**Class Outline**

**Mendelian Simulation**

**Programming Assignment 2**

**March 19, 2018**

**Prepared By**

**Atchima Klomkaew**

**ak0061@uah.edu**

**Prepared for**

**Dr. Rick Coleman**

**CS 307, Object Oriented Programming**

**Computer Science Department**

**University of Alabama in Huntsville**

1. **System Overview………………………………………….…………………….…3**
2. **Relevant Terms and Acronyms………………………….……….…………….…3**
3. **Object Description....…………………………………….………..……………….3**
   1. **MendelianSimulation**
      1. **Member Variables……………………………………………...3**
      2. **Member Functions……………………………………………..3**
         1. **Function *InitializeSimulation*...………………………...3**
         2. **Function *RunSimulation*……………………..………...3**
   2. **OrganismFactory**
      1. **MemberVariables……………………………………………...4**
      2. **MemberFuncitons……………………………………………..4**
         1. **Function *CreateOrganisms*………………………..…...4**
         2. **Function *PrintParentData*………………………..…....4**
   3. **Organism**
      1. **Member Variables……………………………………………...4**
      2. **Member Functions……………………………………………..4**
         1. **Function *LoadData………....*...………………………...4**
   4. **MasterGeneFactory**
      1. **Member Variables……………………………………………...4**
      2. **Member Functions……………………………………………..4**
         1. **Function *CreateMasterGenes*...………………………...4**
   5. **MasterGene**
      1. **Member Variables……………………………………………...5**
      2. **Member Functions……………………………………………..5**
         1. **Function *LoadData………....*...………………………...5**
         2. **Function *PrintData………....*...………………………...5**
   6. **ChromosomeFactory**
      1. **Member Variables……………………………………………...5**
      2. **Member Functions……………………………………………..5**
         1. **Function *CreateChromosomes..*...……....……………...5**
   7. **Chromosome**
      1. **Member Variables……………………………………………...5**
      2. **Member Functions……………………………………………..5**
         1. **Function *CreateGenes………...*...……....……………...5**
         2. **Function *LoadData…………....*...……....……………...6**
   8. **Gene**
      1. **Member Variables……………………………………………...6**
      2. **Member Functions……………………………………………..6**
         1. **Function *LoadData…………....*...……....……………...6**
   9. **Offspring**
      1. **Member Variables……………………………………………...6**
      2. **Member Functions……………………………………………..6**
         1. **Function *LoadData…………....*...……....……………...6**
   10. **OffspringReader**
       1. **Member Variables……………………………………………...6**
       2. **Member Functions……………………………………………..6**
          1. **Function *PrintOffspring…………....*.......……...6**
   11. **Breed**
       1. **Member Variables……………………………………………...6**
       2. **Member Functions……………………………………………..6**
          1. **Function *GenerateOffspring…………....*....…...6**
          2. **Function *GetCorrectFormGene…………...*…...7**
   12. **BinaryTree**
       1. **Member Variables……………………………………………...7**
       2. **Member Functions……………………………………………..7**
          1. **Function *SearchTree…………....*...…….……...7**
          2. **Function *Insert....*...………………………..…...7**
          3. **Function *Insert....*...……………………..……...7**
          4. **Function *PrintOne....*...……………....………...7**
          5. **Function *PrintTree....*...……………....………...7**
4. **System Overview**

This project shall stimulate the processes of inheritance incorporating the Mendelian laws of genetics. The simulation shall read from the input file to identify the types of chromosomes, gene, genes’ definitions along with its crossover chance, and parents’ genotypes. It shall request the user to determine the total amount of offspring to generate. The parents’ genes will cross with one another in order to form an offspring, and the result will be similar to Mendel’s pea plants simulation (second generation). The phenotype ratio of the result is 3:1 for dominant gene and recessive gene respectively. The genotype ratio is 1:2:1 for homozygous dominant, heterozygous dominant, and homozygous recessive respectively. The output of this simulation shall include the amount of offsprings’ genotype generated for each gene, which should be in the ratio mentioned.

1. **Relevant Terms and Acronyms**

Mendelian laws of genetics include law of segregation, law of independent assortment, and law of dominance. The upper-case letter signifies dominant allele while lower-case letter signifies recessive allele. Genotype is the genetic makeup of an organism, i.e. TT Ww Ss cc. On the other hand, phenotype is the organism’s physical appearance, e.g. tall, wrinkled, green, and white. The crossover chance depends on the distance between the two genes in one strand of chromosome. The greater the distance, the greater the chance of the genes to get crossed over.

1. **Object Description**
   1. **MendelianSimulation**
      1. Member Variables

**int numOffsprings** - *number of offspring*

**GeneticsSimDataParser parser** - *instance of GeneticsSimDataParser, which will read the data from xml file*

**OrganismFactory orgFac** - *instance of OrganismFactory where both the parent organisms will be stored*

**MasterGeneFactory masterGeneFac** - *description*

**ChromosomeFactory chromoFac** - *instance of ChromosomeFactory where all the chromosomes will be stored*

**Breed mate** - *instance of Breed. Two instances of organisms will be passed to this instance in order to generate offsprings*

**OffspringReader reader** - *instance of OffspringReader which will generate proper outputs of the results*

* + 1. Member Functions
       1. **Function *InitializeSimulation***

Actions Performed - initializes almost all instances in order to read from file, store definition and data properly in factories

Arguments - not applicable

Return Value - not applicable

* + - 1. **Function *RunSimulation***

Actions Performed - prompts the user for number of offspring to generate, calls a function *Execute* in Breed in order to generate offspring, and calls *PrintOffspring* in OffspringReader to print a proper form of the result

Arguments - not applicable

Return Value - not applicable

* 1. **OrganismFactory**
     1. Member Variables

**vector<Organism> orgCollection** - *collection of Organism instances*

* + 1. Member Functions
       1. **Function *CreateOrganisms***

Actions Performed - generate instances of Organism, calls its *LoadData* function, and then adds it to the vector *orgCollection*

Arguments - the index of parent obtaining

Return Value - an instance of the indicated parent

* + - 1. **Function *PrintParentData***

Actions Performed - prints both parents

Arguments - not applicable

Return Value - not applicable

* 1. **Organism**
     1. Member Variables

**Chromosome chromo** - *chromosome (single strand)*

**vector<Chromosome> chromoPair** - *technically one chromosome but represents 2 strands*

**int chromoNum** - *number of chromosomes*

**String genusName** - *genus name*

**String speciesName** - *species name*

**String scientificName** - *scientific name*

**String commonName** - *common name*

* + 1. Member Functions
       1. **Function *LoadData***

Actions Performed - loads information of the organism from dataParser

Arguments - instance of GeneticsSimDataParser

Return Value - not applicable

* 1. **MasterGeneFactory**
     1. Member Variables

**vector<MasterGene> mgCollection** - *collection of MasterGene instances*

* + 1. Member Functions
       1. **Function *CreateMasterGenes***

Actions Performed - generate instances of GeneMaster, calls its *LoadData* function, and then adds it to the vector *mgCollection*

Arguments - instance of GeneticsSimDataParser

Return Value - not applicable

* 1. **MasterGene**
     1. Member Variables

**char traitName** - *trait name*

**char domAllele** - *dominant allele*

**char domSym** - *dominant symbol*

**char recAllele** - *recessive allele*

**char recSym** - *recessive symbol*

**char homoDom** - *characters of homozygous dominant*

**char hetero** - *characters of heterozygous dominant*

**char homoRec** - *characters of homozygous recessive*

**int geneCount** - *number of genes*

* + 1. Member Functions
       1. **Function *LoadData***

Actions Performed - loads information of the organism’s gene definitions from dataParser

Arguments - instance of GeneticsSimDataParser

Return Value - not applicable

* + - 1. **Function *PrintData***

Actions Performed - prints the data in proper format

Arguments - not applicable

Return Value - not applicable

* 1. **ChromosomeFactory**
     1. Member Variables

**vector<Chromosome> chromoCollection** - *collection of Chromosome instances*

* + 1. Member Functions
       1. **Function *CreateChromosomes***

Actions Performed - generate instances of Chromosome, calls its *LoadData* function, and then adds it to the vector *chromoCollection*

Arguments - instance of GeneticsSimDataParser

Return Value - not applicable

* 1. **Chromosome**
     1. Member Variables

**vector<Gene> geneCollection** - *vector collection of genes contained in this strand of chromosome*

**String strand** - *a strand of the chromosome*

* + 1. Member Functions
       1. **Function *CreateGenes***

Actions Performed - generate instances of Gene, calls its LoadData function, and then adds it to the vector *geneCollection*

Arguments - instance of GeneticsSimDataParser

Return Value - not applicable

* + - 1. **Function LoadData**

Actions Performed - loads information of the organism’s chromosome from dataParser

Arguments - instance of GeneticsSimDataParser

Return Value - not applicable

* 1. **Gene**
     1. Member Variables

**char allele1** - *the first allele of the gene*

**char allele2** - *the second allele of the gene*

* + 1. Member Functions
       1. **Function *LoadData***

Actions Performed - loads information of the organism’s gene from dataParser

Arguments - instance of GeneticsSimDataParser

Return Value - not applicable

* 1. **Offspring** -- inherits from Organism
     1. Member Variables

**Chromosome chromo** - *chromosome (single strand)*

**vector<Chromosome> chromoPair** - *technically one chromosome but represents 2 strands*

**int chromoNum** - *number of chromosomes*

**String genusName** - *genus name*

**String speciesName** - *species name*

**String scientificName** - *scientific name*

**String commonName** - *common name*

* 1. **OffspringReader**
     1. Member Variables

**MasterGeneFactory masterGeneFac** - *instance of MasterGeneFactory for definition of genes*

**int offspringCount** - *number of offspring*

* + 1. Member Functions
       1. **Function *PrintOffspring***

Actions Performed - print all offspring in the correct format

Arguments - instance of BinaryTree

ReturnValue - not applicable

* 1. **Breed**
     1. Member Variables

**String allele1** - *the first allele*

**String allele2** - *the second allele*

**Chromosome chromo** - *instance of chromosome*

**vector<Offspring> offsprings** - *collection of instances of Offspring generated*

**OrganismFactory orgFac** - *instance of OrganismFactory*

* + 1. Member Functions
       1. **Function *GenerateOffspring***

ActionsPerformed - asks each parent for a chromosome, generates x number of offsprings, and stores them in a BinaryTree

Arguments - BinaryTrees, OrganismFactory, numOffspring, numChromo

ReturnValue - not applicable

* + - 1. **Function** ***GetCorrectFormGene***

Actions Performed - ensure uppercase letter/dominant allele is precedent of lowercase letter/recessive allele

Arguments - not applicable

Return Value - not applicable

* 1. **BinaryTree**
     1. Member Variables

**Struct Node** - *contains the key (genotype of offspring) and counter*

**Node root** - *the root node of the tree*

* + 1. Member Functions
       1. **Function *SearchTree***

ActionsPerformed - searches for a node in the tree with the given key value

Arguments - the key to search for

ReturnValue - a node containing that key or null if not found

* + - 1. **Function *Insert***

ActionsPerformed - inserts a new node by key value, which calls another Insert function that inserts by node

Arguments - the key value to inset

ReturnValue - not applicable

* + - 1. **Function *Insert***

ActionsPerformed - inserts by node, which received value from the previous Insert function

Arguments - the node to insert

ReturnValue - not applicable

* + - 1. **Function *PrintOne***

ActionsPerformed - prints a single node

Arguments - a node to print

ReturnValue - not applicable

* + - 1. **Function *PrintTree***

ActionsPerformed - prints all nodes. In-order traversal

Arguments - not applicable

ReturnValue - not applicable

**There will be get and set functions defined for each private member variable.**