotype, and set the length and width of the table. The total length is the square of the tablelength, which constitutes the genotype.

Expression function: This function is to convert the genotype into the initial pattern.If the cell is alive,

The abscissa x represents the number of bits in the chromosome.



Fitness funtion:The initial population given is 1000.This funtion is to select the initial pattern that can run to 1000 generations,which is also the pattern we want to find.



Mutation function,Candidate selection function and Genetic algorithm library code we used the code library from Jenetics. And the initial pattern is randomly generated using the library in jenetics.

1. **Findings & Conclusion**

Through this program,we can find relatively good and excellent inital pattern.