# Introduction

This is a report of my assignment from Infor “CipherMachine”.

CipherMachine is a Console application written in Java using Eclipse. I was tasked to produce a software that could meet the following requirements below while following professional coding standards set by Infor. The finished solution only contains the backend.

# Scenario

Please provide an implementation of CipherMachine.java according to the description below. Where requirements may be vague - please provide a note about the issue and decisions taken.

Solutions should make use of standard platform features - no 3rd party libraries.

---- PART 1 ----

Build a simple substitution 'cipher machine'.

This algorithm will take a String message and 'encode' it such that it is no longer human readable. The approach will be to move the standard alphabet by a 'shift' value, such that a shift of 3 would mean A becomes D, D becomes G. The alphabetic shift should rotate the overhanging values to the beginning, such that Z becomes C. Casing should be preserved, but the character shift is the same, e.g. Z becomes C, z becomes c.

---- PART 1 ----

---- PART 2 ----

Upgrade the 'cipher machine' to include the following fixed substitutions:

: should output !

/ should output -

? should output +

# should output ,

. should output ]

Whitespace characters should be preserved in the output

Any unrecognized characters in the input message should output \_

---- PART 2 ---

---- PART 3 ---

Write the counterpart method 'decode' which takes an encoded String message, and a 'shift' value, and then returns the original human readable message.

e.g. KhoorZruog should output an encoded message of: HelloWorld

Any encoded \_ characters should not be not modified.

Any unrecognized characters should be output as \*

---- PART 3 ---

---- PART 4 ---

Decode the following message

Ugfyjslmdslagfk ]]] Lwdd fgtgvq lzw hskkogjv ak! ,Uzwvvsj,

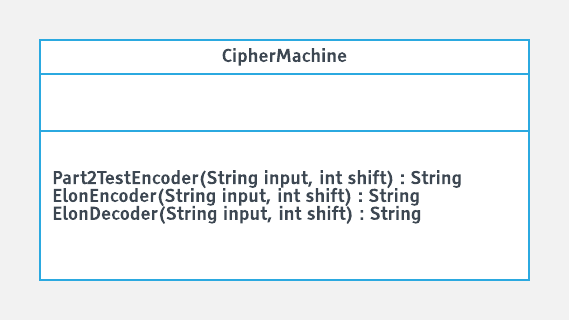
---- PART 4 ---

# Scope

## Non-Functional Requirements

## Must be a Console application written in Java

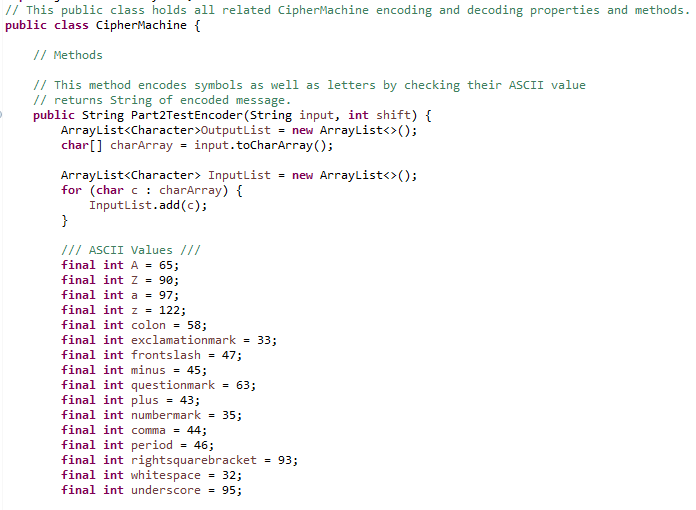
# Class Diagram



# Classes

## CipherMachine.java

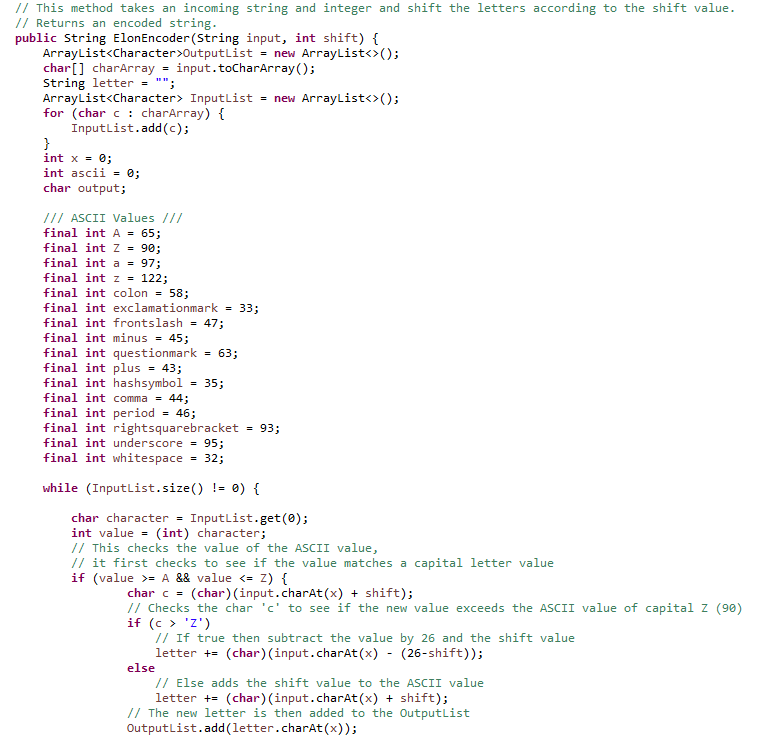
This class holds 3 different methods, however two of them are very similar. ElonEncoder and ElonDecoder are similar in that they either decode or encode a string. The third method is used to encode and decode symbol inputs as they act different.



# Methods

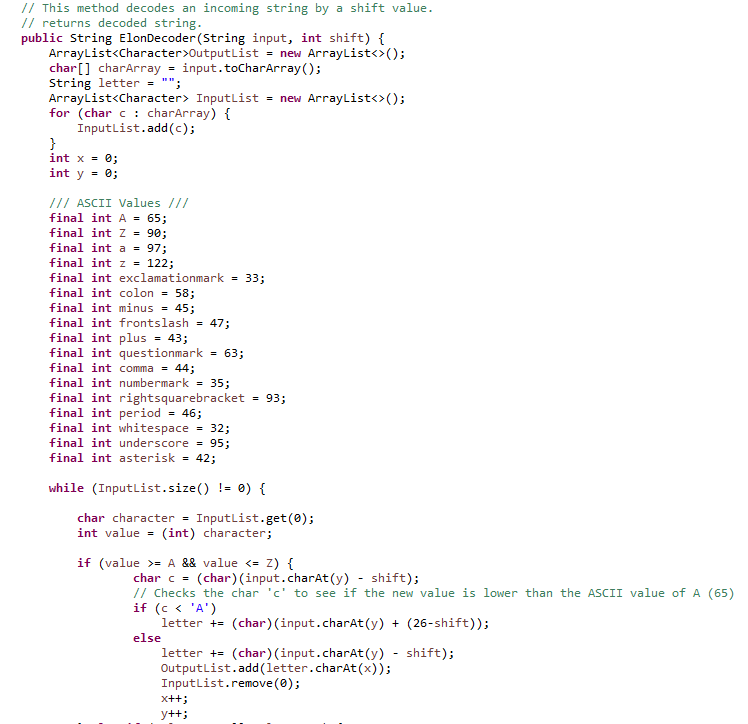
## ElonEncoder

The 'ElonEncoder' method takes a string and encodes it. It does this by moving the ASCII value of the letters by a shift value. It also changes certain symbols to other symbols as asked from the assignment, it does this by checking for certain ASCII values.



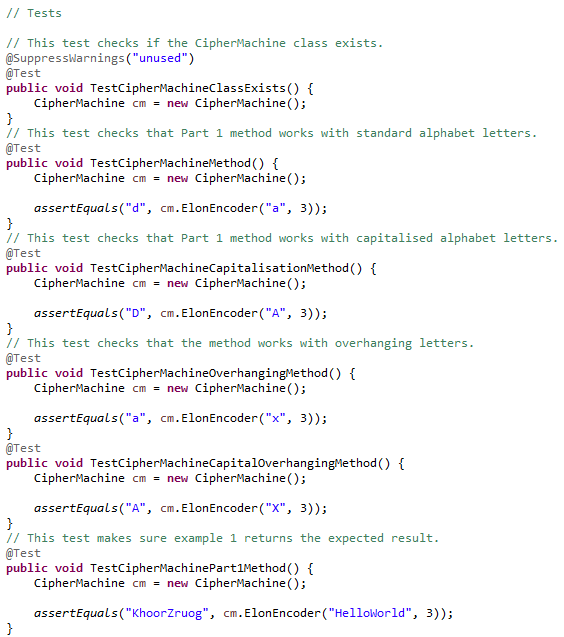
## ElonDecoder

Like the encoder method, this one works almost in a reverse way by decoding a string value. It does this by subtracting the ASCII values by the shift value. The symbols are also reserved too.



# Testing

This class is used to hold all the unit testing for the application. Some tests are simple in that they check if class exists or properties are functioning correctly, while others check if the correct encoded/decoded message is being returned or overhanging values are being returned as English characters. All tests passed successfully.



# Review

## Development

## QA Testing