

Preliminary Class Diagram

Simulation of Mendelian Laws of Genetics

Programming Assignment 2

Date: March 11th, 2018

Prepared by

Sean Mitchell

sm0077@uah.edu

Prepared for

Dr. Rick Coleman

CS 307 - 01: Object Oriented Programming in C++

Computer Science Department

University of Alabama in Huntsville

1 UML Diagram

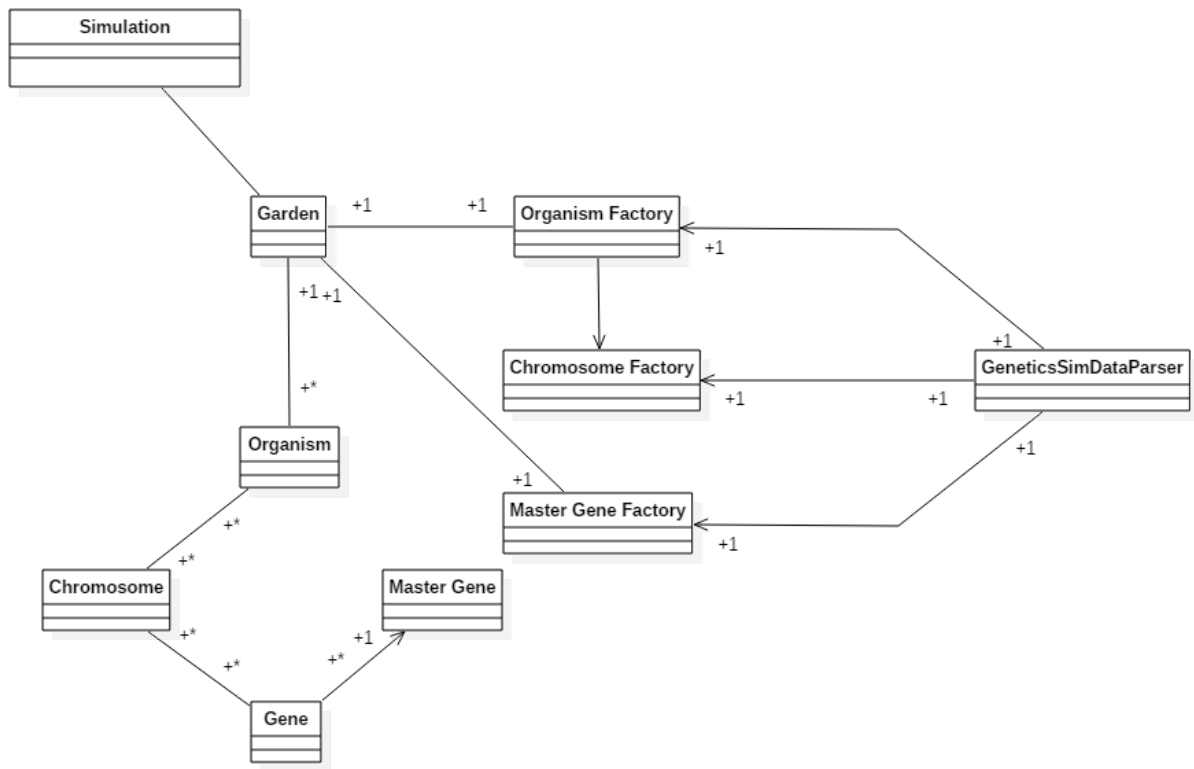


Figure 1: Preliminary UML Class Diagram

2 Class Descriptions

Simulation The Simulation class will perform three main tasks. The first task will be to ask the user how many offspring to create. The second task will be asking the user the name of the data file to read as input. Using the input from the first two tasks, the Simulation object will create an instance of Garden, passing the number of children to create and the input file as arguments.

Garden The Garden class will perform several tasks. The first task will use the data file passed as input to create two parent Organism objects. Once these two parents have been created, the Garden object will allocate data structures to store the children. Next, the Garden object will breed the two parents until the number of offspring created matches the input received from the Simulation object. Once all the offspring are created, the Garden object will sort the data structures. Finally the Garden object will return a report of the results of the simulation.

Parser The Parser object will read the attributes of each gene. This object will be called by the factories to return the data needed to for the factories to create their instances.

Organism The Organism class will be an instance of the object returned from the Organism Factory. The class will contain a Chromosome vector which will store all the Chromosome objects that the Organism has. Other member variables that class will have include both the common name and scientific name. The class will also have accessor and mutator functions to these member variables.

Organism Factory The Organism Factory will perform the creation of instances of Organism. The Organism Factory class is a simple factory. The Organism Factory shall use the Chromosome Factory to create chromosomes for organisms. The Organism Factory shall contain a function which returns an instance of an organism with fully defined genotype. This factory will also be a Singleton.

Chromosome The Chromosome class will be an instance of the object returned from the Chromosome Factory. Instances of Chromosome will store the genes for that particular chromosome.

Chromosome Factory The Chromosome Factory class will perform the creation of instances of chromosomes. This class will be a simple factory. The Chromosome Factory shall use the Master Gene Factory to create genes for chromosomes. This factory will also be a Singleton.

Gene The Gene class will be an instance of the object returned from the Gene Factory. The Gene class will hold a reference to the master gene and variables to hold the specific allele characters for that instance of the gene.

Master Gene The Master Gene class is a single instance for each type of gene. This instance will hold most of the information that is common to many genes.

Master Gene Factory The Master Gene Factory class will perform the creation of instances of both Master Gene and Gene. This class is a simple factory. This factory will also be a Singleton.