

Đồ án lập trình mã hóa cổ điển

Môn: Mã hóa ứng dụng

Thầy: Đoàn Trình Dục

Bài báo cáo

Thành viên 1

Họ tên: Nguyễn Đình Phương

MSSV: DH52201277

Lớp: D22_TH14

Thành viên 2

Họ tên: Đặng Tiến Dũng

MSSV: DH52200530

Lớp: D22_TH15

subScreen

- Cấu trúc:

- Lớp subScreen là lớp cha chứa các thành phần giao diện chung, giúp các lớp con kế thừa và tái sử dụng các thành phần này. Mục tiêu chính là tạo giao diện cơ bản cho các màn hình mã hóa và giải mã trong ứng dụng.
- Các thành phần giao diện chính gồm: JLabel, JTextArea, JButton, JTextField cho việc nhập dữ liệu (plaintext, ciphertext, key) và điều khiển (mã hóa, giải mã, thoát).

- Tính năng:

• Giao diện nhập liệu:

- Người dùng nhập văn bản cần mã hóa/giải mã (plaintext/ciphertext) vào các JTextArea và nhập khóa (key) vào JTextField.
- JTextArea có thanh cuộn để hỗ trợ văn bản dài và được cấu hình để tự động xuống hàng khi vượt quá kích thước vùng hiển thị.

• Nút điều khiển:

- Nút "Encryption": Khi nhấn sẽ thực hiện chức năng mã hóa văn bản (được định nghĩa trong lớp con).
- Nút "Decryption": Khi nhấn sẽ thực hiện chức năng giải mã văn bản (được định nghĩa trong lớp con).
- Nút "Exit": Khi nhấn sẽ xóa tất cả dữ liệu trên giao diện và quay lại màn hình chính.

• Kiểm tra và hiển thị lỗi:

- Giao diện có tính năng kiểm tra đầu vào, không cho phép nhập các ký tự không hợp lệ (như số và ký tự đặc biệt) vào văn bản và khóa.

- Quy trình xử lý:

○ Khởi tạo giao diện:

- **Phương thức Screen()** tạo ra một JPanel chứa tất cả các thành phần giao diện. Các thành phần này bao gồm nhãn (JLabel), các vùng nhập văn bản (JTextArea), các nút điều khiển (JButton), và các trường nhập khóa (JTextField).
- **Thanh cuộn (JScrollPane)** được thêm vào các vùng nhập văn bản (plaintext và ciphertext) để giúp người dùng dễ dàng nhập và xem dữ liệu lớn hơn.

○ Kiểm tra đầu vào:

- **Phương thức showErrorWhenInputWrong()** kiểm tra dữ liệu nhập vào các trường plaintext, ciphertext và key. Nếu phát hiện lỗi chương trình sẽ hiển thị thông báo lỗi qua JOptionPane.

○ Xử lý ký tự không hiển thị:

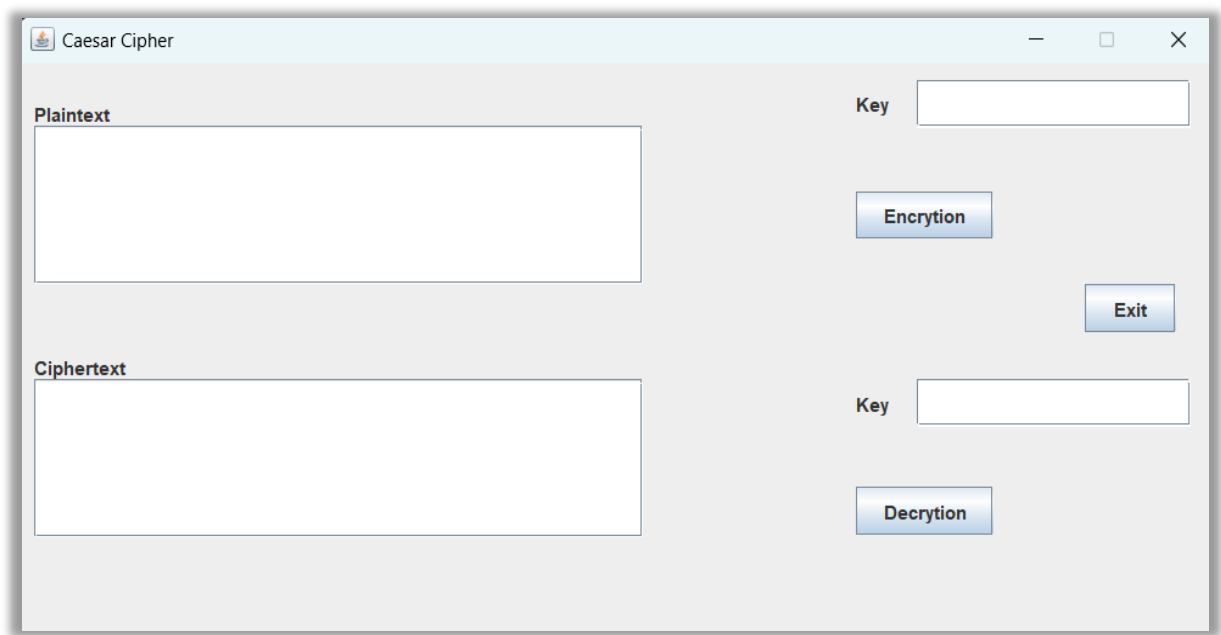
- **Phương thức removeInvisibleChars()** giúp loại bỏ các ký tự không hiển thị để đảm bảo văn bản được xử lý một cách chính xác trước khi mã hóa hoặc giải mã.

Caesar

Hướng dẫn sử dụng

- **Sử dụng để mã hóa:**
 1. Nhập key (nhập số từ 0 -> 25), không nhập số và các kí tự đặc biệt
 2. Nhập text (không nhập số và các kí tự đặc biệt) ở ô plaintext
 3. Bấm “Encryption” để mã hóa và text mã hóa sẽ được xuất ra ở Ciphertext
- **Sử dụng để giải mã:**
 1. Nhập key (nhập số từ 0 -> 25), không nhập số và các kí tự đặc biệt
 2. Nhập text (không nhập số và các kí tự đặc biệt) ở ô ciphertext
 3. Bấm “Decryption” để giải mã và text giải mã sẽ được xuất ra ở Plaintext

Giao diện:



The screenshot shows a window titled "Caesar Cipher" with a standard Windows-style title bar (minimize, maximize, close buttons). The interface is divided into two main sections for encryption and decryption.

Encryption Section (Top):

- Plaintext:** A large text input area on the left.
- Key:** A text input field on the right.
- Encryption:** A button located below the Key field.
- Exit:** A button located to the right of the Encryption button.

Decryption Section (Bottom):

- Ciphertext:** A large text input area on the left.
- Key:** A text input field on the right.
- Decryption:** A button located below the Key field.

- **Cấu trúc:** Màn hình Caesar kế thừa từ lớp cha subScreen, giúp tái sử dụng các chức năng cơ bản và chia sẻ thiết lập màn hình chính của ứng dụng. Lớp này có phương thức chính Screen().
- **Tính năng:**
 - o mã hóa và giải mã: Người dùng nhập khóa vào textfield và nhấn nút "Encryption" hoặc "Decryption". Chương trình sẽ sử dụng lớp Caesar để thực hiện chức năng và hiển thị kết quả vào ô Plaintext hoặc Ciphertext.
- **Mã hóa và giải mã:**
 Thông qua lựa chọn ("Encryption", "Decryption"), chương trình sẽ thực hiện:
 - o Nếu chọn "Encryption", chương trình sẽ mã hóa văn bản từ trường "plaintext" và xuất kết quả vào trường "ciphertext".
 - o Nếu chọn "Decryption", chương trình sẽ giải mã văn bản từ trường "ciphertext" và xuất kết quả vào trường "plaintext".
- **Quy trình xử lý:**
 1. Nhập khóa.
 2. Kiểm tra tính hợp lệ của Khóa nếu không hợp lệ, hiển thị thông báo và nhập lại khóa.
 3. Thực hiện mã hóa hoặc giải mã thông qua lớp Caesar.
- **Bắt lỗi ở giao diện:** Màn hình có tính năng kiểm tra đầu vào nhằm đảm bảo rằng người dùng không nhập vào các ký tự không hợp lệ như số hay ký tự đặc biệt. Nếu phát hiện lỗi, chương trình sẽ hiển thị thông báo và yêu cầu người dùng nhập lại dữ liệu hợp lệ.

Thuật toán:

- **Mô tả:** Thuật toán này thuộc nhóm mã hóa thay thế, trong đó mỗi ký tự trong thông điệp gốc (plaintext) được thay thế bằng một ký tự khác dựa trên việc dịch chuyển vị trí trong bảng chữ cái.
- **Bảng chữ cái (alphabet):** 26 chữ cái.
- **Mã hóa :** Thực hiện lặp từng chữ cái trong bản rõ đồng thời kiểm tra ký tự đó có phải chữ cái hay không nếu phải thì thực hiện mã hóa bằng cách dịch chuyển vị trí của ký tự đến vị trí trong bảng chữ cái với key
VD : P = "ABCF" : Key = 1
 $'A' + 1 = 'B'$, $'B' + 1 = 'C'$, $'C' + 1 = 'D'$, $'F' + 1 = 'G'$
 > C = "BCDJ"
- **Giải mã:** lặp qua từng ký tự của ciphertext và dịch chuyển quay lại vị trí ban đầu với key
VD: C = "BCDJ" : key = 1
 $'B' - 1 = 'A'$, $'C' - 1 = 'B'$, $'D' - 1 = 'C'$, $'J' - 1 = 'I'$

Bảng chữ đơn

Hướng dẫn sử dụng

- Sử dụng để mã hóa:

1. Bấm “Random Key” để được một Key gồm 25 kí tự (có thể tự nhập), không nhập số và các kí tự đặc biệt
2. Nhập text (không nhập số và các kí tự đặc biệt) ở ô plaintext
3. Bấm “Encryption” để mã hóa và text mã hóa sẽ được xuất ra ở Ciphertext

- Sử dụng để giải mã:

1. Bấm “Random Key” để được một Key gồm 25 kí tự (có thể tự nhập), không nhập số và các kí tự đặc biệt
2. Nhập text (không nhập số và các kí tự đặc biệt) ở ô ciphertext
3. Bấm “Decryption” để giải mã và text giải mã sẽ được xuất ra ở Plaintext

Giao diện:

The screenshot shows a Windows-style application window titled "Bảng Chữ Đơn". The interface is divided into two main sections for encryption and decryption. On the left, there are two large text input fields: "Plaintext" at the top and "Ciphertext" at the bottom. On the right, there are two "Key" input fields, each followed by a button labeled "Random Key" that provides a 25-character alphanumeric key. The top section includes an "Encryption" button, and the bottom section includes a "Decryption" button. An "Exit" button is located in the middle-right area of the window.

- **Cấu trúc:** Màn hình Chuyển dịch dòng kế thừa từ lớp cha subScreen, giúp tái sử dụng các chức năng cơ bản và chia sẻ thiết lập màn hình chính của ứng dụng. Lớp này có phương thức chính Screen().
- **Tính năng:**
 - Tạo khóa mã hóa: Nhấn nút Random key để tạo khóa và sử dụng nó cho quá trình mã hóa hoặc giải mã.
- **Mã hóa và giải mã:**
 Thông qua lựa chọn ("Encryption", "Decryption"), chương trình sẽ thực hiện:
 - Nếu chọn "Encryption", chương trình sẽ mã hóa văn bản từ trường "plaintext" và xuất kết quả vào trường "ciphertext".
 - Nếu chọn "Decryption", chương trình sẽ giải mã văn bản từ trường "ciphertext" và xuất kết quả vào trường "plaintext".
-
- **Quy trình xử lý:**
 - 1 Chọn nút RanDomkey để tạo key, nhập Bản rõ hoặc Bản mã
 - 2 Chọn chức năng mã hóa hoặc Giải mã
 - 3 Chương trình loại bỏ các ký tự không thể hiển thị hoặc không hợp lệ từ văn bản đầu vào (plaintext hoặc ciphertext) và khóa mã hóa.
 - 4 Mã hóa: Sử dụng lớp Bangchudon để mã hóa văn bản, sau đó hiển thị kết quả trong trường cipherText.
 - 5 Giải mã: Sử dụng lớp Bangchudon để giải mã văn bản, sau đó hiển thị kết quả trong trường plaintext.
- **Bắt lỗi ở giao diện:** Màn hình có tính năng kiểm tra đầu vào nhằm đảm bảo rằng người dùng không nhập vào các ký tự không hợp lệ như số hay ký tự đặc biệt. Nếu phát hiện lỗi, chương trình sẽ hiển thị thông báo và yêu cầu người dùng nhập lại dữ liệu hợp lệ.

Thuật toán:

- **Mô tả:** Thuật toán Bảng chữ đơn (Monoalphabetic Cipher) sử dụng khóa mã hóa là một hoán vị của bảng chữ cái. Văn bản gốc (plaintext) sẽ được mã hóa bằng cách thay thế mỗi ký tự trong văn bản bằng ký tự tương ứng trong bảng chữ đơn..
- **Tạo khóa mã hóa:** Người dùng tạo mã khóa và nhập bản rõ hoặc bản mã, chương trình sẽ xử lý để loại bỏ các ký tự không hợp lệ,
- **Mã hóa văn bản:**
 Lặp qua từng ký tự trong bản rõ, mỗi chữ của bản rõ được ánh xạ đến một chữ khác nhau của bản mã
 Ví dụ : p = STU : Key: SNGDEHZRATOYMFXBUIQPCVJLWK ,

Bảng chữ cái: ABCDEFGHIJKLMNOPQRSTUVWXYZ

- Tìm vị trí của chữ S trong Bảng chữ cái -> vị trí: 18, lấy vị trí tương ứng với số 18 trong key -> Q, với chữ T và U làm tương tự ta được C = QPC
- **Giải mã văn bản:**
Lặp qua từng kí tự trong Ciphertext mỗi chữ của Ciphertext được ánh xạ đến một chữ khác nhau của bản mã
Ví dụ : C = QPC: Key: SNGDEHZRATOYMFXBUIQPCVJLWK ,
Bảng chữ cái: ABCDEFGHIJKLMNOPQRSTUVWXYZ
- Tìm vị trí của chữ Q trong Key -> vị trí: 18, lấy vị trí tương ứng với số 18 trong Bảng chữ cái -> Q, với chữ T và U làm tương tự ta được P = STU;

Playfair

Hướng dẫn sử dụng

- Sử dụng để mã hóa:

Nhập Key (không nhập số và các kí tự đặc biệt)

Nhập text (không nhập số và các kí tự đặc biệt) ở ô plaintext

Chọn “Encryption” và bấm Start để mã hóa và text mã hóa sẽ được xuất ra ở Ciphertext

- Sử dụng để giải mã:

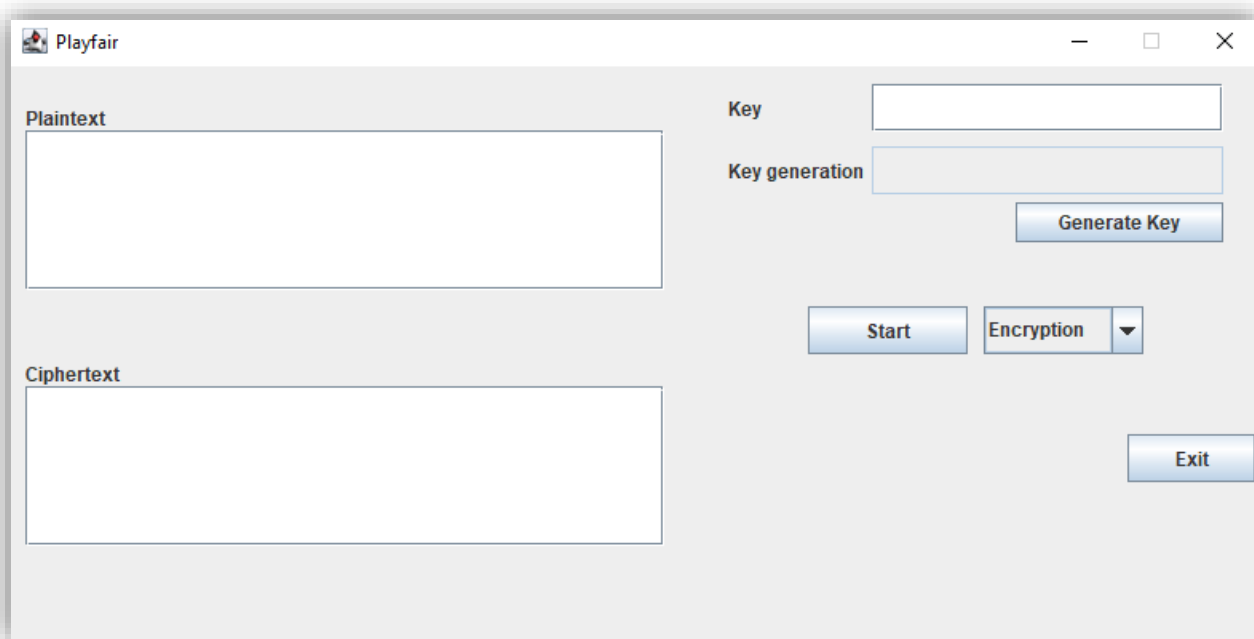
Nhập Key (không nhập số và các kí tự đặc biệt)

Nhập text (không nhập số và các kí tự đặc biệt) ở ô ciphertext

Chọn “Decryption” để giải mã và text giải mã sẽ được xuất ra ở Plaintext

- Generate Key: Tạo ra “Key generation” với Key + bảng chữ cái (phải có Key)

Giao diện:



The screenshot shows a Windows-style application window titled "Playfair". The interface is divided into several sections:

- Plaintext:** A large text input area on the left side.
- Ciphertext:** A large text input area at the bottom left.
- Key:** A text input field at the top right.
- Key generation:** A text input field below the Key field.
- Generate Key:** A button located below the Key generation field.
- Start:** A button located below the Generate Key button.
- Encryption:** A dropdown menu with a downward arrow, currently showing "Encryption".
- Exit:** A button located at the bottom right.

- **Cấu trúc:** Màn hình Playfair kế thừa từ lớp cha subScreen, giúp tái sử dụng các chức năng cơ bản và chia sẻ thiết lập màn hình chính của ứng dụng. Lớp này có phương thức chính Screen().
- **Tính năng:**
 - o Tạo khóa mã hóa: Người dùng nhập khóa vào textfield và nhấn nút "Generate Key". Mã sẽ loại bỏ các ký tự không hợp lệ và tạo khóa mã hóa bằng cách sử dụng lớp PlayFair. Nếu khóa hợp lệ, chương trình sẽ hiển thị khóa đã được xử lý trong trường "Key Generation".
 - o **Quy trình xử lý:**
 - Loại bỏ các ký tự đặc biệt không hiển thị.
 - Kiểm tra tính hợp lệ của khóa.
 - Sử dụng lớp PlayFair để sinh khóa và hiển thị.
- **Mã hóa và giải mã:** Thông qua combobox choice ("Encryption", "Decryption"), dựa vào lựa chọn, chương trình sẽ thực hiện:
 - Nếu chọn "Encryption", chương trình sẽ mã hóa văn bản từ trường "plaintext" và xuất kết quả vào trường "ciphertext".
 - Nếu chọn "Decryption", chương trình sẽ giải mã văn bản từ trường "ciphertext" và xuất kết quả vào trường "plaintext".
- **Quy trình xử lý:**
 1. Lựa chọn chế độ mã hóa hoặc giải mã từ combobox choice.
 2. Loại bỏ các ký tự đặc biệt không cần thiết từ văn bản.
 3. Kiểm tra tính hợp lệ của văn bản đầu vào.
 4. Thực hiện mã hóa hoặc giải mã thông qua lớp PlayFair.
- **Bắt lỗi ở giao diện:** Màn hình có tính năng kiểm tra đầu vào nhằm đảm bảo rằng người dùng không nhập vào các ký tự không hợp lệ như số hay ký tự đặc biệt. Nếu phát hiện lỗi, chương trình sẽ hiển thị thông báo và yêu cầu người dùng nhập lại dữ liệu hợp lệ.

Thuật toán:

- **Mô tả:** Là một phương pháp mã hóa dựa trên mật mã thay thế cặp (digraph substitution cipher). Thay vì mã hóa từng chữ cái, Playfair mã hóa cặp chữ, giúp tăng độ phức tạp cho mã hóa, vì khi mã hóa theo từng cặp chữ, tần số xuất hiện của từng chữ cái sẽ không thể hiện rõ như trong mã hóa đơn chữ.
- **Bảng chữ cái (alphabet):** 26 chữ cái, với I và J được xem là một chữ cái duy nhất để giữ cho bảng là 5x5 ô vuông.
- **array2d:** Mảng 2D (5x5): Chứa bảng chữ cái được tạo ra từ khóa (key) và các chữ cái còn lại của bảng chữ cái.
- **Chuẩn bị khóa (Key), chuẩn hóa Key**
 - o **Khử các ký tự lặp trong khóa:** Khóa được xử lý để loại bỏ các ký tự lặp lại. Ví dụ: "ABCAA" sẽ trở thành "ABC".

- **Xử lý ký tự I và J:** Khi gặp I hoặc J trong khóa, chỉ giữ lại một trong hai và loại bỏ ký tự còn lại.
- **Nhập khóa vào bảng 5x5:** Khóa sau khi chuẩn hóa sẽ được đưa vào bảng 5x5 trước. Sau đó, các chữ cái còn lại sẽ được thêm vào bảng theo thứ tự của bảng chữ cái (trừ I hoặc J tùy vào khóa).

- **Xử lý plaintext (Văn bản cần mã hóa)**

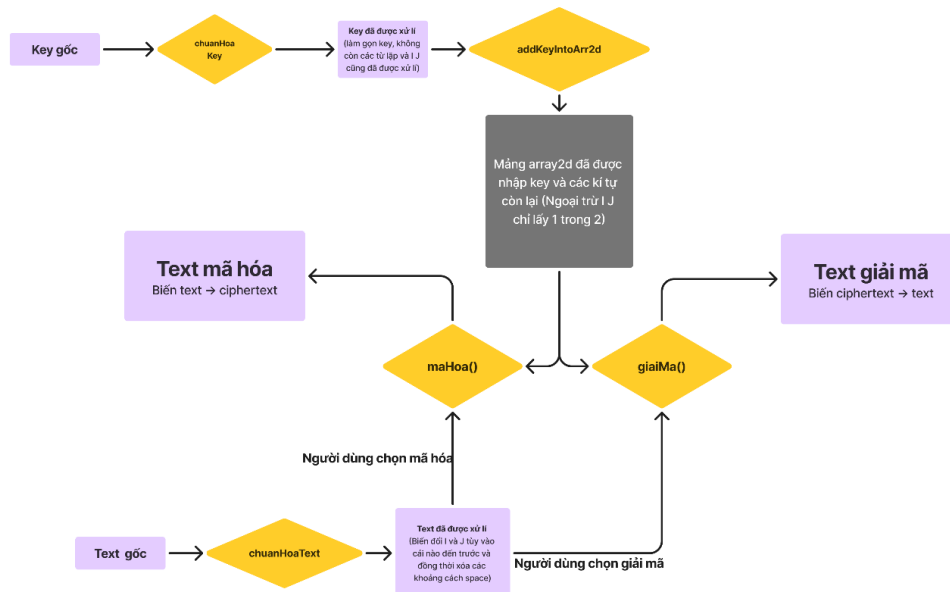
- **Chuẩn hóa:** Văn bản sẽ được chuyển thành chữ in hoa và loại bỏ các khoảng trắng. Nếu có cặp chữ cái giống nhau đứng gần nhau (ví dụ AA), thì một ký tự thay thế sẽ được thêm vào giữa (thường là X).
- **Xử lý I và J và khoảng cách:** Biến đổi I và J tùy vào cái nào đến trước và không lấy khoảng cách

- **Mã hóa**

- **Cùng hàng:** Nếu hai chữ cái trong cùng một hàng, mỗi chữ cái sẽ được thay thế bằng chữ cái tiếp theo bên phải (quay vòng nếu hết hàng).
- **Cùng cột:** Nếu hai chữ cái trong cùng một cột, mỗi chữ cái sẽ được thay thế bằng chữ cái phía dưới (quay vòng nếu hết cột).
- **Khác hàng và cột:** Nếu hai chữ cái không cùng hàng và cột, chúng sẽ được thay thế bằng các chữ cái ở góc đối diện của hình chữ nhật mà chúng tạo thành.

- **Giải mã**

- **Cùng hàng:** Di chuyển sang trái thay vì phải.
- **Cùng cột:** Di chuyển lên thay vì xuống.
- **Khác hàng và cột:** Tương tự như khi mã hóa, tìm các chữ cái ở góc đối diện của hình chữ nhật.



Vegenere

Hướng dẫn sử dụng

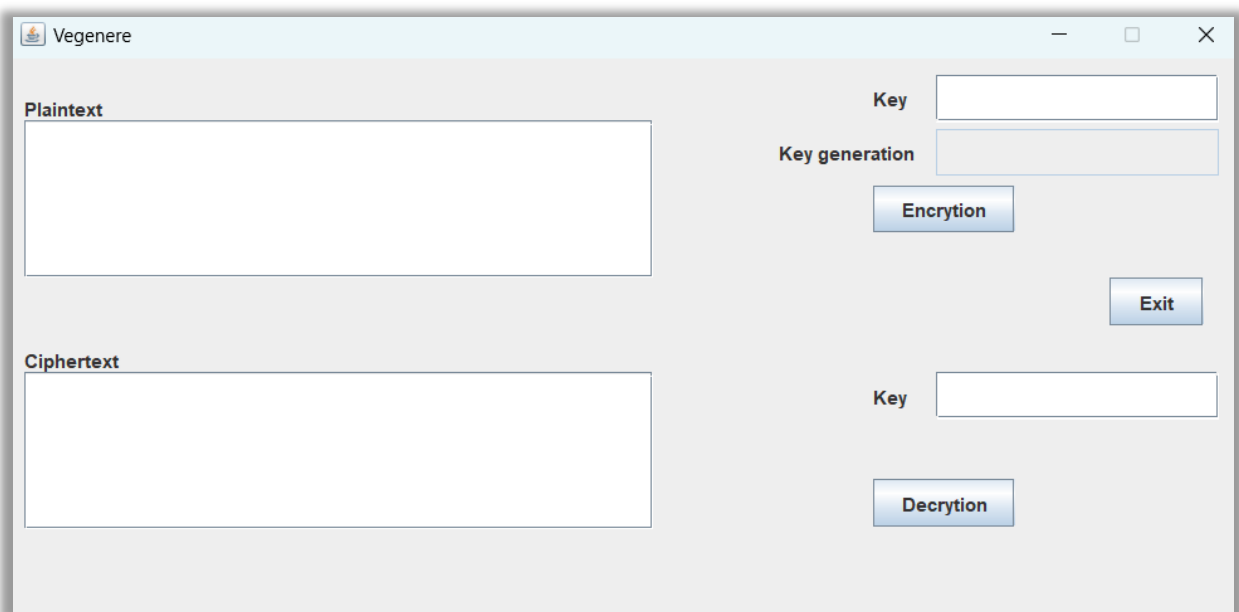
- Sử dụng để mã hóa:

1. Nhập Key ((không nhập số và các kí tự đặc biệt)
2. Nhập text (không nhập số và các kí tự đặc biệt) ở ô plaintext
3. Bấm “Encryption” để mã hóa và text mã hóa sẽ được xuất ra ở Ciphertext

- Sử dụng để giải mã:

1. Nhập Key (không nhập số và các kí tự đặc biệt)
2. Nhập text (không nhập số và các kí tự đặc biệt) ở ô ciphertext
3. Bấm “Decryption” để giải mã và text giải mã sẽ được xuất ra ở Plaintext

Giao diện:



- **Cấu trúc:** Màn hình Chuyển dịch dòng kế thừa từ lớp cha subScreen, giúp tái sử dụng các chức năng cơ bản và chia sẻ thiết lập màn hình chính của ứng dụng. Lớp này có phương thức chính Screen().
 - **Tính năng:**
 - o Cho phép nhập khóa, Nhập Plaintext, Ciphertext và thực hiện chức năng mã hóa hoặc giải mã
- Mã hóa và giải mã:**

Thông qua lựa chọn ("Encryption", "Decryption"), chương trình sẽ thực hiện:

- Nếu chọn "Encryption", chương trình sẽ mã hóa văn bản từ trường "plaintext" và xuất kết quả vào trường "ciphertext".
- Nếu chọn "Decryption", chương trình sẽ giải mã văn bản từ trường "ciphertext" và xuất kết quả vào trường "plaintext".

- **Quy trình xử lý:**

- 1 Nhập Key , nhập Bản rõ hoặc Bản mã
- 2 Chọn chức năng mã hóa hoặc Giải mã
- 3 Chương trình sẽ kiểm tra tính hợp lệ của khóa nếu hợp lệ thì tiếp tục chương trình và hiển thị khóa mới có độ dài bằng với bản rõ nếu không thì hiển thị lỗi và nhập lại khóa.
- 4 Mã hóa: Sử dụng lớp Vegenere để mã hóa văn bản, sau đó hiển thị kết quả trong trường cipherText.
- 5 Giải mã: Sử dụng lớp Vegenere để giải mã văn bản, sau đó hiển thị kết quả trong trường plaintext.

- **Bắt lỗi ở giao diện:** Màn hình có tính năng kiểm tra đầu vào nhằm đảm bảo rằng người dùng không nhập vào các ký tự không hợp lệ như số hay ký tự đặc biệt. Nếu phát hiện lỗi, chương trình sẽ hiển thị thông báo và yêu cầu người dùng nhập lại dữ liệu hợp lệ.

Thuật toán:

- **Mô tả:** Ta sử dụng từ khoá để chỉ rõ chọn bảng nào được dùng cho từng chữ trong bản tin. Sử dụng lần lượt các bảng theo từ khoá đó và lặp lại từ đầu sau khi kết thúc từ khoá. Độ dài khoá là chu kỳ lặp của các bảng chữ

Tạo khóa mã hóa: Người dùng tạo mã khóa và nhập bản rõ hoặc bản mã, chương trình sẽ xử lý để loại bỏ các ký tự không hợp lệ.

- **Mã hóa văn bản:**

Lặp lại khóa sao cho bằng với độ dài của bản mã

Sử dụng mỗi chữ của từ khoá như khoá của mã Caesar Mã chữ tương ứng của bản rõ với bước nhảy tương ứng

VD:

key: de -> keynew = dede

plaintext: wear

w + d -> Z

e + e -> I

a + d -> D

r + e -> V

C : ZIDV

- **Giải mã văn bản:**

Lập lại khóa sao cho bằng với độ dài của bản mã

Sử dụng mỗi chữ của từ khoá như khoá của mã Ceasar Mã chữ tương ứng của Ciphertext với bước nhảy lùi tương ứng

VD:

key: de -> keynew = dede

Ciphertext: ZIDV

z - d -> W

i - e -> E

d - d -> A

v - e -> R

C : WEAR

Chuyển dịch dòng

Hướng dẫn sử dụng

- Sử dụng để mã hóa:

1. Nhập Key (không được nhập kí tự đặc biệt)
2. Nhập text (không nhập số và các kí tự đặc biệt) ở ô plaintext
3. Chọn “Encryption” và bấm Start để mã hóa và text mã hóa sẽ được xuất ra ở Ciphertext

- Sử dụng để giải mã:

1. Nhập Key (không được nhập kí tự đặc biệt)
2. Nhập text (không nhập số và các kí tự đặc biệt) ở ô ciphertext
3. Bấm “Decryption” để giải mã và text giải mã sẽ được xuất ra ở Plaintext

Giao diện:

The screenshot shows a Windows-style application window titled "Chuyển dịch dòng". The interface is divided into two main sections for input and output. On the left, there are two large text input fields: the top one is labeled "Plaintext" and the bottom one is labeled "Ciphertext". On the right side, there is a "Key" input field, a "Start" button, and a dropdown menu currently set to "Encryption". At the bottom right, there is an "Exit" button. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

- **Cấu trúc:** Màn hình Chuyển dịch dòng kế thừa từ lớp cha subScreen, giúp tái sử dụng các chức năng cơ bản và chia sẻ thiết lập màn hình chính của ứng dụng. Lớp này có phương thức chính Screen().
- **Tính năng:**
 - Tạo khóa mã hóa: Người dùng có thể nhập khóa vào trường textfield và sử dụng nó cho quá trình mã hóa hoặc giải mã. Trước khi thực hiện, chương trình sẽ kiểm tra khóa nhập vào, loại bỏ các ký tự không hợp lệ và tiếp tục thực hiện mã hóa hoặc giải mã.
 - **Quy trình xử lý:**
 - Loại bỏ các ký tự đặc biệt không hiển thị.
 - Kiểm tra tính hợp lệ của khóa.
- **Mã hóa và giải mã:** Thông qua combobox choice ("Encryption", "Decryption"), dựa vào lựa chọn, chương trình sẽ thực hiện:
 - Nếu chọn "Encryption", chương trình sẽ mã hóa văn bản từ trường "plaintext" và xuất kết quả vào trường "ciphertext".
 - Nếu chọn "Decryption", chương trình sẽ giải mã văn bản từ trường "ciphertext" và xuất kết quả vào trường "plaintext".
- **Quy trình xử lý:**
 - Người dùng lựa chọn chế độ xử lý (mã hóa hoặc giải mã) từ combobox choice.
 - Chương trình loại bỏ các ký tự không thể hiển thị hoặc không hợp lệ từ văn bản đầu vào (plaintext hoặc ciphertext) và khóa mã hóa.
 - Kiểm tra tính hợp lệ của dữ liệu đầu vào để đảm bảo không có ký tự đặc biệt hoặc số trong khóa.
 - Dựa trên lựa chọn của người dùng:
 - Mã hóa: Sử dụng lớp ChuyenDichDong để mã hóa văn bản, sau đó hiển thị kết quả trong trường cipherText.
 - Giải mã: Sử dụng lớp ChuyenDichDong để giải mã văn bản, sau đó hiển thị kết quả trong trường plaintext.
- **Bắt lỗi ở giao diện:** Màn hình có tính năng kiểm tra đầu vào nhằm đảm bảo rằng người dùng không nhập vào các ký tự không hợp lệ như số hay ký tự đặc biệt. Nếu phát hiện lỗi, chương trình sẽ hiển thị thông báo và yêu cầu người dùng nhập lại dữ liệu hợp lệ.

Thuật toán:

- Mô tả: **Lớp Chuyển Dịch Dòng** được sử dụng để mã hóa và giải mã văn bản dựa trên phương pháp chuyển dịch dòng, trong đó các ký tự của văn bản sẽ được xếp vào mảng 2D và sắp xếp theo thứ tự của khóa mã hóa.

- **Tạo khóa mã hóa:** Người dùng nhập khóa vào, chương trình sẽ xử lý để loại bỏ các ký tự không hợp lệ, tạo khóa phù hợp bằng cách chuyển đổi các ký tự của khóa thành các số thứ tự và lưu vào mảng.
- **Mã hóa văn bản:**
 - o Văn bản sẽ được chia thành các dòng và cột dựa trên độ dài của khóa, sau đó các ký tự sẽ được xếp vào mảng 2D.
 - o Sau khi xếp xong, các ký tự sẽ được đọc theo thứ tự của khóa và kết quả mã hóa sẽ được xuất ra dưới dạng chuỗi.
- **Giải mã văn bản:**

Quá trình giải mã tương tự như mã hóa nhưng ngược lại. Văn bản mã hóa sẽ được xếp vào mảng 2D theo thứ tự của khóa và sau đó sẽ được đọc lại theo thứ tự dòng và cột để khôi phục văn bản gốc.

