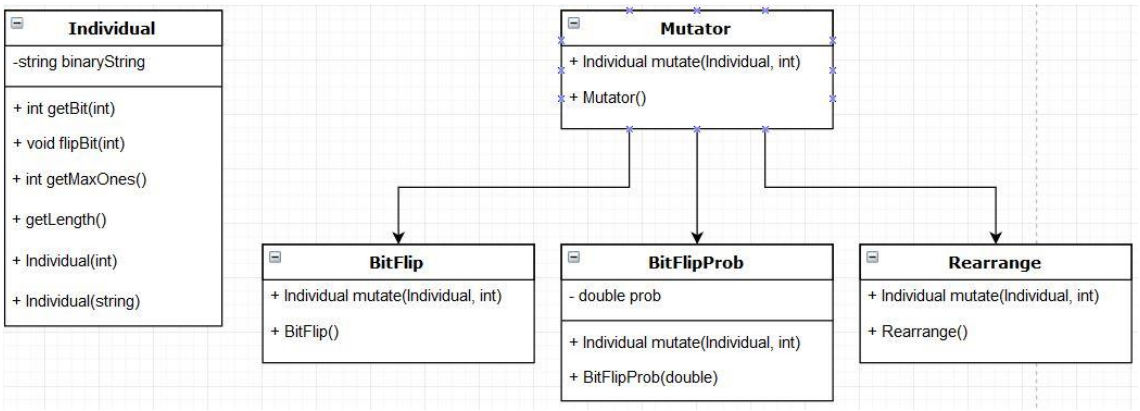
ADDS Prac 7 Design

Hideki Yoshinaga | a1658945

UML Diagram



Description

- Individual
  - String binaryString： his string stores 1s and 0s for each individual
  - Void flipBit：This function takes an integer and flip the bit(0to1 or 1to0) at the position of that integer in binaryString.
  - Int getMaxOnes：The function returns the longest consecutive sequence of '1' digits in binaryString (e.g. calling the function on "1001110" will obtain 3).
  - getLength：The function returns the length of binaryString
  - Individual(int length):A constructor that takes in the length of the binary DNA and creates the binarySring. Each binary value in the list should be given a value of 0 by default.
  - Individual(string)：This constructor that takes in a string and copy it to binaryString.
- Mutator
  - Individual mutate(Individual int) ：This function is a virtual function, takes in an Individual and an integer index k as parameter and returns the offspring after mutation.
- BitFlip
  - Individual mutate(Individual int)：This function takes in an

Individual and an integer index k as parameter and returns the offspring with position k's bit fliped.

- BitFlipProb
  - Double prob : this holds the probability in value between 0~1
  - Individual mutate(Individual int) : This function takes in an Individual and an integer index k as parameter and returns the offspring with position k's bit fliped in a predefined probability
  - BitFlipProb(double) : the constructor takes in a double value and pass it to prob. This will decide the probability of mutation.
- Rearrange
  - Individual mutate(Individual int) : the mutate function rearranges the list. The function will select the k-th digit in the bitstring (again, counting in circles). This digit and all digits after it (all the way to the tail) will be moved to the start of the list. For example, if you were rearranging the list (a,b,c,d,e) and k = 3, the function would return an Individual with the list (c,d,e,a,b).
- Main
  - Collect a string vector from user input and sparate values into strings and integers. Use these strings to create two Individuals , BitFlip mutator and Rearrange mutator. Apply BitFlip mutator to the first Individual with the first integer, apply Rearrange mutator to the second Individual with the second integer. Use getString function to cout the result of these binaryStrings and use getMaxOnes function to count the max ones in second binaryString. Results are printed with a space in between.

Testing

1.
   i. Input: 111000 2 000111 2
   ii. Output: 101000 001110 3

This will check if the bitFlip and Rearrange and getMaxOnes are working correctly

2.
   i. Input: 000001 8 011100 9
   ii. Output : 010001 110001 2

This will check when k is larger than string length, it will over lap or not.

3.

i. Input : 0 5 1 2

ii. Output : wrong input

When binaryString only has length of 1, it can not go over lap, should give an error message.