Final Project Report

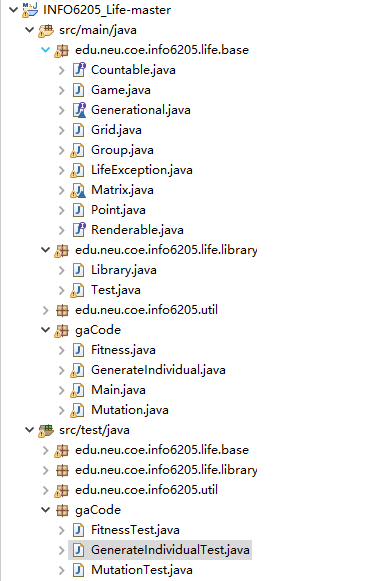
001087420 Yajie Li

001057288 Jiaqi Zhang

## Overview

We chose to use genetic algorithm to do Conway life game. We did not use Jenetics library but chose to write our own genetic algorithm, including generating individuals’ genotype and phenotype, selecting based on generations and mutating genotype randomly.

## Structure of program



Picture1

As picture1 shown, basically the main program contains two parts: one is the life game program provided by Prof. Robin, the other part is the GA code which contains GenerateIndividual.java, Fitness.java, Mutation.java and Main.java.

Main.java. is the main class and users should run it to run the program, it invokes all the methods containing in other java files.

GenerateIndividual.java will generate population base on the seed pass from Main class, the seed is got by current time in millisecond.

Fitness.java contains a method called selection which will remove bad individuals from the population based on the result of run method in Game.java, just like the environment kills weak animals in the natural.

Mutation.java contains a method called mutate which will do mutation to every individual’s gene, then the genotype will be convert to phenotype and passed to following selection.

Main.java invokes all the method written in other files and give an output.

The other part of program is testing code which tries to ensure the code works as we hope.

GenerateIndividualTest.java test the number of individuals that the geneIndi method create.

FitnessTest.java uses the data from Library.java which has obviously 13 generations and then test if the individual is selected under different standard.

## How the program works

To running the program, users just need to run Main.java in gaCode package. In the Main.java file, there first initializes a seed base on the system time and then uses this seed to generate a population contains 1000 individual. Then the Main.java file invokes selection method and mutation method many times (more and more strict) to find good patterns. After selection and mutation, the program will automatically output the final string pattern which pass all the selection and its original phenotype.

To test the program users just need to run different test java files in test package.

#1 What’s our gene like?

In the program, to generate genotype, we randomly generate 0 or 1 bits and form them into a string,

The genotype thus is like 011110101001. We save the genotype in HashMap<Integer, String>.

#2 How to transfer genotype to phenotype

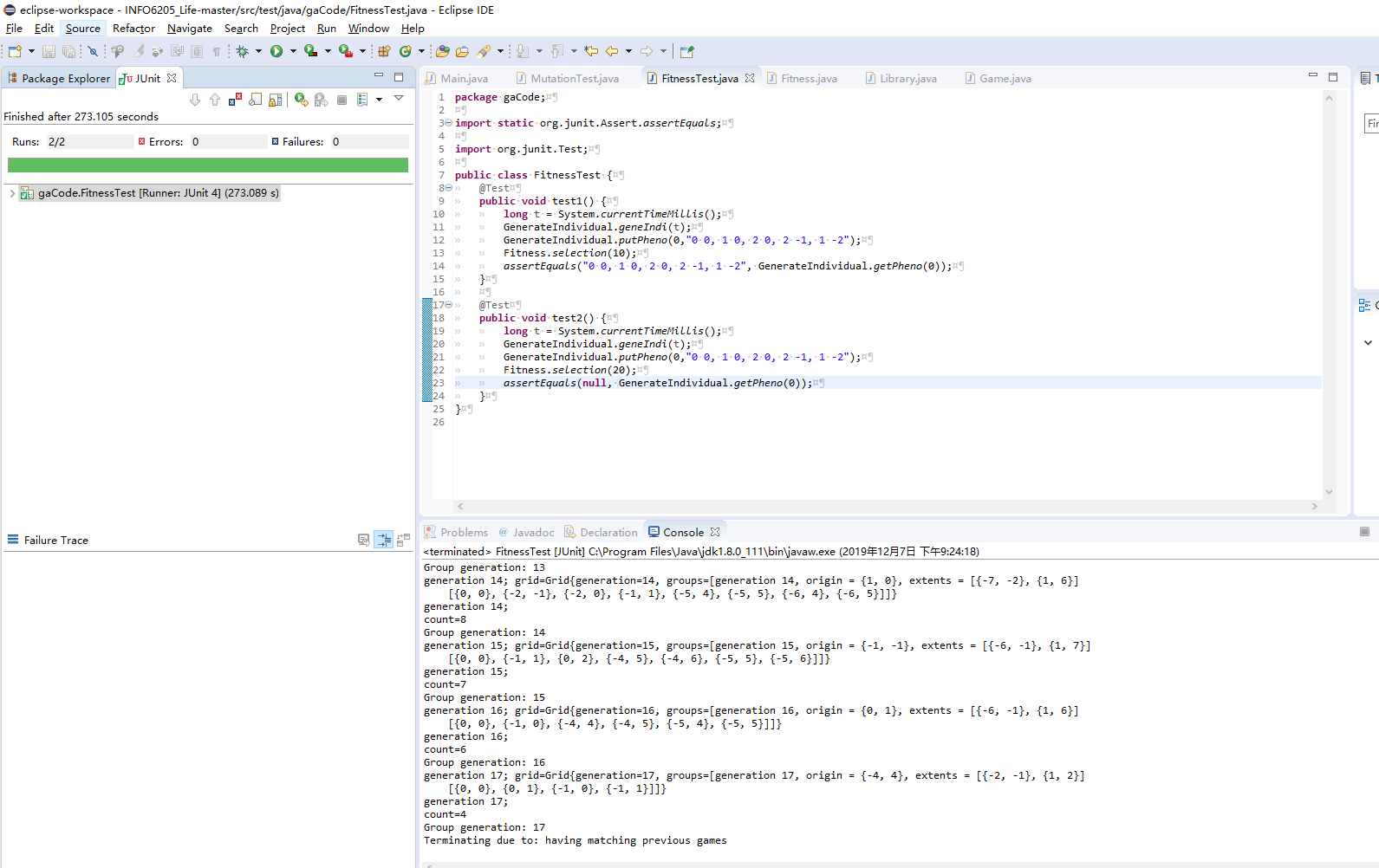
that is the chromosome, then we convert the genotype to phenotype by covert numbers in 2 binary system into decimal system. For example, the gene is 011|110|101|001, then we separately convert 011, 110, 101, 001 into “3 2, 5 1” as phenotype, save them in a HashMap and then pass the value to run() method.

#3 Fitness

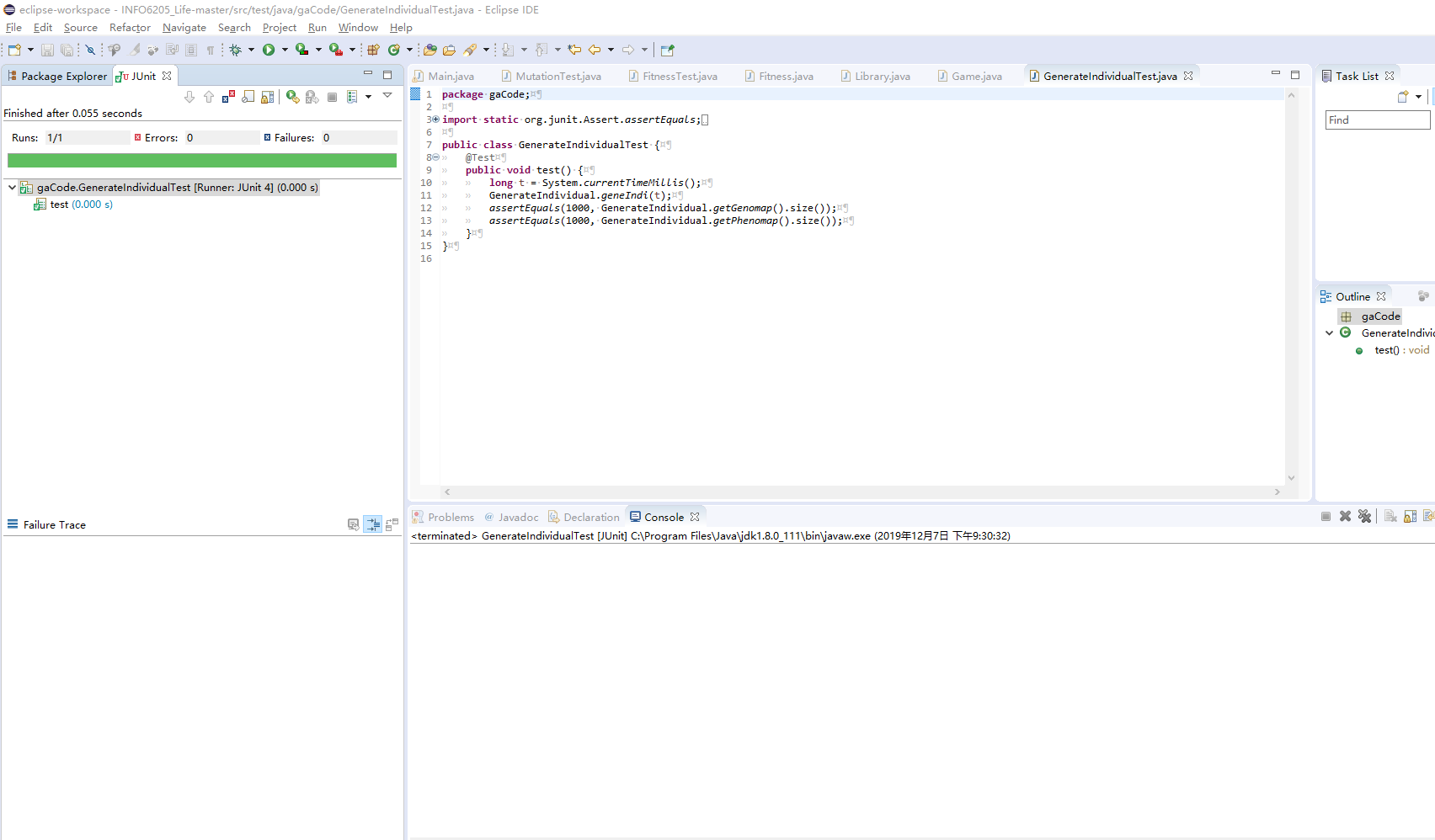
After running run() method by using coordinates generate above, we get the generation of each pattern and judge weather a pattern is good based on the numbers of generations, the longer the game last, the better the pattern is. If a individual is not good, it will be killed, so that pattern will be removed from both genotypemap and phenotypemap.

## Unit Test result

The following pictures will show that we successfully pass the unit-test



Picture2



Picture3

Picture 2 shows that we pass the FitnessTest and picture 3 shows that we pass the GenerateIndividualTest

## Conclusion

By using this program we successfully find good pattern which can last very long in Conway life game.

The seed is: 1575768638128

The program generate 1000 individuals and do selection and mutation to them

The result string pattern(Phenomap) which fit the requirement:

4 2, 7 6, 7 5, 3 2, 1 5, 7 3, 5 4, 6 3, 2 1, 2 5, 5 7, 5 0, 5 4, 4 7, 6 7, 7 4, 0 1, 2 5, 2 2, 1 3, 5 7, 2 1, 7 1, 3 3, 5 5, 5 4, 7 4

The origin string pattern(Phenomap) of the pattern above:

0 2, 6 6, 7 6, 3 2, 3 5, 3 3, 5 1, 6 4, 4 5, 2 6, 5 6, 6 5, 6 6, 5 7, 6 7, 6 6, 7 2, 0 6, 2 2, 0 7, 4 7, 6 0, 2 1, 3 3, 5 0, 4 1, 7 7

By this project we coded a simple GA program and use it to find a good solution to the life game. We also basically understand the concepts of GA including genotype, phenotype, population, selection, mutation, etc.