CALCULATOR: The Champernowne Constant (C10)

Nellybett Irahola
ID #40079991
Repository:https://github.com/NellybettIrahola/SOEN6481-Calculator

Concordia University— August 02, 2019

List of Figures

List of Tables

Contents

Chapter 1

Introduction

This project is based in the development of a calculator that computes the value of the Champernowne Constant (C10) and perform several operations that reflect its different uses.

The purpose of the project is to engage with all the different concepts involved in the Requirement Specification Process. The calculator follows the user interests and expected functionalities.

In this project the first problem will present the user stories. The second problem will present a backwards traceability matrix and the last one will show the details of the implementation of the calculator.

Chapter 2

Problems

2.1 PROBLEM 6: User Stories

2.1.1 Description

The user stories' information will be presented in two sections. The first one with a table containing the id, description, priority, frequency of use, category and estimation. The estimation point of reference is one day which is represented by 8. The scale is defined using the fibonacci numbers.

The range for the priority and frequency is like the one provided by the CalCentral project [1]. It uses three levels Low, Medium and High.

The second section will present the constraints, acceptance test and acceptance criteria for the user stories. The constraints are presented by the relation with a non-functional requirement such as usability, flexibility, efficiency, security and others.

2.1.2 User Stories Characteristics

ID	User Story Description	Category	Priority	Frequency	Estimation (Story points)
EN-US-1	A student can use the calculator to show the decimal expansion of the Champernowne Constant in different bases, so he can get a better understanding of its behaviour for their research.	Research & Learning	High	High	5
EN-US-2	A Number's Theory specialist can use the calculator to verify the presence of a numeric pattern in the Champernowne Constant, so they can confirm the characteristics of normal numbers and use it for their research.	Research	Medium	Low	3
EN-US-3	A student can use the calculator to encrypt messages using a substitution cipher based on the number, so they can use it for their projects.	Learning & Security	High	High	5
EN-US-4	A student can use the calculator to decrypt messages using a substitution cipher based on the number and a key, so they can use it for their projects.	Learning & Security	High	High	5
EN-US-5	A student can use the calculator to encrypt messages using a one-time pad cipher based on the number, so they can use it for their projects.	Learning & Security	High	High	8
EN-US-6	A student can use the calculator to decrypt messages using a one-time pad cipher based on the number, so they can use it for their projects.	Learning & Security	High	High	8
EN-US-7	A student can use the calculator to perform elementary arithmetic operations that involve the number, so he can use it for specific calculations in his work.	Learning & Research	Medium	High	8
EN-US-8	A Number's Theory specialist can use the calculator to show the continued fraction expansion of the Champernowne Constant, so they evaluate some characteristics of the number and use it for their research.	Learning & Research	Low	Low	2
EN-US-9	A software engineer can use the calculator to show random numbers generated base on the Champernowne Constant, so they can use it for some interface features in their systems.	Learning & Work	Low	Low	3
EN-US-10	A student can use the calculator to show a graph of the number, so they can study their behaviour for their research.	Learning & Research	Medium	Low	5

Table 2.1: User Stories. Personal Creation.

2.1.3 Global Constraints

Usability: The program doesn't provide the option of increasing the size of the window.
Flexibility: the system is designed to work only in the English language.

2.1.4 User Stories Constraints and Acceptance Criteria

The elements are identified in a sequential order, and the corresponding acceptance test is associated with the constraint as explained in the classes material. Additional acceptance criteria are provided expressed in a pass and fail format.

1. **EN-US-1** : A student can use the calculator to show the decimal expansion of the Champernowne Constant in different bases, so he can get a better understanding of its behaviour for their research.

   Constraints

   - C1(Usability): It is important that the system provide a default value for the base of the number.
   - C2(Usability):The system should also provide a maximum number of decimals since the device process resources are limited (Champernowne Constant is an infinite number) and the screen size varies per device.
   - C3(Flexibility): the system should allow to select multiple bases for the number.
   - C4(Efficiency): the system should show the number in maximum 2 second after the requirement was made, since response time is really important for the users.

   Acceptance Criteria

   - C1-T1: if the user doesn't select the base, he can use the predefined value. To pass the system provides a predefined value.
   - C2-T1: if the the user inputs decimal more than 700 the system indicates to the user when he inputs more than the maximum number of decimals. Input:800, Expected result: error message.
   - C3-T1: the user can select multiple bases for the number. To pass the system should provide multiple bases (more than one).
   - C4-T1: the user receives the result in less than 2 seconds.
   - Cr1: if the user selects the Champernowne Constant with a number of decimals lower than 700 the system should show the Champernowne Constant with the specified number of decimals. If the input is 699 the system should show the number.
   - Cr2: if the user inputs a number of decimals that is not integer, the system should show an error.

2. **EN-US-2**: A Number's Theory specialist can use the calculator to verify the presence of a numeric pattern in the Champernowne Constant, so they can confirm the characteristics of normal numbers and use it for their research.

   Constraints

   - C5(Efficiency): the system should find the pattern in less than 4 seconds.
   - C6(Flexibility): the system should allow to find the pattern in multiple bases of the Champernowne Constant.
   - C7(Usability): the pattern must contain only numeric values, the system must give an error message in other case.

   Acceptance Criteria

   - C5-T1: if the input is correct the user receive the result in less than 4 seconds.
   - C6-T1: the user search the number with base 2 and introduces a binary pattern the system shows the position of the pattern if it is in the number.
   - C7-T1: the user introduces letters and gets an error.
   - Cr4: the system should return the position of the pattern if it exist in the decimals of the Champernowne Constant obtained by the user (base 10).
   - Cr5: if the pattern is not found the system should provide a message.

3. **EN-US-3**:A student can use the calculator to encrypt messages using a substitution cipher based on the number, so they can use it for their projects.

Constraints

- C8(Flexibility): not only the letters should be encrypted by the algorithm, the system should also provide encryption for common symbols.
- C9(Confidentiality): only the user that encrypt the message should be provided with the key to decrypt it.

Acceptance Criteria

- C8-T1: the user introduces a symbol (?,!,etc.) and it is processed following the algorithm.
- C9-T1: the key is provided as part of the message only to the user.
- Cr6: as a result the message provided to the user should be encrypted following the substitution cipher algorithm.
- Cr7: the algorithm should not change a not-supported symbol if they appear in the message. To pass the user provides a not-supported symbol and the system doesn't modifies it

4. **EN-US-4**:A student can use the calculator to decrypt messages using a substitution cipher based on the number and a key, so they can use it for their projects.

   Constraints

   - C10(Flexibility): not only the letters should be decrypted by the algorithm, the system should also provide decryption for common symbols.
   - C11(Integrity): the decrypted message should be the same than the original message provided by the user before encryption.
   - C12(Confidentiality): the message should only be decrypted if a valid key is provided.

   Acceptance Criteria

   - C10-T1: the message to decrypt has symbols and they are treat as any other element.
   - C11-T1: the user encrypts a message and then decrypts it with a valid key obtaining the original message.
   - C12-T1: the decrypted message will be totally different if a valid key is not provided.
   - Cr8: provided the right parameters the decrypted messages should be the same than the message before encryption.
   - Cr9: if the parameters are not correct, the system should show an error message.
   - Cr10: the algorithm should not change not-supported symbols if they appear in the message.

5. **EN-US-5**:A student can use the calculator to encrypt messages using a one-time pad cipher based on the number, so they can use it for their projects.

   Constraints

   - C13(Flexibility): not only the letters should be encrypted by the algorithm, the system should encrypt the common symbols.
   - C14(Confidentiality): the procedure to encrypt the message must be confidential.

   Acceptance Criteria

   - C13-T1: the user introduces a common symbol and it is encrypted as any other character.
   - C14-T1: the user introduces a message and the key and encryption is only provided to him.
   - Cr11: as a result the message provided by the user should be encrypted following the one-time pad cipher algorithm.

- Cr12: the algorithm should not change not-supported symbols if they appear in the message.

6. **EN-US-6**:A student can use the calculator to decrypt messages using a one-time pad cipher based on the number, so they can use it for their projects.

   Constraints

   - C15(Flexibility): not only the letters should be decrypted by the algorithm, the system should also take in consideration common symbols.
   - C16(Integrity): the decrypted message should be the same than the original message provided by the user before encryption.

   Acceptance Criteria

   - C15-T1: the user introduces a common symbol and it is decrypted as any other character.
   - C16-T1: the user provides a message for encryption and given a valid key and the encrypted message the system get the original message after the decryption process.
   - Cr13: provided a valid encrypted message and a valid key the decrypted messages should be the same than the message before encryption.
   - Cr14: the algorithm should not change not-supported symbols if they appear in the message.

7. **EN-US-7**: A student can use the calculator to perform elementary arithmetic operations that involve the number, so he can use it for specific calculations in his work.

   Constraints

   - C16(Usability): the number should be represented by its symbol in the interface provided to the user.
   - C17(Usability): a result will only be shown if equals is press by the user.
   - C18(Flexibility): the user should be able to input multiple operations as part of a mathematical expression.
   - C19(Usability): no delete button is provided in case of a mistake the user will have to delete the complete operation.
   - C20(Usability): the calculator only provide an specific number of decimals.
   - C21(Usability): the negative symbol for a number should be differentiated from the negative sign of a subtraction.

   Acceptance Criteria

   - C16-T1: the symbol of the number is C10, it should be provided in the interface.
   - C17-T1: if the user doesn't press equal the result will not be displayed.
   - C18-T1: the expression is calculated only after pressing equal and it can be composed of several operations.
   - C19-T1: the expression is deleted when the user press "C" symbol.
   - C20-T1: the expression is calculated with 4 decimals per number in the expression and the result shows all the available decimals.
   - C21-T1: the negative of a number is represented by +/- symbol.
   - Cr15: given a valid mathematical expression the result must be the calculation of the mathematical expression.
   - Cr16: the system should provide an error message if the mathematical expression is not valid.

8. **EN-US-8**: A Number's Theory specialist can use the calculator to show the continued fraction expansion of the Champernowne Constant, so they evaluate some characteristics of the number and use it for their research.

   Constraints

   - C21(Reliability): the system must show all the available elements of the fraction expansion.
   - C22(Usability): the user should be able to see for which bases is available the fraction expansion of the number.
   - C23(Usability):The system will only show elements with less than 200 numbers.

   Acceptance Criteria

   - C21-T1: the result coincides with the mathematical calculation of the fraction expansion of the number.
   - C22-T1: the bases are shown in the interface and can be selected by the user.
   - C23-T1: the system shows only elements with less than 200 numbers after the user initiates the operation.
   - Cr17: the result must be the mathematical expansion in the specified base.

9. **EN-US-9**: A software engineer can use the calculator to show random numbers generated base on the Champernowne Constant, so they can use it for some interface features in their systems.

   Constraints

   - C24(Usability): The system should indicate what is the maximum value for the range.
   - C25(Flexibility): the system should provide ranges for positive and negative integers and indicate the type of format that is not supported by the system.

   Acceptance Criteria

   - C24-T1: the user inputs a range over the limit and the system shows an error message.
   - C25-T1: the user inputs a negative range and the system gives a response.
   - Cr18: the result must be a number in the range provided by the user.

10. **EN-US-10**: A student can use the calculator to show a graph of the number, so they can study their behaviour for their research.

    Constraints

    - C26(Flexibility): the user should be able to choose the type of graph and the base of the number.
    - C27(Usability): the user should be able to copy the graph generated by the system, and select the information to be displayed (axis, tittle, and others).

    Acceptance Criteria

    - C26-T1: the system shows the types of graph available in the interface.
    - C27-T1: the user can select the information to be displayed and copy the graph generated by the system.
    - Cr19: the result must be a graph of the number in the base specified by the user only showing the information required by the user.

2.2 PROBLEM 7: Traceability Matrix

2.2.1 Nomenclature

1. Use Cases
   - ID: EN-UC-0
     Name: Calculate Champernowne Constant
   - ID: EN-UC-1
     Name: Show Number
   - ID: EN-UC-2
     Name: Find Numeric Pattern
   - ID: EN-UC-3
     Name: Encrypt Message
   - ID: EN-UC-4
     Name: Decrypt Message
2. Interviews
   - ID: EN-IN-1
     Interviewee Name: Hershy Kisilevsky
   - ID: EN-IN-2
     Interviewee Name: Daniel Morales
3. Persona
   - ID: EN-PE-1
     Name: David Wilson
   - ID: EN-PE-2
     Name: James Brown
4. Articles
   - ID: EN-AR-1
     Name: Transcendental Numbers and Cryptography
     Link: http://www.m-hikari.com/ams/ams-2014/ams-173-176-2014/viswanathAMS173-176-2014.pdf
   - ID: EN-AR-2
     Name: A CIPHER BASED ON THE RANDOM SEQUENCE OF DIGITS IN IRRATIONAL NUMBERS
     Link: http://www.iacis.org/iis/2016/1_iis_2016_14-25.pdf
   - ID: EN-AR-3
     Name: Champernowne Constant
     Link: http://mathworld.wolfram.com/ChampernowneConstant.html
5. Domain Model
   - Entities: Find Numeric Pattern(FNP), Message Encryption(ME), Message Decryption(MD), Calculator(C), Mathematical Expression (ME) and Number(N).

		Use Cases	User Stories	Interview	Persona	Reference Articles	Domain Model
User Stories	EN-US-1	EN-UC-0, EN-UC-1		EN-IN-1.Question4	EN-PE-2		N
	EN-US-2	EN-UC-0, EN-UC-2		EN-IN-1.Question5, EN-IN-2.Question10	EN-PE-1, EN-PE-2		FNP
	EN-US-3	EN-UC-0, EN-UC-3		EN-IN-2.Question6, EN-IN-2.Question8	EN-PE-1	EN-AR-1	ME
	EN-US-4	EN-UC-0, EN-UC-4		EN-IN-2.Question6, EN-IN-2.Question8	EN-PE-1	EN-AR-1	MD
	EN-US-5			EN-IN-2.Question6, EN-IN-2.Question8	EN-PE-1	EN-AR-2	ME
	EN-US-6			EN-IN-2.Question6, EN-IN-2.Question8	EN-PE-1	EN-AR-2	MD
	EN-US-7		EN-US-1	EN-IN-2.Question10, EN-IN-2.Question11	EN-PE-1		C, ME
	EN-US-8			EN-IN-1.Question4	EN-PE-2		N
	EN-US-9			EN-IN-1.Question8	EN-PE-2		
	EN-US-10					EN-AR-3	

Table 2.2: Backwards Traceability Matrix. Personal Creation.

REQUIREMENTS(Use Cases, Interviews, Personas, Articles, User Stories, Domain Model)

		EN-UC-0	EN-UC-1	EN-UC-2	EN-UC-3	EN-UC-4	EN-IN-1	EN-IN-2	EN-PE-1	EN-PE-2	EN-AR-1	EN-AR-2	EN-AR-3	EN-US-1	DM
USER STORIES	EN-US-1	x	x				x			x					x
	EN-US-2	x		x			x	x	x	x					x
	EN-US-3	x			x			x	x		x				x
	EN-US-4	x				x		x	x		x				x
	EN-US-5							x	x			x			x
	EN-US-6							x	x			x			x
	EN-US-7							x	x					x	x
	EN-US-8						x			x					x
	EN-US-9						x			x					
	EN-US-10												x		

Table 2.3: Extended Backwards Traceability Matrix. Personal Creation.

2.3 PROBLEM 8: Implementation of the Calculator

2.3.1 Conventions and Patterns

The application follows the Model-View-Controller design pattern. In this pattern the models manage the data of the application, the controllers manage the data flow to the controller and updates the view when neccessary and the view represents the visualization of the data. In the case of this application the views where developed using swing library, the controllers are represented by listeners and the model by two principal classes one Calculator.java that contains the calculator functions and ChampernowneModel.java that contains the functions related to the Champernowne Constant.

A modified version of Memento design pattern was used to retrieve previous results of the calculator basic operations. Additionally, the CamelCase Notation was used as naming convention. An organization of the directories and a class diagram is provided to show the interaction between models, views and controllers.

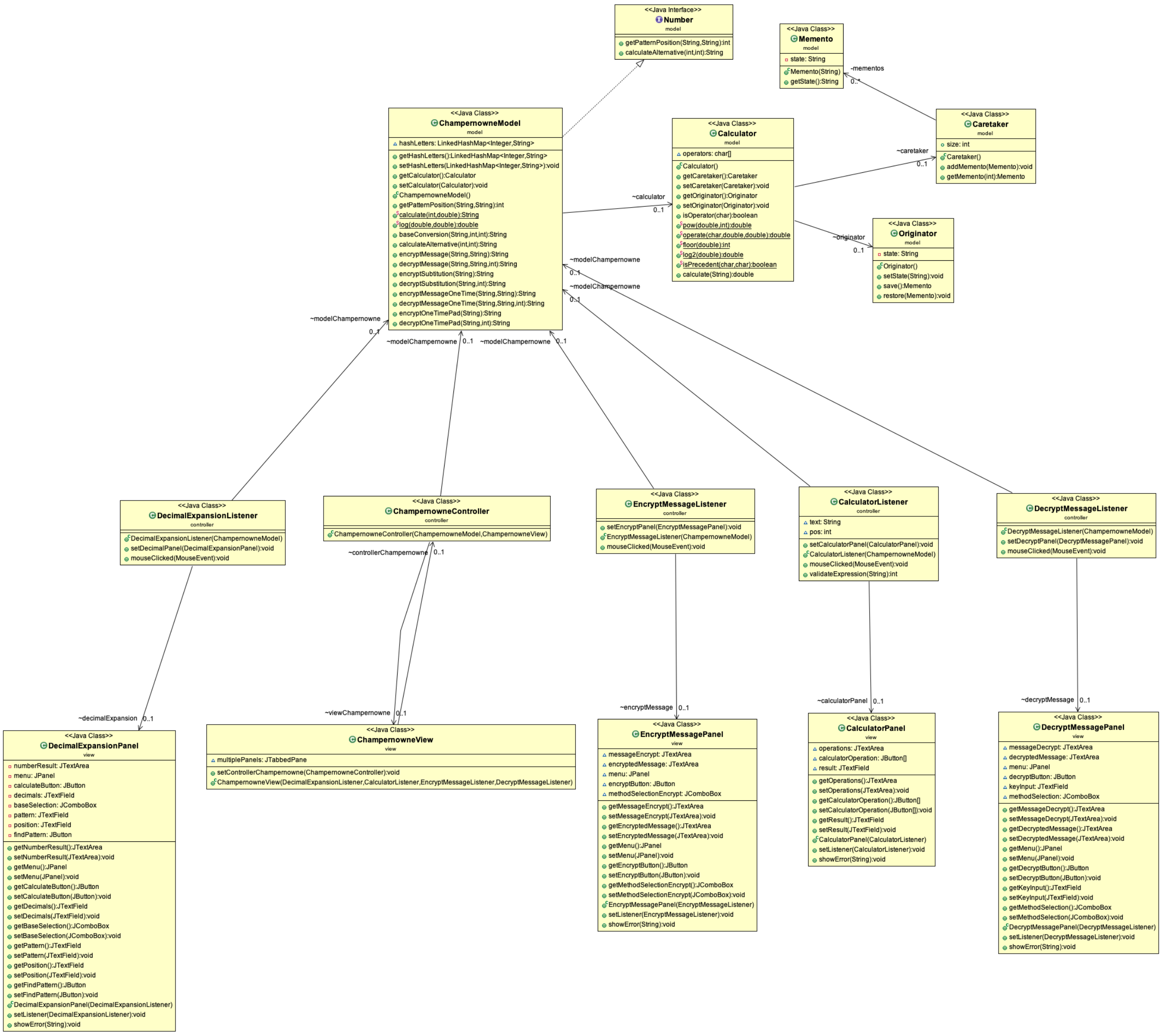
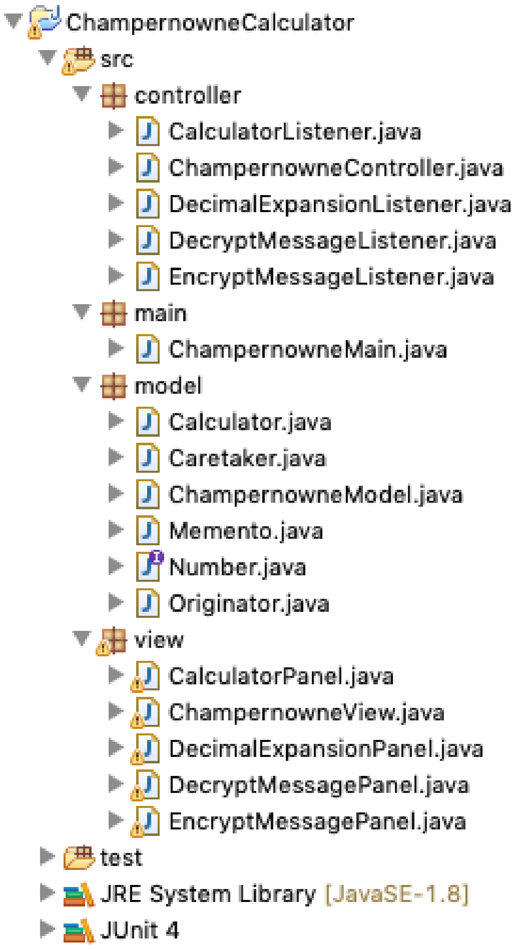

<<Java Interface>>
Number
model
getPatternPosition(String,String):int
calculateAlternative(int,int):String
<<Java Class>>
Memento
model
state: String
Memento(String)
getState():String
-mementos
0..*
<<Java Class>>
ChampernowneModel
model
hashLetters: LinkedHashMap<Integer,String>
getHashLetters():LinkedHashMap<Integer,String>
setHashLetters(LinkedHashMap<Integer,String>):void
getCalculator():Calculator
setCalculator(Calculator):void
ChampernowneModel()
getPatternPosition(String,String):int
calculate(int,double):String
log(double,double):double
baseConversion(String,int,int):String
calculateAlternative(int,int):String
encryptMessage(String,String):String
decryptMessage(String,String,int):String
encryptSubtitution(String):String
decryptSubstitution(String,int):String
encryptMessageOneTime(String,String):String
decryptMessageOneTime(String,String,int):String
encryptOneTimePad(String):String
decryptOneTimePad(String,int):String
<<Java Class>>
Calculator
model
operators: char[]
Calculator()
getCaretaker():Caretaker
setCaretaker(Caretaker):void
getOriginator():Originator
setOriginator(Originator):void
isOperator(char):boolean
pow(double,int):double
operate(char,double,double):double
floor(double):int
log2(double):double
isPrecedent(char,char):boolean
calculate(String):double
~calculator
0..1
<<Java Class>>
Caretaker
model
size: int
Caretaker()
addMemento(Memento):void
getMemento(int):Memento
~caretaker
0..1
<<Java Class>>
Originator
model
state: String
Originator()
setState(String):void
save():Memento
restore(Memento):void
~originator
0..1
~modelChampernowne
0..1
~modelChampernowne
0..1
~modelChampernowne
0..1
~modelChampernowne
0..1
~modelChampernowne
0..1
<<Java Class>>
DecimalExpansionListener
controller
DecimalExpansionListener(ChampernowneModel)
setDecimalPanel(DecimalExpansionPanel):void
mouseClicked(MouseEvent):void
<<Java Class>>
ChampernowneController
controller
ChampernowneController(ChampernowneModel,ChampernowneView)
~controllerChampernowne
0..1
<<Java Class>>
EncryptMessageListener
controller
setEncryptPanel(EncryptMessagePanel):void
EncryptMessageListener(ChampernowneModel)
mouseClicked(MouseEvent):void
<<Java Class>>
CalculatorListener
controller
text: String
pos: int
setCalculatorPanel(CalculatorPanel):void
CalculatorListener(ChampernowneModel)
mouseClicked(MouseEvent):void
validateExpression(String):int
<<Java Class>>
DecryptMessageListener
controller
DecryptMessageListener(ChampernowneModel)
setDecryptPanel(DecryptMessagePanel):void
mouseClicked(MouseEvent):void
~decimalExpansion
0..1
<<Java Class>>
DecimalExpansionPanel
view
numberResult: JTextArea
menu: JPanel
calculateButton: JButton
decimals: JTextField
baseSelection: JComboBox
pattern: JTextField
position: JTextField
findPattern: JButton
getNumberResult():JTextArea
setNumberResult(JTextArea):void
getMenu():JPanel
setMenu(JPanel):void
getCalculateButton():JButton
setCalculateButton(JButton):void
getDecimals():JTextField
setDecimals(JTextField):void
getBaseSelection():JComboBox
setBaseSelection(JComboBox):void
getPattern():JTextField
setPattern(JTextField):void
getPosition():JTextField
setPosition(JTextField):void
getFindPattern():JButton
setFindPattern(JButton):void
DecimalExpansionPanel(DecimalExpansionListener)
setListener(DecimalExpansionListener):void
showError(String):void
~viewChampernowne
0..1
<<Java Class>>
ChampernowneView
view
multiplePanels: JTabbedPane
setControllerChampernowne(ChampernowneController):void
ChampernowneView(DecimalExpansionListener,CalculatorListener,EncryptMessageListener,DecryptMessageListener)
~encryptMessage
0..1
<<Java Class>>
EncryptMessagePanel
view
messageEncrypt: JTextArea
encryptedMessage: JTextArea
menu: JPanel
encryptButton: JButton
methodSelectionEncrypt: JComboBox
getMessageEncrypt():JTextArea
setMessageEncrypt(JTextArea):void
getEncryptedMessage():JTextArea
setEncryptedMessage(JTextArea):void
getMenu():JPanel
setMenu(JPanel):void
getEncryptButton():JButton
setEncryptButton(JButton):void
getMethodSelectionEncrypt():JComboBox
setMethodSelectionEncrypt(JComboBox):void
EncryptMessagePanel(EncryptMessageListener)
setListener(EncryptMessageListener):void
showError(String):void
~calculatorPanel
0..1
<<Java Class>>
CalculatorPanel
view
operations: JTextArea
calculatorOperation: JButton[]
result: JTextField
getOperations():JTextArea
setOperations(JTextArea):void
getCalculatorOperation():JButton[]
setCalculatorOperation(JButton[]):void
getResult():JTextField
setResult(JTextField):void
CalculatorPanel(CalculatorListener)
setListener(CalculatorListener):void
showError(String):void
~decryptMessage
0..1
<<Java Class>>
DecryptMessagePanel
view
messageDecrypt: JTextArea
decryptedMessage: JTextArea
menu: JPanel
decryptButton: JButton
keyInput: JTextField
methodSelection: JComboBox
getMessageDecrypt():JTextArea
setMessageDecrypt(JTextArea):void
getDecryptedMessage():JTextArea
setDecryptedMessage(JTextArea):void
getMenu():JPanel
setMenu(JPanel):void
getDecryptButton():JButton
setDecryptButton(JButton):void
getKeyInput():JTextField
setKeyInput(JTextField):void
getMethodSelection():JComboBox
setMethodSelection(JComboBox):void
DecryptMessagePanel(DecryptMessageListener)
setListener(DecryptMessageListener):void
showError(String):void

Figure 2.1: Class Diagram.Personal Creation



```
ChampernowneCalculator
  src
    controller
      CalculatorListener.java
      ChampernowneController.java
      DecimalExpansionListener.java
      DecryptMessageListener.java
      EncryptMessageListener.java
    main
      ChampernowneMain.java
    model
      Calculator.java
      Caretaker.java
      ChampernowneModel.java
      Memento.java
      Number.java
      Originator.java
    view
      CalculatorPanel.java
      ChampernowneView.java
      DecimalExpansionPanel.java
      DecryptMessagePanel.java
      EncryptMessagePanel.java
  test
  JRE System Library [JavaSE-1.8]
  JUnit 4
```

Figure 2.2: Directory organization.Personal Creation

2.3.2 Implementation by View

The application is form by four panels: Calculator, Champernowne Constant, Encrypt Message and Decrypt Message. This section will describe the user stories implemented in each panel. All the use cases mentioned for the delivery 1 were implemented with their related user story. The use cases related to the basic calculator operation, operation with the number and number applications were covered.

Calculator

This view involves the implementation of EN-US-7 related to the execution of basic mathematical operations. The main functions for the implementation of this function are the validateExpression that is in the controller CalculatorListener.java and the calculate function in the model Calculator.java. The view is represented by the CalculatorPanel.java.

The Champernowne Constant is represented by the C10 symbol and the negative sign for number is represented by +/-.

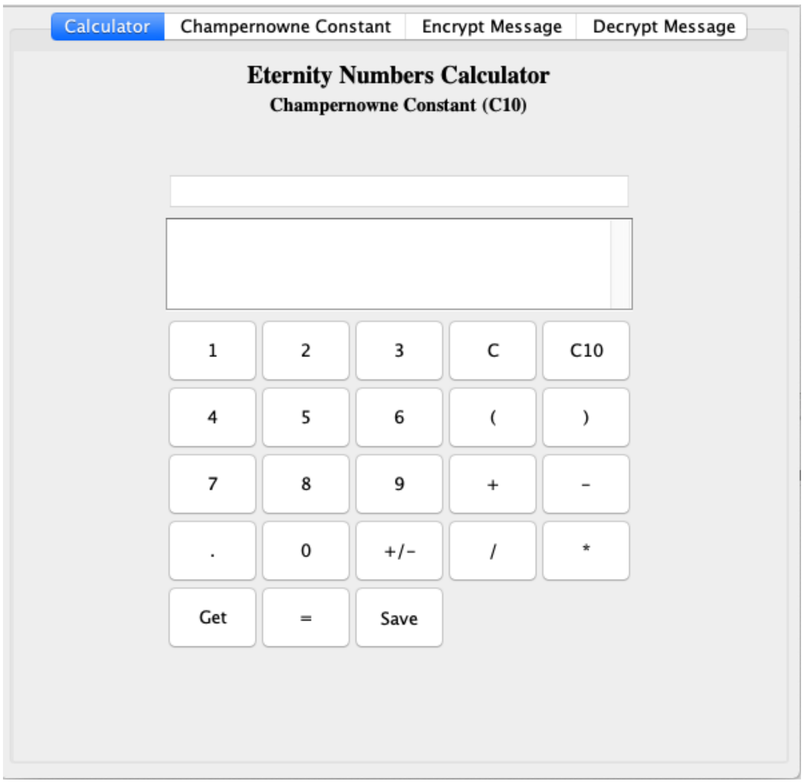
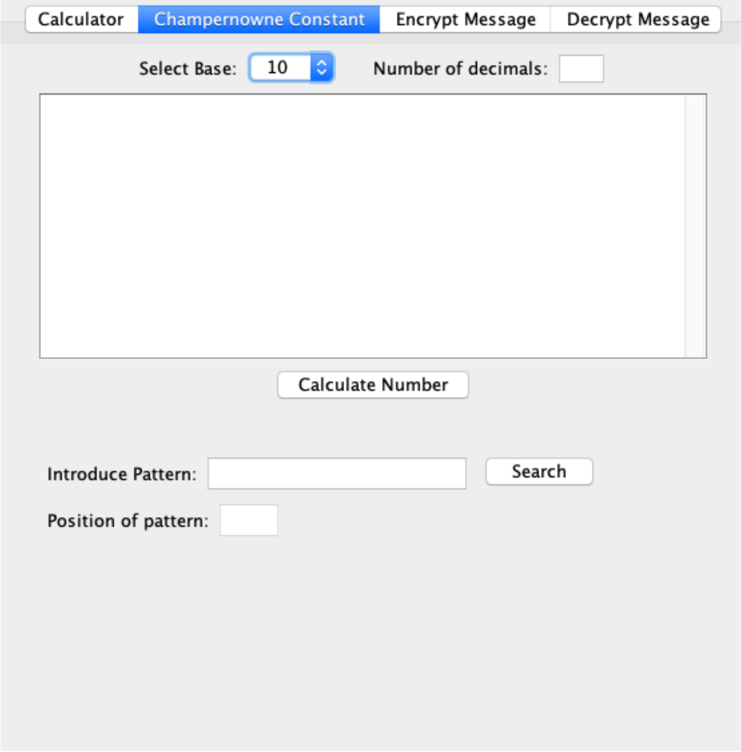


Figure 2.3: Calculator View.Personal Creation

Champernowne Constant View

This view involves the implementation of EN-US-1 and EN-US-2, since the Champernowne Constant is a built number it was for by the concatenation of strings which provided the possibility of presenting more decimals to the user for the identification of patterns in those decimals. The calculation of the number by formula is also provided for the EN-US-7.

The controller for this view is the DecimalExpansionListener.java, the view the DecimalExpansionPanel.java and the model the ChampernowneModel.java.



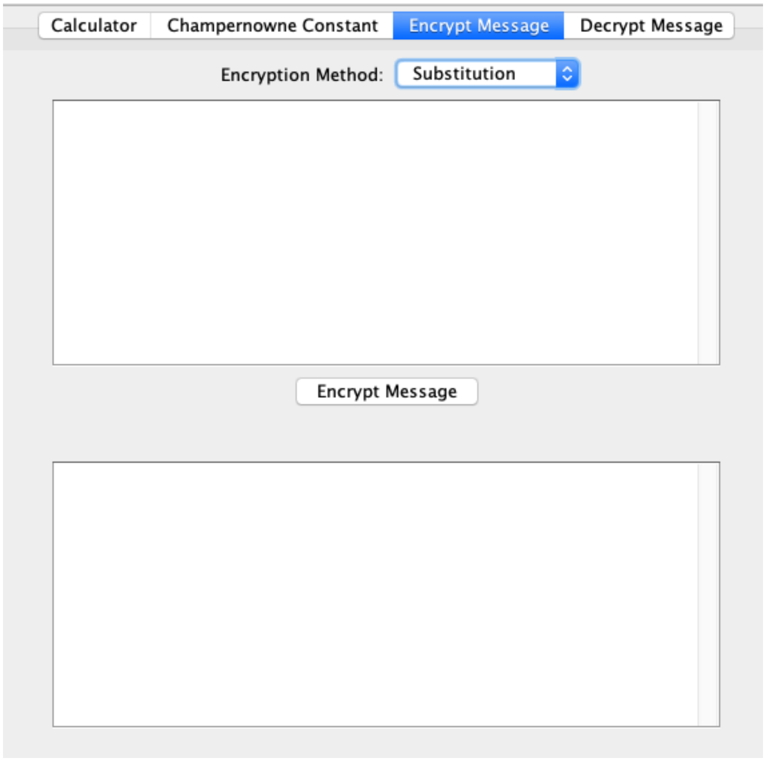
Select Base: 10

Figure 2.4: Champernowne Constant View.Personal Creation

Encrypt Message View

This view includes the implementation of the EN-US-3. The substitution cipher was creating by using a hash map with the most common symbols of the language, then selecting a position that represents a starting element in the Champernowne Constant and modifying the letter using the value in that position and the hash getting a new symbol as a result.

The view for this implementation is the EncryptMessagePanel.java, the controller the EncryptMessageListener.java and the model the ChampernowneModel.java.



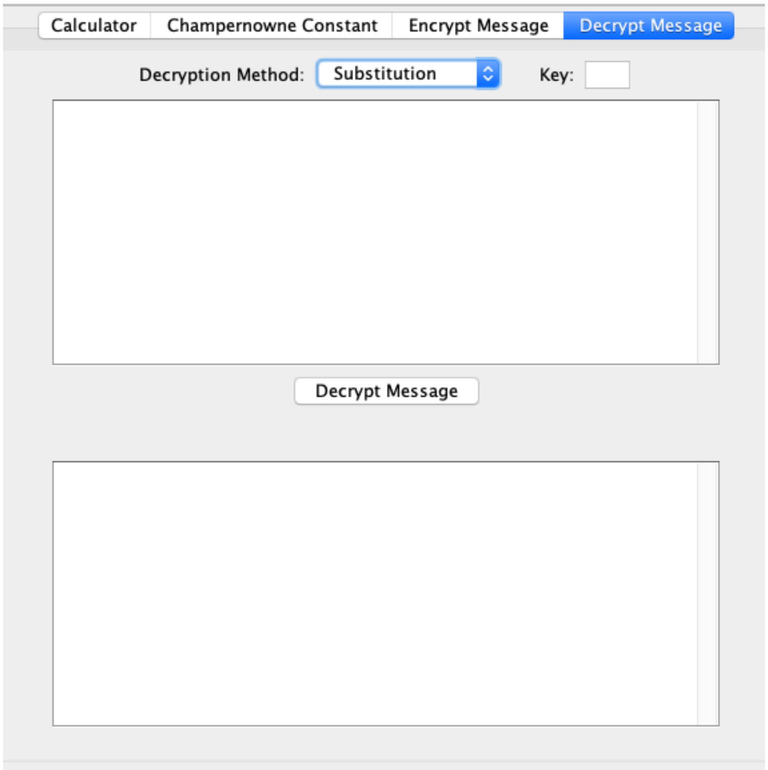
Encryption Method: Substitution

Figure 2.5: Encrypt Message View.Personal Creation

Decrypt Message View

This view includes the implementation of EN-US-4. The decryption for the substitution cipher was implementing by reverting the movement previously made according to a position in the Champernowne Constant (the key).

The view for this implementation is the DecryptMessagePanel.java, the controller the DecryptMessageListener.java and the model the ChampernowneModel.java.

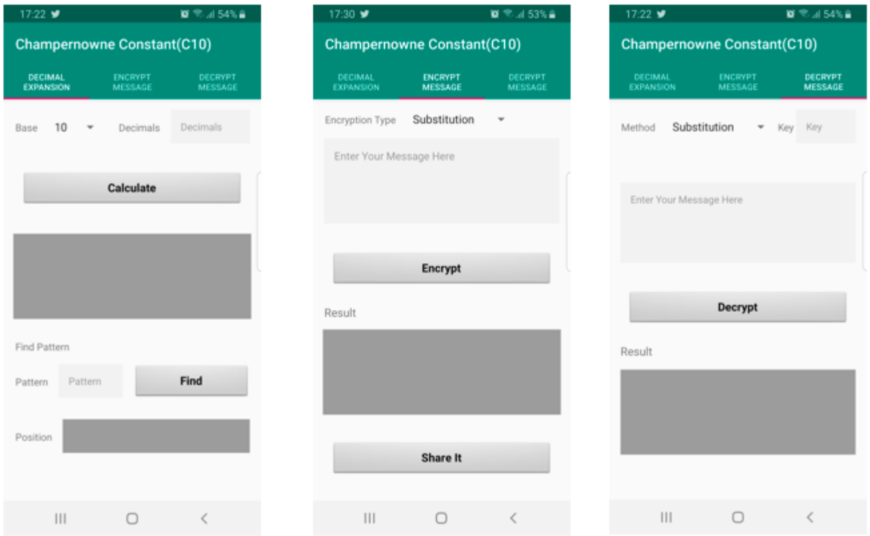


Decryption Method: Substitution

Figure 2.6: Decrypt Message View.Personal Creation

2.3.3 Additional Work

Some unit test case were developed for the most important functions in the model and the controllers using JUnit and the documentation using javadoc. Also, as requested for the interview a prototype of an android application for the calculator was developed. The code is provided in the repository.



Base 10

Encryption Type Substitution

Method Substitution

Figure 2.7: Mobile Application Prototype Views.Personal Creation

Chapter 3

Conclusion

The objectives in each problem were achieved, even though the Champernowne Constant is not known for a lot people. The problems allowed to gather different requirements for potential users, gain understanding about the applications and characteristics of the Champernowne Constant, and catch a glimpse of the features that the calculator should have.

It is important to mention that the operation of the calculator are specifically related with the characteristics of the number. The find pattern is possible since the Champernowne Constant is a normal number, and the cryptography features are based on the fact that it is a transcendental number.

This report shows the implementation of the functionalities of the system; it was developed as a first version, multiple modifications can be added in the future. The extensibility features include tackling the global constraints to make a more flexible user interface and provide support for multiple languages.

Additionally, it would be useful to increase the range of action of the encryption and decryption tool by adding other transcendental numbers which would increase the randomness and security of the application.

Bibliography

[1] P. Kantham. USER STORIES IN CONTEXT.(2019). Retrieve from:
`https://users.encs.concordia.ca/ kamthan/courses/soen-6481/user_stories_context.pdf`

[2] P. Kantham. USER STORIES IN CONTEXT.(2019). Retrieve from:
`https://users.encs.concordia.ca/ kamthan/courses/soen-6481/user_stories_context.pdf`

[3] P. Kantham. TRACEABILITY IN SOFTWARE REQUIREMENTS. (2019). Retrieve from:
`https://users.encs.concordia.ca/k̃amthan/courses/soen-6481/software_requirements_traceability.pdf`

[4] P. Kantham. 1INTRODUCTION TO SOFTWARE PRODUCT QUALITY. (2019). Retrieve from:
`https://users.encs.concordia.ca/k̃amthan/courses/soen-6481/software_product_quality_introduction.pdf`

[5] M.K. Viswanath. Transcendental Numbers and Cryptography. (2019). Retrieve from:
`http://www.m-hikari.com/ams/ams-2014/ams-173-176-2014/viswanathAMS173-176-2014.pdf`

[6] J. L. González-Santander. A CIPHER BASED ON THE RANDOM SEQUENCE OF DIGITS IN IRRATIONAL NUMBERS. (2019). Retrieve from:
`http://www.iacis.org/iis/2016/1_iis_2016_14-25.pdf`