The structure of the encrypter and the decrypter

KeyGenerator:

Creates a public and private key pair

Writes them to disk

Encrypter:

Reads a message from a file

Reads a public key from a file

Encrypt the message, given a public key

Converts the message to a BigInteger array

Encrypts the BigInteger array using the public key

Converts the BigInteger array to a string

Writes the encrypted version to a file

Decrypter:

Reads a cipher text from a file

Reads a private key from a file

Decrypt the message, given a private key

Converts the message to a BigInteger array

Decrypts the BigInteger array using the private key

Translates the array to a string of Unicode characters

Writes the decrypted string to a file

Encrypter:

Given a public key, converts a BigInteger array to an encrypted BigInteger array.

Decrypter:

Given a private key, converts a cipher BigInteger array to a decrypted BigInteger array.

StringManipulation:

Convert a string to BigInteger array using Unicode values

Convert a BigInteger array into a string using Unicode values

Concatenate a BigInteger array into a string

Convert a ciphertext string into a BigInteger array

How the private key is found

How e is found

How d is found

How the prime numbers are found

Process:

Group discussion around how to understand the algorithm

Test driven development – produced single class

Extensive refactoring into classes

First had a Message value object being passed around

Moved to an object for each stage of the process

Initial value object inside the KeyGenerator to pass values from function to function