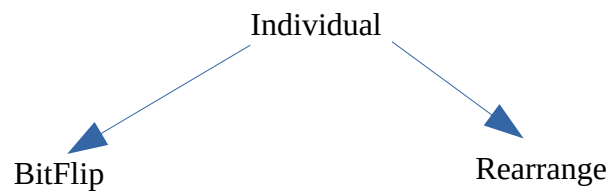


## Design Prac 8

### UML



#### BinaryNode

Bool x – saves the binary digit

BinaryNode\* next – saves the next position of the linked list

Bool getX – returns the boolean

BinaryNode\* getNext – returns the next

- This is to create a linked list
- will save what is inputted

#### Individual

BinaryNode\* firstBit – sets the head

string binaryStr – stores the binary string

int listLength – stores the length of the list

String getString() - returns the binary string

BinaryNode\* getFirstBit() - returns the head

void setFirstBit(BinaryNode\* newHead) – sets the head

Int getMaxOnes() - returns the max number of 1's in a row

int getLength() - returns the length of the list

virtual void execute(int k) – will be defined in sub classes

- A constructor which will take in a string and will create a linked list
- another constructor that will create a list from a length
- A destructor which will iterate over the list and delete each binary node
- stores the head of the list, the binary string and the lists length
- will include the header file for binary node
- will not be able to be made due to virtual function
- has many functions to act on the list

#### BitFlip

void execute(int k) – will flip the digit at k

- Allows the list to be altered by defining execute
- will be a sub class of individual

## Design Prac 8

<b>Rearrange</b>
void execute(int k) - will rearrange the list

- Allows the list to be altered by defining execute
- will be a sub class of individual

### Main:

- will take in input as a string and an integer
- will construct a BitFlip and a Rearrange object
- will then call execute on both of these objects
- will print out the lists after they have been altered

### Testing:

Input 1:

11110000 2 1010 3

Output 1:

10110000 1010

Input 2:

101 4 101 5

Output 2:

001 011

Input 3:

00000 1 10 2

Output 3:

10000 01

Input 4:

11111 5 1100 6

Output 4:

11110 0011

Input 5:

1 1 1 1

Output 5:

0 1