UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FINAL PROJECT

Object-Oriented Programming

TEAM 5

Alberto Iván Del Valle Ramos
1545911

Jesús Tadeo Fernández Yépez
1580993

Marcela Estefanía Guel Mendoza
1589640

Saúl Iván Loredo Alanís
1639997

David Alberto Ruíz Castillo
1560718

Christopher Argenis Saucedo Morales
1616588

We dedicate this work to all important people to us. Thanks for all.

Team five

Contenido

Introduction

We arrive to last phase of object-oriented programming class. We'll show an application to write and encrypt a secret letter. In the same way save the letter on hard disk. Only the same application can decode and show the letter content.

For extra points, we used a graphic user interface (GUI), making the application more attractive and easy use.

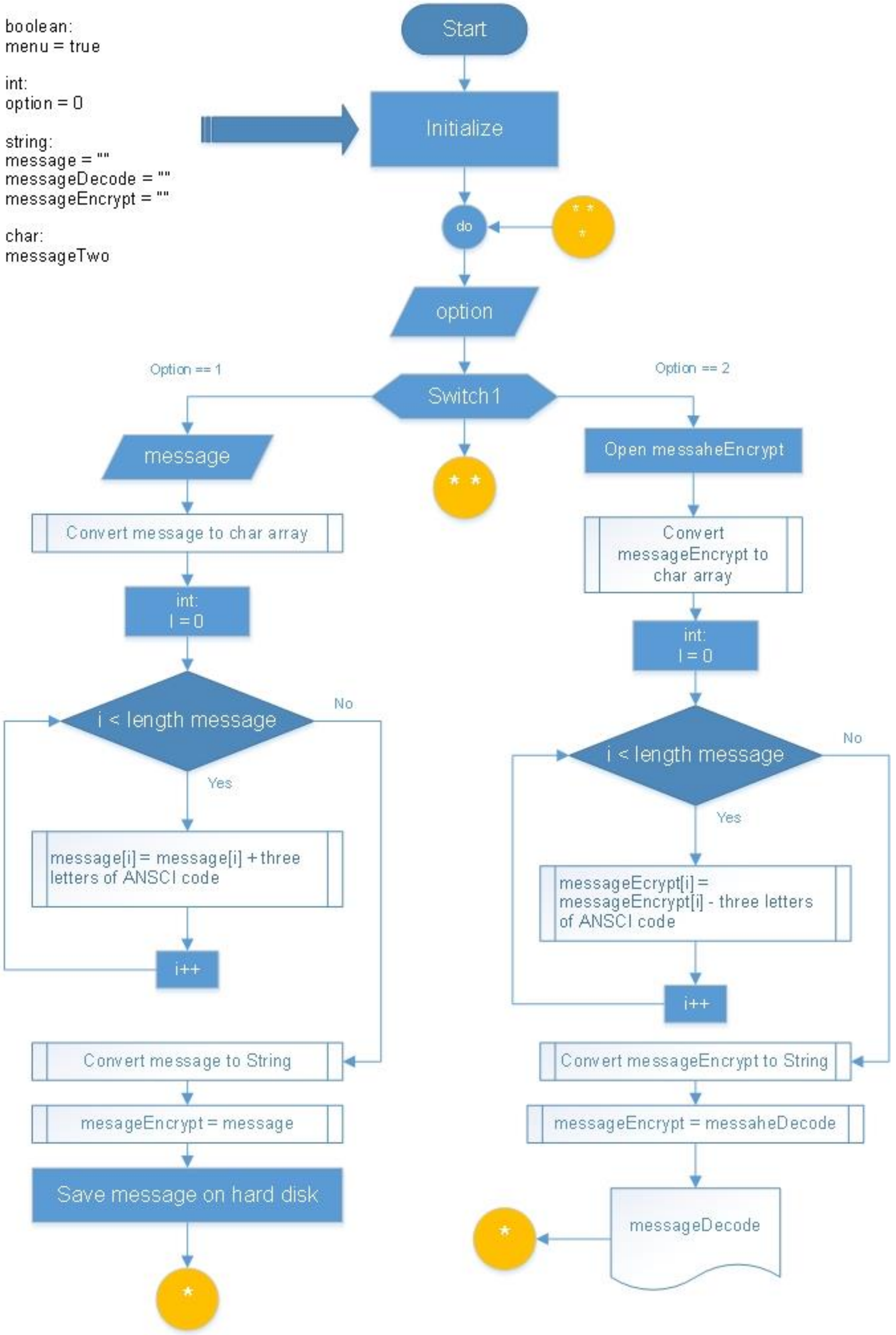
We will also describe step by step how we did.

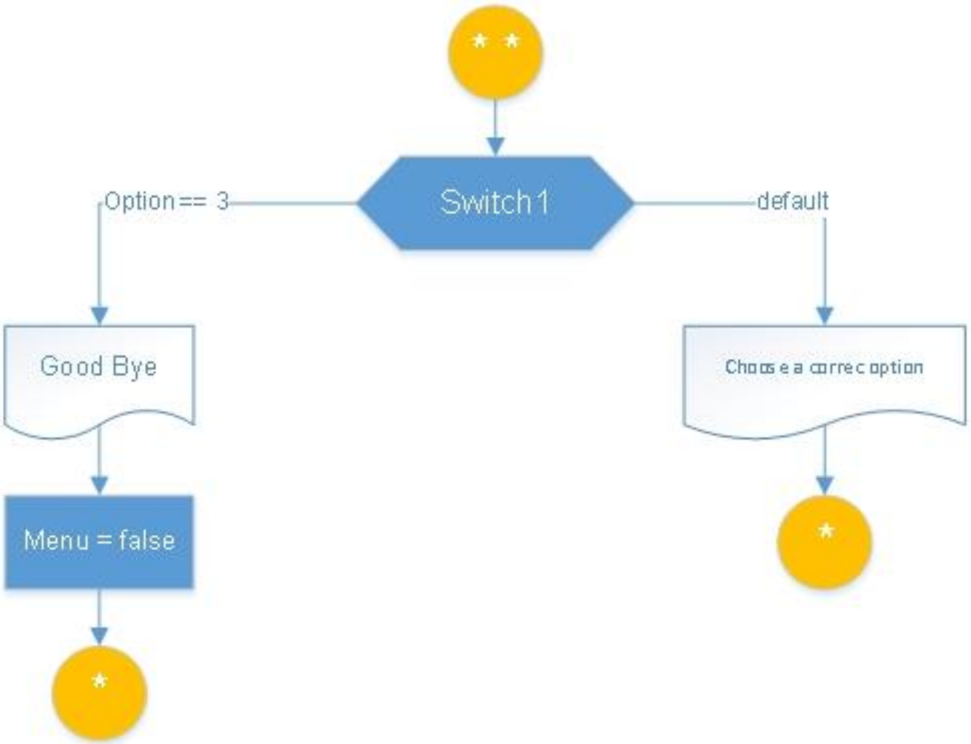
DIAGRAMS

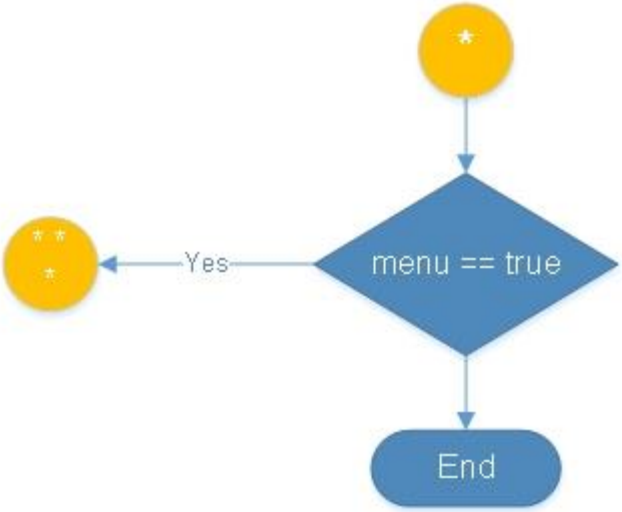
Algorithm

```
1.-  Start
2.-  Variables declaration
        1.- menu = true: boolean
        2.- option = 0: int
        3.- message = "", messageDecode = "", messageEncrypt = "": String
        4.- messageTwo: char
3.-  do
        1.- print ("Please, choose an option")
        2.- print ("1) Write message")
        3.- print ("2) Read message")
        4.- print ("3) Exit")
        5.- read ( option )
                1.- Switch( option )
                        1.- case 1:
                                1.- read( message )
                                2.- i = 0: int
                                3.- converted message to char array
                                4.- for ( i < length of message; i ++ )
                                        1.- messageEncrypt[i] = message[i] + three letters of ANSCI code
                                5.- End for cicle
                                6.- converted messageEncrypt to String
                                7.- save messageEncrypt
                                8.- break
                        2.- case 2:
                                1.- open the messageEncrypt
                                2.- Convert messageEncrypt to char array
                                3.- i = 0: int
                                4.- for( i < length messageEncrypt; i ++ )
                                        1.- messageDecode[i] = messageEncrypt[i] - three letters of ANSCI code
                                5.- End for cicle
                                6.- converted messageDecode  to String
                                7.- print ( messageDescode )
                                8.- break
                        3.- case 3:
                                1.- menu = false
                                2.- break
                        4.- default:
                                1.- print ("Please, select a correct option")
                                2.- break
                2.- End switch
4.-  while( menu == true  )
5.-  print("good bye")
6.- End
```

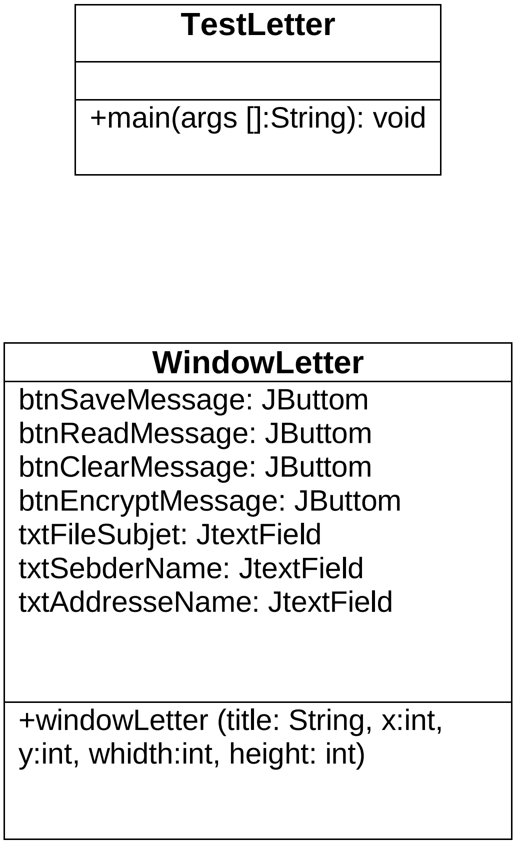
Flowchart


boolean:
menu = true
int:
option = 0
string:
message = ""
messageDecode = ""
messageEncrypt = ""
char:
messageTwo
Start
Initialize
do
* * *
option
Option == 1
Switch 1
Option == 2
message
* *
Open messaheEncrypt
Convert message to char array
Convert messageEncrypt to char array
int:
I = 0
int:
I = 0
i < length message
No
Yes
message[i] = message[i] + three letters of ANSCI code
i++
i < length message
No
Yes
messageEcrypt[i] = messageEncrypt[i] - three letters of ANSCI code
i++
Convert message to String
mesageEncrypt = message
Save message on hard disk
*
Convert messageEncrypt to String
messageEncrypt = messaheDecode
messageDecode
*


* *
Switch1
Option== 3
default
Good Bye
Choose a correc option
Menu = false
*
*


*
menu == true
Yes
* *
*
End

UML



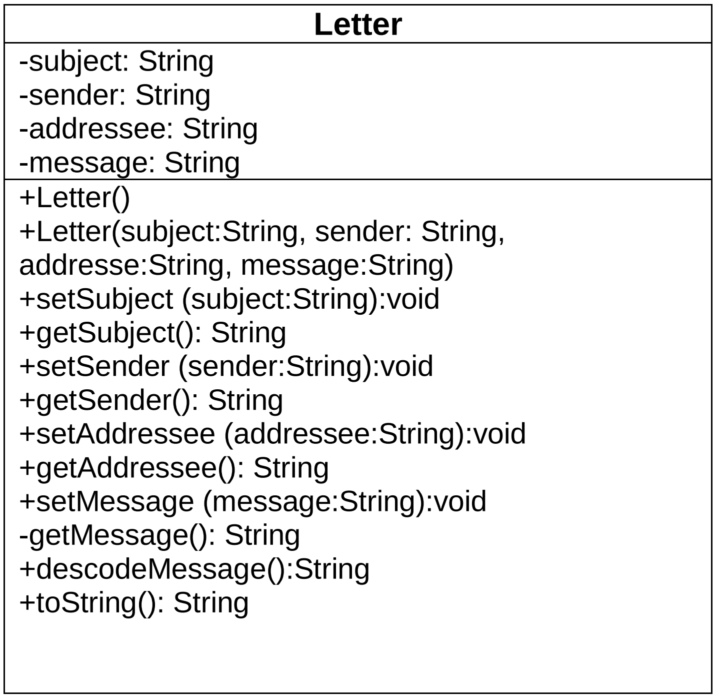
TestLetter

+main(args []:String): void

WindowLetter

btnSaveMessage: JButtom
btnReadMessage: JButtom
btnClearMessage: JButtom
btnEncryptMessage: JButtom
txtFileSubjet: JtextField
txtSebderName: JtextField
txtAddresseName: JtextField

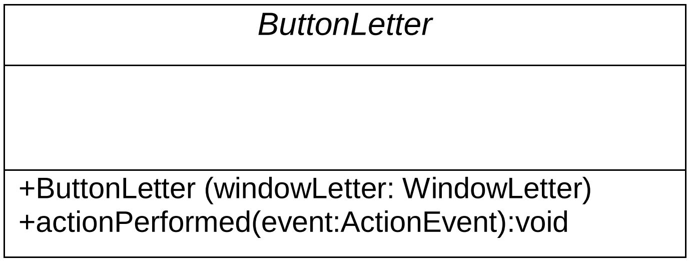
+windowLetter (title: String, x:int, y:int, whidth:int, height: int)



Letter

-subject: String
-sender: String
-addressee: String
-message: String

+Letter()
+Letter(subject:String, sender: String, addresse:String, message:String)
+setSubject (subject:String):void
+getSubject(): String
+setSender (sender:String):void
+getSender(): String
+setAddressee (addressee:String):void
+getAddressee(): String
+setMessage (message:String):void
-getMessage(): String
+descodeMessage():String
+toString(): String



ButtonLetter

+ButtonLetter (windowLetter: WindowLetter)
+actionPerformed(event:ActionEvent):void

Specifications

Class	Description
TestLetter	This class has the main method to run the program.
WindowLetter	This class extends JFrame class. Here is where develops the graphic interface (Buttons, JLabels and JTextArea).
ButtonLetter	This class has all buttons functions.
Letter	Inside this class are the set and get methods, and the toString method. Also are private variables.

Variables	Description
subject	Is inside the Letter class. Is private and complements the methods getSubject and setSubject. This variable permit that the letter has a Subject.
sender	Is inside the Letter class. Is private and complements the methods getSender and setSender. This variable permit that the letter has a Sender.
addressee	Is inside the Letter class. Is private and complements the methods getAddressee and setAddressee. This variable permit that the letter has an Addressee.
message	Is inside the Letter class. Is private and complements the methods getMessage and setMessage. This variable permit that write the secret message.

Methods	*Description*
main	Is in the TestLetter class. Permit the execution of program.
getSubject and setSubject	Everybody can instantiate the private variable subject if use this methods. Is inside the Letter class.
getSender and setSender	Everybody can instantiate the private variable sender if use this methods. Is inside the Letter class.
getAddressee and setAddresse	Everybody can instantiate the private variable addressee if use this methods. Is inside the Letter class.
getMessage and setMessage	Everybody can instantiate the private variable if use this methods.
decodeMessage	This method permit decode the secret message. Is inside the Letter class.
toString	This method describe the Letter class. Is inside the Letter class.

Conclusion

We enjoyed do this work, and we want the user also enjoy the application. All important documentation is here. We check the application run correctly.

Maybe isn't a perfect application, but is stable and has all the necessary for encrypt, save, write and read messages.

Bibliografía

Deitel, D. y. (s.f.). *Cómo programar en Java* (Novena edición ed.). Pearson.

Deitel, D. y. (s.f.). *Cómo programar en Java* (Septima edicion ed.). Pearson.

Oracle. (n.d.). *Oracle*. Retrieved from https://docs.oracle.com/javase/7/docs/api/

Parr, D. B. (s.f.). *JAVA para estudiantes* (Sexta ed.). Pearson.

Sierra, A. J. (s.f.). *Certificado en Java 2 Curso práctico* (2 edición ed.). Alfaomega.