Assignment 3, Final design

**cryptoManager class**

1. Method Boolean stringinbounds that accepts string variable plain text
   1. Start for loop
   2. For integer i initialized to 0, if i is less than plainText length, increment i
      1. if plainText character at index i is less than LOWER\_BOUND or plainText character at index i is larger than UPPER\_BOUND
      2. return false
      3. exit loop
   3. return true
2. Method for string encrypCaesar that accepts string plain text and integer key
   1. if key is larger than UPPER\_BOUND
      1. do decrement key by range while key is larger than UPPER\_BOUND
      2. if key is larger than RANGE
         1. decrement key by range
   2. Declare new instance character array called stringArr with plainText length
      1. Begin for loop
      2. Declare integer variable i initialized to 0, if i is less than the length of plainText, increment i
         1. Initialize stringArr at index i to stringArr at index i plus key (type convert to char)
         2. if stringArr at index i is greater than UPPER\_BOUND
            1. decrement stringArr at index i by RANGE
         3. if stringArr at index i is less than LOWER\_BOUND
            1. increment stringArr at index i by RANGE
   3. Declare string instance enryptString and initialize with stringArr that is concatenated
   4. Return encryptString
3. Declare string method encryptBellaso that accepts string plaintext and string bellasoStr
   1. while plainText length is larger than bellasoStr length
      1. increment bellasoStr by itself
   2. create new instance for char array plainTextArr to hold array with plainText length
   3. Begin for loop
      1. Declare integer variable i initialized to 0, if i is less than the length of plainText, increment i
      2. initialize plainTextArr at index i to char type plainText character at index i plus bellasoStr character at index i minus RANGE
      3. check range
         1. if plainTextArr is larger than UPPER\_BOUND, initialize plainTextArr to itself minus RANGE
      4. check range
         1. if plainTextArr is less than LOWER\_BOUND, initialize plainTextArr to itself plus RANGE
   4. declare string instance encryptString and initialize to concatenate stringArr
   5. return encryptString
4. Method decryptCaesar that accepts encryptedText and int key
   1. if key is larger than UPPER\_BOUND
      1. do decrement key by range while key is larger than UPPER\_BOUND
      2. if key is larger than RANGE
         1. decrement key by range
   2. Declare new instance character array called stringArr with encryptedText length
      1. Begin for loop
      2. Declare integer variable i initialized to 0, if i is less than the length of encryptedText, increment i
         1. Initialize stringArr at index i to encryptedText at character index i minus key (type convert to char)
         2. if stringArr at index i is greater than UPPER\_BOUND
            1. decrement stringArr at index i by RANGE
         3. if stringArr at index i is less than LOWER\_BOUND
            1. increment stringArr at index i by RANGE
   3. Declare string instance decryptString and initialize with stringArr that is concatenated
   4. Return decryptString
5. Declare string method for decryptBellasco that accepts string encryptedText and string bellasoStr
   1. while plainText length is larger than bellasoStr length
      1. increment bellasoStr by itself
   2. declare array of type character
      1. encryptedTextArr initialize to encyptedText string converted to characters
      2. bellasoStrArr initialize to bellasoStr string converted to characters
   3. Begin for loop
      1. Declare integer variable i initialized to 0, if i is less than the length of encryptedText, increment i
      2. initialize encryptedTextArr at index i to char type encryptedTextArr character at index i plus RANGE minus bellasoStrArr at index i minus
      3. check range
         1. if encryptedTextArr at index i is larger than UPPER\_BOUND
            1. initialize encryptedTextArr at index i to itself minus RANGE
      4. check range
         1. if encryptedTextArr at index i is less than LOWER\_BOUND
            1. initialize encryptedTextArr at index i to itself plus RANGE
   4. declare string instance decryptString and initialize to concatenate encryptedTextArr
   5. return decryptString

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Input text | Input Key | Encrypted (method1) | Encrypted (method2) | Decrypt (method1) | Decrypt (method2) | Pass? |
| ODALIS IS AWESOME | Caesar: 3  Bellaso:  QUARANTIN | RGDOLV#LV  #DZHVRPH | YB^J!4R!%R,F%P[Y | ODALIS IS AWESOME | ODALIS IS AWESOME | Y |
| CAESAR | Caesar: 5  Bellaso:  FIVE | HFJXFW | IJ[XG[ | CAESAR | CAESAR | Y |
| EXAMPLE | Caesar: 3  Bellaso: CSMC | H[DPSOH | H%TPSYX | EXAMPLE | EXAMPLE | Y |
| HOW MUCH WOOD CAN A WOODCHUCK CHUCK IF A WOOD CHUCK COULD CHUCK | Caesar: 15  Bellaso: HELLO | W^&/\$RW/&^^S/  RP]/P/&^^SRW$  RZ/RW$RZ/XU/  P/&^^S/RW$RZ/R  ^$[S/RW$RZ | PT#,\]HT,&WTP,  RIS,M/\_T[PRPZOW  /KM!OZ(NR,P  (\[[S(HT!RS%O  [$TI,OW]HW | HOW MUCH WOOD CAN A WOODCHUCK CHUCK IF A WOOD CHUCK COULD CHUCK | HOW MUCH WOOD CAN A WOODCHUCK CHUCK IF A WOOD CHUCK COULD CHUCK | Y |
| FA{{IL | Caesar: 2  Bellaso: TWO | HC==KN | ZXJO [ | FA;;IL | FA;;IL | N |