Adam Cox

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* ChromosomeDriver.Java

\* <Adam Cox>

\*

\* This program takes 3 chromesomes and changes some information in them

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**public** **class** ChromosomeDriver

{

**public** **static** **void** main(String[] args)

{

Chromosome c1, c2, c3; // chromosomes

c1 = **new** Chromosome(**new** **int**[] {1, 2, 3, 4, 5, 6});

c2 = **new** Chromosome(**new** **int**[] {11, 12, 13, 14, 15, 16});

c3 = c1.replicate();

c3.mutate();

System.***out***.println("c3 = " + c3.getData());

c1.crossover(c2);

System.***out***.println("c1 = " + c1.getData());

System.***out***.println("c2 = " + c2.getData());

} // end main

} // end ChromosomeDriver

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Chromosome.Java

\*

\* The various methods that will be used by ChromosomeDriver

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**import** java.util.Random;

**public** **class** Chromosome

{

**public** **int**[] genes; // contains genes

Random rand = **new** Random(); // generate random number

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**public** Chromosome(**int**[] gene)

{

**this**.genes=gene;

} // end Chromosome constructor

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Mutates a random gene to a random value between 0 and max int

**public** **void** mutate()

{

**int** selectedGene; // Random gene selected to be replaced

**int** geneValue; // Random number replacing gene

selectedGene = rand.nextInt(genes.length);

geneValue = rand.nextInt(Integer.***MAX\_VALUE***);

genes[selectedGene] = geneValue;

} // end mutate

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Replicates genes from one strand to another

**public** Chromosome replicate()

{

**int**[] backup = **new** **int**[genes.length]; // stores gene values

**for** (**int** i=0; i<genes.length; i++) // allows values to transfer

{

backup[i]=genes[i];

} // end for;

**return** **new** Chromosome (backup);

} // end replicate

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Changes 2 chromosomes at a random int

**public** **void** crossover(Chromosome c2)

{

**int** selectedGene; // Number of gene selected for change

**int** backup = 0; // Backups the first list

selectedGene = rand.nextInt(genes.length - 1);

**for** (**int** i=selectedGene+1; i<genes.length; i++) // Loop that changes the genes

{

backup = **this**.genes[i];

**this**.genes[i] = c2.genes[i];

c2.genes[i] = backup;

} // end for

} // end crossover

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// returns the values of the chromosomes into a string

**public** String getData()

{

String info = ""; // used to return the numbers

**for** (**int** i=0; i<genes.length; i++) // puts numbers into string with a ,

{

info += genes[i];

**if**( i < (genes.length - 1))

{

info += ", ";

}

} // end for

**return** info;

} // end getData

} // end class Chromosome