

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



SOICT

Project 1 (IT3150)

ĐỀ TÀI 11

Xây dựng một phần mềm ứng dụng chuyển đổi hệ số cơ sở

GIẢNG VIÊN HƯỚNG DẪN : TS. NGUYỄN THỊ THU HƯƠNG

Mã HỌC PHẦN : IT3150

TÊN MÃ HỌC PHẦN : PROJECT I

HỌC KÌ : 2022.3

LỚP : 732378

SINH VIÊN THỰC HIỆN : SOK SOKONG

MSSV : 20211005

HÀ NỘI - 2023

Muc Luc

System number Converter

Chương 1 : MÔ TẢ TỔNG QUAN ĐỀ TÀI	5
1. Bài Toán Lực Chọn.....	5
1.1 Mô Tả Bài Toán.....	5
1.2 Công Nghệ Sử Dụng.....	5
1.3 Các Yêu cầu.....	6
Chương 2 : ĐẶC TẢ CÁC HỆ THỐNG SỐ.....	7
2. Hệ Thống Số.....	7
a. Chuyển đổi giữa Binary and decimal.....	8
b. Chuyển đổi giữa Hexadecimal và Decimal.....	8
c. Chuyển đổi giữa Binary và Hexadecimal numbers	9
d. Chuyển Đổi Từ Octal to Decimal.....	10
e. Chuyển Đổi Từ Decimal Sang Octal.....	10
2.1. Hệ Thống số Binary (Nhị phân).....	11
2.2 Hệ Thống số Decimal (Thập phân).....	11
2.3 Hệ Thống số Hexadecimal (Thập lục phân).....	12
2.4. Hệ Thống số Octal (Bát phân).....	12
2.5 Ưu điểm của các hệ thống số.....	13
2.6 Ứng dụng của các hệ thống số.....	13
Chương 3 : THUẬT TOÁN TRONG ỨNG DỤNG CHUYỂN ĐỔI HỆ SỐ SỐ TRONG JAVA.....	14
3. Thuật Toán	
3.1 Thuật Toán Chuyển Đổi Từ Hệ 10 Sang Hệ Khác.....	14
3.1.1 Chuyển Đổi Từ Decimal Sang Binary.....	14
3.1.2 Chuyển Đổi Từ Decimal Sang Octal và Hexadecimal.....	15
3.2 Thuật Toán Chuyển Đổi Từ Hệ Khác Sang Decimal.....	15
3.3 Xử Lý Số Thực Và Số Âm.....	16
3.3.1 Chuyển Đổi Số Thực.....	16
3.3.2 Xử Lý Số Âm.....	16
3.4 Tối Ưu Hóa Và Mở Rộng.....	16
Chương 4 : Giao Diện Người Dùng (User Interface).....	17
4. Giao Diện Người Dùng.....	17
4.1 Mô Tả Giao Diện Người Dùng	17
4.2 Các Chức Năng Của Giao Diện	18
4.3 Hướng Dẫn Sử dụng.....	18
4.4 Mục Tiêu Thiết Kế Giao Diện.....	20
4.5 Thách Thức Trong Thiết Kế Giao Diện.....	20
4.6 Hình Ảnh Minh Họa.....	20

ASCII Converter

Chương 1 : MÔ TẢ TỔNG QUAN ĐỀ TÀI.....	22
1.1 Mô Tả Bài Toán.....	22
1.2 Công Nghệ Sử Dụng.....	23
1.3 Các Yêu cầu.....	23
Chương 2 : Thiết Kế và Cài Đặt Giao Diện.....	24
2.1 Thiết Kế Giao Diện.....	24
2.2 Cài Đặt Giao Diện.....	25
2.3 Giao Diện Cuối Cùng.....	25
Chương 3 : Triển Khai và Kiểm Thử.....	26
3.1 Triển Khai Ứng Dụng.....	26
3.2 Kiểm Thử.....	26
3.3 Ghi Chú Về Quản Lý Phiên Bản.....	27
Chương 4 : Thuật Toán và Cách Tính Toán.....	28
4.1 Thuật Toán Chuyển Đổi ASCII.....	28
4.2 Tính Toán Checksum.....	29
Chương 5 : Giao Diện Người Dùng (User Interface).....	30
5.1 Mô Tả Giao Diện Người Dùng.....	30
5.2 Các Chức Năng Của Giao Diện.....	31
5.3 Mục Tiêu Thiết Kế Giao Diện.....	31
5.4 Thách Thức Trong Thiết Kế Giao Diện.....	32
5.5 Hình Ảnh Minh Họa.....	32
Chương: Kết Luận Chung.....	35
1. Kết Luận	35
2. Reference	35

LỜI NÓI ĐẦU

Ngày nay, ứng dụng công nghệ thông tin ngày càng phát triển và công nghệ thông tin là một những yếu tố mang tính quyết định trong hoạt động của các tổ chức, ngân hàng, công ty, ý tế và giáo dục. Công nghệ đóng vai trò hết sức quan trọng, có thể tạo ra những bước đột phá mạnh mẽ.

Dự án “ Chuyển Đổi Hệ số” là một phần quan trọng của nỗ lực của chúng tôi để tạo ra một ứng dụng đơn giản nhưng hữu ích cho cộng đồng. Việc chuyển đổi giữa các hệ số số học là một nhiệm vụ thường xuyên đối với nhiều người, đặc biệt là trong lĩnh vực khoa học máy tính và kỹ thuật. Dự án này đã bắt đầu như một cơ hội học tập và trải nghiệm thực tế, và đóng góp vào sự phát triển của tôi trong lĩnh vực công nghệ thông tin. Dự án này không chỉ đơn giản thuần là một nhiệm vụ học tập, mà còn là một cơ hội để thực hiện những ý tưởng, khám phá các thách thức, và cùng nhau xây dựng điều đó hữu ích cho cộng đồng.

Trong báo cáo này, tôi sẽ giới thiệu về dự án của chúng tôi, mục tiêu ban đầu, và quá trình phát triển. Tôi cũng sẽ thảo luận về thử nghiệm và kiểm tra đã thực hiện để đảm bảo tính năng và độ tin cậy của ứng dụng.

Dự án này không thể thực hiện mà không có sự hỗ trợ và hướng dẫn từ giảng viên, đồng nghiệp, và bạn bé. Em xin được gửi lời cảm ơn chân thành đến cô hướng dẫn đề tài **NGUYỄN THỊ THU HƯƠNG**, Giảng viên Khoa Khoa học Máy tính Đại học Bách Khoa Hà Nội - đã hết lòng giúp đỡ, hướng dẫn, chỉ dạy tận tình để em hoàn thành đề tài này. Em xin bày tỏ lòng biết ơn sâu sắc đến tất cả những người đã đóng góp vào dự án và luôn ủng hộ em trong quá trình học tập.

Cảm ơn mọi người đã dành thời gian để đọc báo cáo này.

Trân trọng,

SOK SOKONG

System number Converter

Chương 1 : MÔ TẢ TỔNG QUAN ĐỀ TÀI

1. Bài Toán Lựa Chọn

1.1 Mô Tả Bài Toán

Ứng dụng chuyển đổi hệ số cơ sở là một ứng dụng phát triển bằng Java Swing, cho phép người dùng chuyển đổi giữa các hệ số cơ sở phổ biến : Binary (nhị phân), Octal (bát phân), Decimal (Thập phân), và Hexadecimal (thập lục phân). Mục đích của ứng dụng là giúp người sử dụng dễ dàng thực hiện các phép chuyển đổi giữa các hệ số cơ sở này một cách hiệu quả.

Bài toán lựa chọn trong đề tài này liên quan đến việc chọn lựa công nghệ hoặc phương pháp cụ thể để giải quyết một vấn đề hoặc tác vụ cụ thể. Trong trường hợp này, bài toán lựa chọn là về việc xây dựng một phần mềm ứng dụng để chuyển đổi giữa các hệ thống số như Binary, Decimal, Octal and Hexadecimal.

1.2 Công Nghệ Sử Dụng

Công nghệ sử dụng trong đề tài này ngôn ngữ Java. Java là một ngữ lập trình phổ biến và mạnh mẽ, có khả năng chạy trên nền tảng khác nhau. Lý do chúng em đã chọn Java cho dự án này bao gồm :

- **Tính di động** : Java là một ngôn ngữ di động, có khả năng chạy trên nhiều hệ điều hành và môi trường khác nhau. Điều này cho phép ứng dụng hoạt động trên nhiều loại máy tính và thiết bị khác nhau mà không cần viết lại mã nguồn cho từng nền tảng riêng biệt.
- **Đa Nền Tảng** : Java được thiết kế để hỗ trợ việc phát triển ứng dụng đa nền tảng.

- **Giao Diện Người Dùng Đồ Họa** : Java cung cấp thư viện đồ họa mạnh mẽ, cho phép thiết kế giao diện người dùng đẹp và dễ sử dụng cho ứng dụng của mình.

1.3 Các Yêu cầu

Các yêu cầu cụ thể của dự án bao gồm :

- Xây dựng một ứng dụng Java có khả năng chuyển đổi giữa các hệ thống số quan trọng như Binary, Decimal, Octal and Hexadecimal.
- Thiết kế giao diện người dùng dễ sử dụng cho ứng dụng, bao gồm ô nhập liệu, hộp chọn cho hệ thống số và đích, nút chuyển đổi và nút đặt lại.
- Hỗ trợ chuyển đổi số có phần thập phân, cho phép người dùng làm việc số thực.
- Hiển thị kết quả chuyển đổi một cách dễ đọc trên giao diện.
- Cung cấp tính năng đặt lại để xóa dữ liệu đã nhập và kết quả chuyển đổi .
- Có khả năng mở rộng dự án trong tương lai bằng cách thêm tính năng hỗ trợ chuyển đổi từ và sang các hệ thống khác như hệ bát phân.

Chương 2 : ĐẶC TẢ CÁC HỆ THỐNG SỐ

2. Hệ Thống Số

Máy sử dụng số nhị phân bên trong vì máy tính được tạo ra một cách tự nhiên để lưu trữ và xử lý các số 0 và 1. Hệ thống số nhị phân có hai số chữ số 0 và 1. Một số hoặc ký tự được lưu dưới dạng chuỗi 0 và 1. Mỗi 0 hoặc 1 được gọi là một bit (chữ số nhị phân).

Tuy nhiên để xử lý một hệ điều hành, chúng ta tiếp cận “ cấp độ máy” bằng cách sử dụng số thập phân. Số thập phân có xu hướng rất dài và cồng kềnh. Thông thường các số thập lục phân được sử dụng để viết tắt chúng, với mỗi chữ số thập lục phân đại diện cho bốn chữ số thập phân. Hệ thống số thập lục phân có 16 chữ số : 0-9 và A-F. Các chữ cái A, B, C, D, E, F tương ứng với các số thập phân 10, 11, 12, 13, 14, 15.

Các chữ số trong hệ thống số thập phân là 0, 1, 2, 3, 4, 5, 6, 7, 8 và 9. Số thập phân được biểu thị bằng một dãy gồm một hoặc nhiều chữ số này. Giá trị mà mỗi chữ số biểu thị phụ thuộc vào vị trí của nó, biểu thị lũy thừa nguyên của 10.

Ví dụ : các chữ số 7, 4, 2 và 3 trong số thập phân 7423 lần lượt đại diện cho 700, 400, 20 và 3 như được hiển thị dưới :

$$7423 = 7 \times 10^3 + 4 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$$

Hệ thống số thập phân có mười chữ số và các giá trị vị trí là lũy thừa nguyên của 10. Chúng ta nói rằng 10 là cơ số hoặc cơ sở của hệ thống số thập phân. Tương tự, vì hệ thống số nhị phân có hai chữ số nên cơ sở của nó là 2 và vì hệ thống số hex có 16 chữ số nên cơ sở của nó là 16.

Dưới đây là phần mô tả về các hệ thống số quan trọng như Binary, Decimal, Hexadecimal, và Octal trong phần nguồn của ứng dụng chuyển đổi :

a. Chuyển đổi giữa Binary and decimal

Cho số nhị phân (Binary) $b_n b_{n-1} b_{n-2} \dots b_2 b_1 b_0$, giá trị tương ứng là

$$b_n \times 2^n + b_{n-1} \times 2^{n-1} + b_{n-2} \times 2^{n-2} + \dots + b_2 \times 2^2 + b_1 \times 2^1 + b_0 \times 2^0.$$

Đây là một số ví dụ chuyển đổi binary thành decimals:

1. Binary : 10

$$\text{Công thức chuyển đổi : } 1 \times 2^1 + 0 \times 2^0$$

Decimal : 2

2. Binary : 10101011

$$\text{Công thức chuyển đổi : } 1 \times 2^7 + 0 \times 2^6 + \dots + 1 \times 2^1 + 0 \times 2^0$$

Decimal : 171

Chuyển Decimal thành binary tìm các bits $b_n b_{n-1} b_{n-2} \dots b_2 b_1 b_0$ như

$$d = b_n \times 2^n + b_{n-1} \times 2^{n-1} + b_{n-2} \times 2^{n-2} + \dots + b_2 \times 2^2 + b_1 \times 2^1 + b_0 \times 2^0$$

Ví dụ, Decimal number 123 là 1111011 trong binary. Chuyển đổi được thực hiện như sau :

$$\begin{array}{r} 2 \mid 123 \\ 2 \mid 30 \quad 1 \\ 2 \mid 15 \quad 0 \\ 2 \mid 7 \quad 1 \\ 2 \mid 3 \quad 1 \\ \mid 1 \quad 1 \end{array}$$

Hơn đếm số nhị phân từ dưới lên Lên. 1111011

b. Chuyển đổi giữa Hexadecimal và Decimal

Cho số hexadecimal $h_n h_{n-1} h_{n-2} \dots h_2 h_1 h_0$, giá trị tương ứng là

$$h_n \times 16^n + h_{n-1} \times 16^{n-1} + h_{n-2} \times 16^{n-2} + \dots + h_2 \times 16^2 + h_1 \times 16^1 + h_0 \times 16^0$$

Ví dụ, chuyển đổi hexadecimal thành decimals :

1. Hexadecimal : 7F

$$\text{Công thức chuyển đổi : } 7 \times 16^1 + 15 \times 16^0$$

Decimal : 127

2. Hexadecimal : FFFF

$$\text{Công thức chuyển đổi : } 15 \times 16^3 + 15 \times 16^2 + 15 \times 16^1 + 15 \times 16^0$$

Decimal : 65535

Chuyển đổi decimal d thành hexadecimal tìm các bits $h_n h_{n-1} h_{n-2} \dots h_2 h_1 h_0$ như

$$d = h_n \times 16^n + h_{n-1} \times 16^{n-1} + h_{n-2} \times 16^{n-2} + \dots + h_2 \times 16^2 + h_1 \times 16^1 + h_0 \times 16^0$$

Những các bit này có thể được tìm thấy bằng cách chia d liên tiếp cho 16 cho đến khi thương số bằng 0. Số dư là : $h_n h_{n-1} h_{n-2} \dots h_2 h_1 h_0$

Ví dụ, decimal number 123 là 7B trong hexadecimal. Chuyển đổi thực hiện như sau :

$$\begin{array}{r} 16 \mid 123 \\ 16 \mid 7 \quad 11 \\ \mid 0 \quad 7 \end{array}$$

Hơn đếm số thập lục phân từ dưới lên. 7B

c. Chuyển đổi giữa Binary và Hexadecimal numbers

Để chuyển đổi hexadecimal thành binary number, chỉ cần chuyển đổi từng chữ số trong số thập lục phân thành số nhị phân có bốn chữ số, sử dụng Bảng

Hexadecimal	Binary	Decimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

Ví dụ, hexadecimal number 7B is 1111011, trong đó 7 là 111 ở dạng nhị phân và B là 1011 ở dạng nhị phân.

Để chuyển đổi một số nhị phân thành số thập lục phân, hãy chuyển đổi bốn chữ số nhị phân từ phải sang trái trong số nhị phân thành số thập lục phân.

Ví dụ: số nhị phân 1110001101 là 38D, vì 1101 là D, 1000 là 8 và 11 là 3.

d. Chuyển Đổi Từ Octal to Decimal

Để chuyển đổi từ hệ thống số Octal sang hệ thống số Decimal, bạn có thể sử dụng công thức sau:

Cho một số Octal $h_n h_{n-1} h_{n-2} \dots h_2 h_1 h_0$, giá trị tương ứng trong hệ thống số Decimal là:

$$\text{Decimal} = h_0 \times 8^0 + h_1 \times 8^1 + h_2 \times 8^2 + \dots + h_n \times 8^n$$

Trong đó $h_0 h_1 h_2 \dots h_n$, là các chữ số Octal tại các vị trí tương ứng, bắt đầu từ phải sang trái.

Ví dụ:

Octal: 73

$$\text{Decimal} = 3 * 8^0 + 7 * 8^1 = 3 * 1 + 7 * 8 = 3 + 56 = 59$$

e. Chuyển Đổi Từ Decimal Sang Octal

Để chuyển đổi từ hệ thống số Decimal sang hệ thống số Octal, bạn có thể sử dụng thuật toán sau:

- Chia số Decimal (d) cho 8 và lưu kết quả vào một biến.
- Lấy phần dư của phép chia và ghi lại nó là một chữ số Octal.
- Lặp lại quá trình với kết quả của phép chia để tìm chữ số Octal tiếp theo.
- Lặp lại cho đến khi kết quả của phép chia bằng 0.
- Ghép các chữ số Octal lại theo thứ tự từ trái sang phải để có số Octal hoàn chỉnh.

Ví dụ:

Decimal: 59

Quá trình chuyển đổi:

$59 \div 8 = 7$, dư 3 (3 là chữ số Octal đầu tiên, 7 là phần nguyên).

$7 \div 8 = 0$, dư 7 (0 là kết quả của phép chia, 7 là phần nguyên thứ hai).

Kết quả: Octal = 73

2.1. Hệ Thống số Binary (Nhị phân)

Trong mã nguồn, hệ thống số nhị phân được thể hiện thông qua chuỗi ký tự chứa các ký tự '0' và '1'. Đây là một ví dụ về cách chuyển đổi từ hệ thống số nhị phân sang hệ thống số Decimal :

```
// Chuyển đổi từ Binary sang Decimal
int decimalIntegerPart = Integer.parseInt(binaryIntegerPart, 2);
// Chuyển đổi phần thập phân (nếu có) từ Binary sang Decimal
double decimalFractionalPart = 0.0;
for (int i = 0; i < binaryFractionalPart.length(); i++) {
    char binaryDigit = binaryFractionalPart.charAt(i);
    int digitValue = Character.getNumericValue(binaryDigit);
    decimalFractionalPart += digitValue / Math.pow(2, i + 1);
}
```

2.2 Hệ Thống số Decimal (Thập phân)

Hệ thống số thập phân là hệ thống số phổ biến và quen thuộc trong lập trình. Trong mã nguồn, hệ thống số Decimal được xử lý dưới dạng số nguyên và số thực. Dưới đây là một ví dụ về cách chuyển đổi từ hệ thống số Decimal sang các hệ thống số khác:

```
// Chuyển đổi từ Decimal sang Binary
String binaryValue = Integer.toBinaryString(decimalValue);

// Chuyển đổi từ Decimal sang Hexadecimal
String hexadecimalValue = Integer.toHexString(decimalValue).toUpperCase();

// Chuyển đổi từ Decimal sang Octal
String octalValue = Integer.toOctalString(decimalValue);
```

2.3 Hệ Thống số Hexadecimal (Thập lục phân)

Hệ thống số thập lục phân sử dụng các ký tự từ '0' đến '9' và 'A' đến 'F' để biểu diễn các giá trị từ 0 đến 15. Trong mã nguồn, hệ thống số Hexadecimal được chuyển đổi tương tự như Decimal, nhưng sử dụng các hàm khác nhau:

```
// Chuyển đổi từ Hexadecimal sang Decimal
int decimalValue = Integer.parseInt(hexValue, 16);

// Chuyển đổi từ Hexadecimal sang Binary
String binaryValue = new BigInteger(hexValue, 16).toString(2);

// Chuyển đổi từ Hexadecimal sang Octal
String octalValue = new BigInteger(hexValue, 16).toString(8);
```

2.4. Hệ Thống số Octal (Bát phân)

Hệ thống số Octal sử dụng các ký tự từ '0' đến '7' để biểu diễn các giá trị từ 0 đến 7. Dưới đây là một ví dụ về cách chuyển đổi từ hệ thống số Oct sang Decimal :

```
// Chuyển đổi từ Oct sang Decimal
int decimalValue = Integer.parseInt(octalValue, 8);

// Chuyển đổi từ Oct sang Binary
String binaryValue = new BigInteger(octalValue, 8).toString(2);

// Chuyển đổi từ Octal sang Hexadecimal
String hexadecimalValue = new BigInteger(octalValue, 8).toString(16).toUpperCase();
```

Những đoạn mã này được sử dụng trong ứng dụng để thực hiện các phép chuyển đổi giữa các hệ thống số khác nhau. Điều này giúp người dùng dễ nhập dữ liệu trong một hệ thống số và xem kết quả tương ứng trong các hệ thống số khác.

2.5 Ưu điểm của các hệ thống số

- **Binary (Nhị phân):** Được sử dụng rộng rãi trong lĩnh vực máy tính và lập trình như biểu diễn dữ liệu bộ nhớ và thao tác logic.
- **Decimal (Thập phân):** Là hệ thống số phổ biến trong cuộc sống hàng ngày và dễ hiểu, thích hợp cho các tính toán thông thường.
- **Hexadecimal (Thập lục phân):** Thường được sử dụng để biểu diễn các giá trị máy tính như màu sắc, địa chỉ bộ nhớ và trong lập trình nhúng.
- **Octal (Bát phân):** Ít phổ biến hơn những hệ thống số khác, nhưng vẫn còn được sử dụng trong một số ngữ cảnh đặc biệt.

2.6 Ứng dụng của các hệ thống số

- **Binary:** Sử dụng rộng rãi trong lập trình máy tính, biểu diễn dữ liệu bộ nhớ, mã hóa và thao tác logic.
- **Decimal:** Sử dụng trong cuộc sống hàng ngày, toán học, tài chính, và hầu hết các môi trường tính toán thông thường.
- **Hexadecimal:** Thường được sử dụng trong lập trình máy tính (ví dụ: mã màu HTML, địa chỉ bộ nhớ), cơ học lượng tử, và điện tử.
- **Octal:** Ít phổ biến hơn những hệ thống số khác và thường không được sử dụng trong lập trình hiện đại.

Chương 3 : Thuật Toán Trong Ứng dụng Chuyển Đổi Hệ Số Số Trong Java

3. Thuật Toán

Trong ứng dụng, thuật toán chính đó là thuật toán chuyển đổi giữa các hệ số cơ số: Decimal (hệ thập phân) sang các hệ số khác như Binary (nhị phân), Octal (bát phân), và Hexadecimal (thập lục phân), và ngược lại. Dưới đây là một số điểm quan trọng về thuật toán:

3.1 Thuật Toán Chuyển Đổi Từ Hệ 10 Sang Hệ Khác

Trong phần này, chúng ta sẽ tìm hiểu về các thuật toán quan trọng đã được sử dụng trong ứng dụng để chuyển đổi giữa các hệ số cơ sở như Binary, Octal, Decimal và Hexadecimal.

3.1.1 Chuyển Đổi Từ Decimal Sang Binary

Thuật toán chuyển đổi từ hệ thập phân (Decimal) sang hệ nhị phân (Binary) được thực hiện như sau:

- **Lấy Phần Nguyên và Thập Phân:** Đầu tiên, ứng dụng tách số hệ thập phân thành hai phần: phần nguyên và phần thập phân.
 - **Chuyển Đổi Phần Nguyên:** Phần nguyên của số hệ thập phân được chuyển đổi sang hệ nhị phân bằng cách lấy phần dư của phần nguyên khi chia liên tục cho 2 và lưu trữ các phần dư này theo thứ tự đảo ngược.
 - **Chuyển Đổi Phần Thập Phân:** Phần thập phân của số hệ thập phân được chuyển đổi sang hệ nhị phân bằng cách nhân liên tục với 2 và lấy phần nguyên của kết quả. Kết quả được lưu trữ theo thứ tự xuôi.
- Kết Hợp Kết Quả: Kết quả của cả hai phần được kết hợp để tạo số nhị phân hoàn chỉnh.

Thuật toán này chạy trong thời gian $O(\log_{10}(n))$, với n là số hệ thập phân.

Ví dụ:

Giả sử chúng ta có số Decimal là 27. Chúng ta có thể thực hiện chuyển đổi như sau:

- Lấy phần nguyên: $27 / 2 = 13$, phần dư 1.
- Tiếp tục: $13 / 2 = 6$, phần dư 1.
- Tiếp tục: $6 / 2 = 3$, phần dư 0.
- Cuối cùng: $3 / 2 = 1$, phần dư 1.

Kết quả cuối cùng là "11011," là biểu diễn nhị phân của 27.

3.1.2 Chuyển Đổi Từ Decimal Sang Octal và Hexadecimal

Quy trình chuyển đổi từ Decimal sang Octal and Hexadecimal tương tự như chuyển đổi sang Binary, với sự khác biệt ở hệ cơ số sử dụng (8 cho Octal và 16 cho Hexadecimal).

Ví dụ :

- Số Decimal: 134
- Chuyển đổi sang Octal:
- Chia lần lượt: $134 / 8 = 16$, phần dư 6; $16 / 8 = 2$, phần dư 0; $2 / 8 = 0$, phần dư 2.

Kết quả: "206" là biểu diễn Octal của số Decimal 134.

3.2 Thuật Toán Chuyển Đổi Từ Hệ Khác Sang Decimal

Ứng dụng cũng hỗ trợ chuyển đổi từ các hệ số khác (Binary, Octal và Hexadecimal) sang hệ thập phân (Decimal). Quy trình chuyển đổi này đơn giản hơn:

Chia Số Theo Hệ Cơ Số: Số ở hệ số khác được chia theo hệ cơ số tương ứng để lấy giá trị từng chữ số.

Tính Tổng: Các giá trị sau khi chia được tổng hợp lại để tạo giá trị hệ thập phân.

Thuật toán này chạy trong thời gian $O(n)$, với n là số chữ số trong số đầu vào.

Ví dụ:

Giả sử chúng ta có số Binary là "10101." Để chuyển đổi sang Decimal, chúng ta thực hiện như sau:

$$1 * 2^4 + 0 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0 = 16 + 0 + 4 + 0 + 1 = 21.$$

Kết quả cuối cùng là số Decimal 21.

3.3 Xử Lý Số Thực Và Số Âm

Ví dụ 1:

- Số Decimal thập phân: -2.25
- Chuyển đổi:
- Phần nguyên: -2 (chuyển đổi như bình thường, lưu ý dấu trừ).
- Phần thập phân: 0.25
- Nhân 0.25 với 2: $0.25 * 2 = 0.5$. Lấy phần nguyên, kết quả là 0.
- Tiếp tục: $0.5 * 2 = 1$. Lấy phần nguyên, kết quả là 1.
- Kết quả: "0.01" là biểu diễn nhị phân của phần thập phân.
- Kết quả cuối cùng: "-10.01" là biểu diễn nhị phân của số thực -2.25.

Ví dụ 2:

- Để chuyển đổi số thực, hãy xem xét số Decimal thập phân 13.625. Chúng ta có thể chuyển đổi phần nguyên bằng cách sử dụng thuật toán như trước. Sau đó, chúng ta có thể xử lý phần thập phân như sau:
- Nhân phần thập phân 0.625 với 2: $0.625 * 2 = 1.25$. Lấy phần nguyên, kết quả là 1.
- Tiếp tục: $0.25 * 2 = 0.5$. Lấy phần nguyên, kết quả là 0.
- Tiếp tục: $0.5 * 2 = 1$. Lấy phần nguyên, kết quả là 1.
- Kết quả cuối cùng là "1011.101" là biểu diễn nhị phân của số thực 13.625.

3.3.1 Chuyển Đổi Số Thực

Để hỗ trợ số thực, ứng dụng sử dụng một phần riêng của thuật toán chuyển đổi Decimal sang các hệ khác sau khi đã xử lý phần nguyên. Phần này xử lý phần thập phân của số thực bằng cách nhân liên tục với cơ số của hệ đó và lưu các số nguyên kết quả.

3.3.2 Xử Lý Số Âm

Để hỗ trợ số âm, ứng dụng sử dụng một biến cờ (flag) để xác định dấu của số và sau đó loại bỏ dấu trước khi thực hiện chuyển đổi. Kết quả cuối cùng sẽ có dấu trả lại theo yêu cầu.

3.4 Tối Ưu Hóa Và Mở Rộng

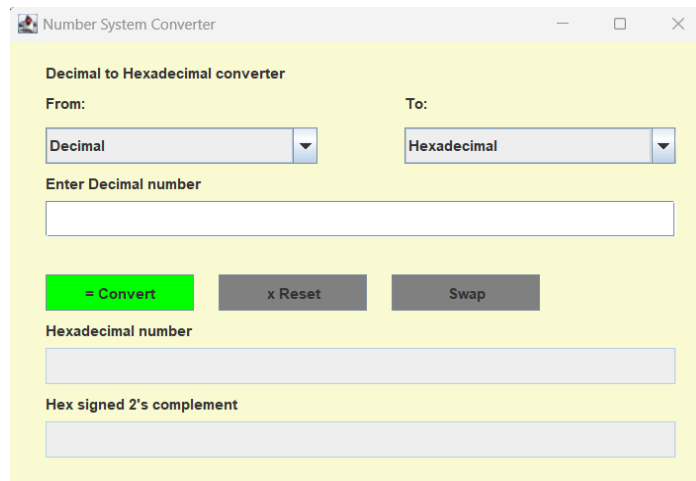
Ứng dụng có tiềm năng được mở rộng bằng cách thêm tính năng hỗ trợ chuyển đổi giữa các hệ thống số khác như hệ bát phân (Base-8). Để thực hiện điều này, chúng ta có thể sử dụng thuật toán tương tự như chuyển đổi từ Decimal và chỉ cần cập nhật giao diện người dùng để hỗ trợ hệ cơ số mới.

Chương 4 : Giao Diện Người Dùng

4. Giao Diện Người Dùng

4.1 Mô Tả Giao Diện Người Dùng

Trong chương này, chúng ta sẽ mô tả giao diện người dùng của ứng dụng chuyển đổi hệ thống số. Giao diện người dùng thiết kế với một giao diện đơn giản và thân thiện để hỗ trợ chuyển đổi giữa các hệ thống số khác nhau. Hãy xem xét hình dưới đây để hiểu cách giao diện được tổ chức :

The image shows a window titled "Number System Converter". Inside the window, there is a section titled "Decimal to Hexadecimal converter". Below this title, there are two dropdown menus: "From:" and "To:". The "From:" dropdown is currently set to "Decimal" and the "To:" dropdown is set to "Hexadecimal". Below these dropdowns is a text input field labeled "Enter Decimal number". Underneath the input field are three buttons: a green button labeled "= Convert", a grey button labeled "x Reset", and a grey button labeled "Swap". Below the buttons, there are two more text input fields: one labeled "Hexadecimal number" and another labeled "Hex signed 2's complement".

- Ô Nhập Liệu (Input Field) : Đây là nơi người dùng nhập giá trị số cần chuyển đổi. Đối với chuyển đổi từ hệ thống số nào sang hệ thống số khác, người dùng sẽ nhập giá trị vào ô này.
- Nút Chuyển Đổi(Convert button) : Sau khi nhập giá trị, người dùng có thể vào nút “Chuyển Đổi” để thực hiện phép chuyển đổi. Khi nhấn vào nút này, ứng dụng sẽ tiến hành tính toán và hiển thị kết quả.
- Nhút Đặt Lại (Reset Button) : Nút “Đặt Lại” cho phép người dùng xóa giá trị nhập vào ô nhập liệu và đặt lại kết quả chuyển đổi.
- Lựa Chọn Hệ Thống số (Number System Selection) : Ứng dụng cho phép người dùng chọn hệ thống số nguồn và hệ thống đích trong các menu thả xuống. Khi chuyển đổi, người dùng sẽ chọn hệ thống số ban đầu và hệ thống mục tiêu.
- Các Combobox (Dropdown Lists) : Ứng dụng có hai Combobox để người dùng chọn hệ thống số nguồn và hệ thống số đích. Người dùng có thể chọn từ các tùy chọn như “Binary”, “Decimal”, “Octal”, và “Hexadecimal”.

- Kết Quả Chuyển Đổi (Conversion Result) : Sau khi nhấn nút “Chuyển Đổi”, kết quả của phép chuyển đổi sẽ được hiển thị ở đây.

4.2 Các Chức Năng Của Giao Diện

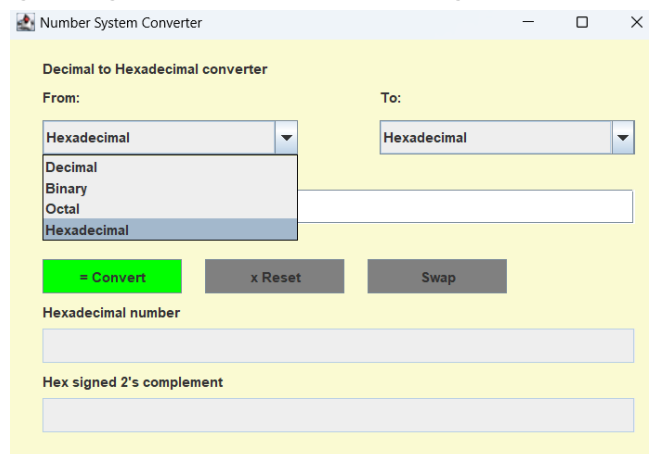
Giao Diện người dùng hỗ trợ các chức năng sau:

- Chuyển đổi giá trị giữa các hệ thống số khác nhau(Binary, Decimal, Hexadecimal, Octal).
- Xử lý các số nguyên và số thực
- Hiển thị kết quả chuyển đổi cho người dùng

4.3 Hướng Dẫn Sử Dụng

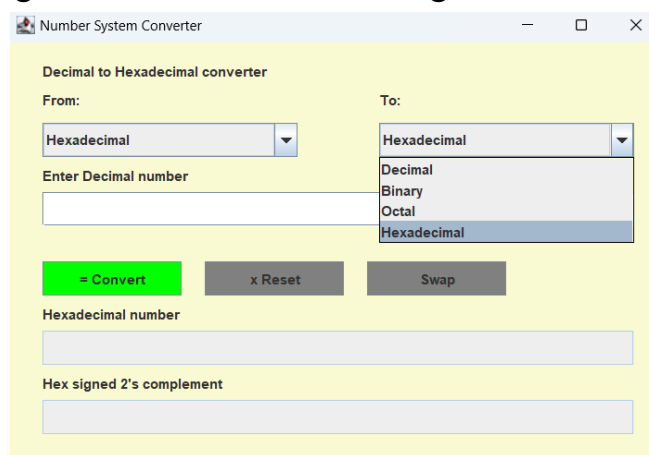
Để thực hiện chuyển đổi giữa các hệ thống số, người dùng cần làm theo các bước sau:

- Chọn hệ thống số nguồn từ menu thả xuống Từ “From”



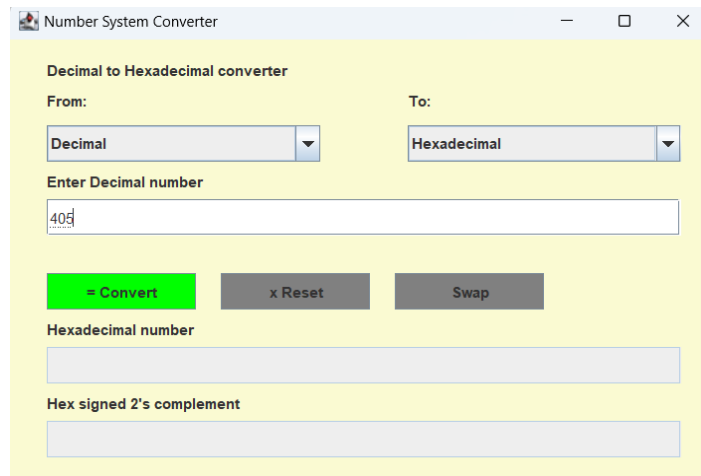
The screenshot shows the 'Number System Converter' window. The 'From:' dropdown menu is open, displaying a list of options: Hexadecimal, Decimal, Binary, Octal, and Hexadecimal. The 'To:' dropdown menu is also open, showing the same list. The 'Convert' button is highlighted in green. Below the dropdowns, there are input fields for 'Hexadecimal number' and 'Hex signed 2's complement'.

- Chọn hệ thống số đích từ Combobox Sang “To”



The screenshot shows the 'Number System Converter' window. The 'To:' dropdown menu is open, displaying a list of options: Hexadecimal, Decimal, Binary, Octal, and Hexadecimal. The 'From:' dropdown menu is also open, showing the same list. The 'Convert' button is highlighted in green. Below the dropdowns, there are input fields for 'Hexadecimal number' and 'Hex signed 2's complement'.

➤ Nhập giá trị cần chuyển đổi vào ô nhập liệu



Number System Converter

Decimal to Hexadecimal converter

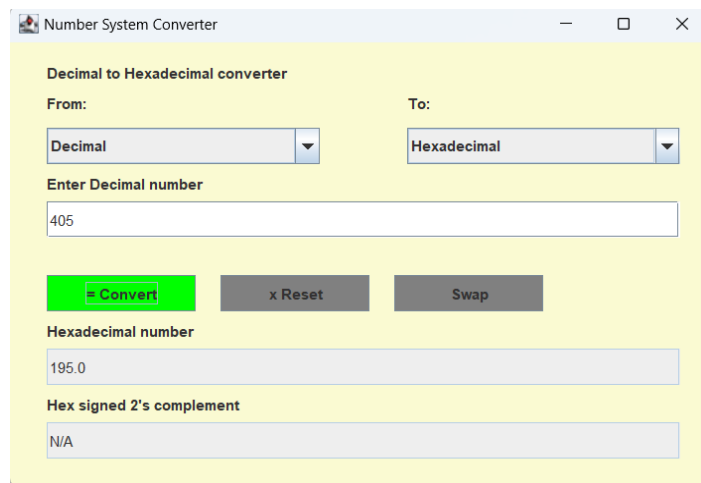
From: To:

Enter Decimal number

Hexadecimal number

Hex signed 2's complement

➤ Nhấn nút “Chuyển Đổi” (Convert) để thị kết quả



Number System Converter

Decimal to Hexadecimal converter

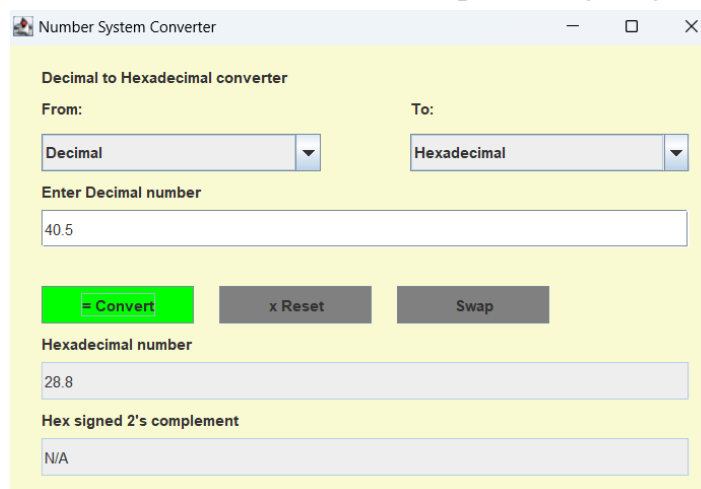
From: To:

Enter Decimal number

Hexadecimal number

Hex signed 2's complement

➤ Kết quả chuyển đổi sẽ hiển thị ở các ô kết quả tương ứng.



Number System Converter

Decimal to Hexadecimal converter

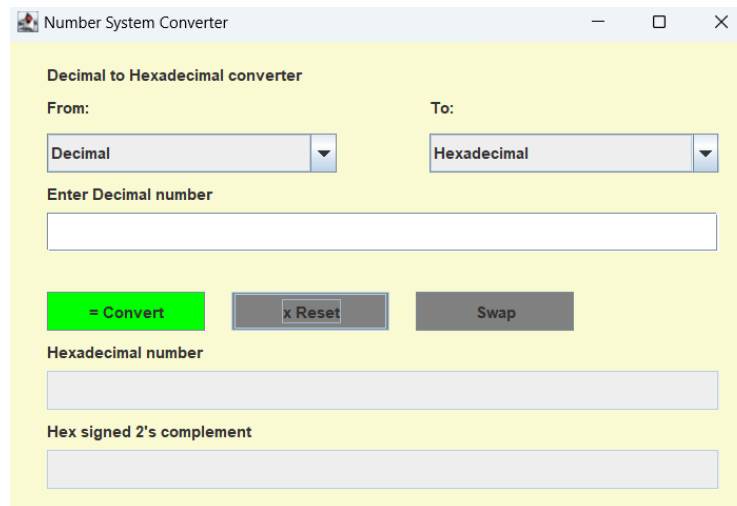
From: To:

Enter Decimal number

Hexadecimal number

Hex signed 2's complement

- Nếu muốn xóa giá trị và kết quả chuyển đổi, nhấn nút “Đặt Lại (Reset)”.



4.4 Mục Tiêu Thiết Kế Giao Diện

Giao diện được thiết kế với mục tiêu đơn giản, dễ sử dụng và hiệu quả. Người dùng có thể thực hiện các phép chuyển đổi một cách dễ dàng mà không gặp khó khăn.

4.5 Thách Thức Trong Thiết Kế Giao Diện

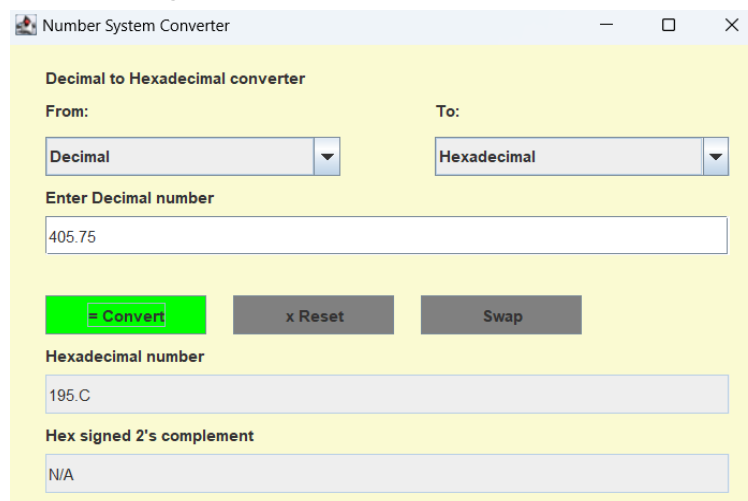
Trong quá trình thiết kế giao diện, chúng ta đã đối mặt số thách thức, bao gồm :

- Xử lý các số thực và số nguyên đồng thời trang trong một giao diện đơn giản
- Đảm bảo tính thẩm mỹ và sự hiệu quả trong trải nghiệm người dùng

4.6 Hình Ảnh Minh Họa

Đây là một số hình ảnh minh họa cho giao diện người dùng của ứng dụng :

- Chuyển từ Decimal sang Hexadecimal



➤ Chuyển từ Hexadecimal sang Decimal

Number System Converter

Hexadecimal to Decimal converter

From: Hexadecimal To: Decimal

Enter Hexadecimal number

405.75

Convert x Reset Swap

Decimal number

1029.4570312500

Decimal from signed 2's complement

N/A

➤ Chuyển từ Binary sang hexadecimal

Number System Converter

Decimal to Hexadecimal converter

From: Binary To: Hexadecimal

Enter Decimal number

1010

Convert x Reset Swap

Hexadecimal number

A

Hex signed 2's complement

10.000

➤ Chuyển từ Octal sang Hexadecimal

Number System Converter

Decimal to Hexadecimal converter

From: Octal To: Hexadecimal

Enter Decimal number

2023

Convert x Reset Swap

Hexadecimal number

413

Hex signed 2's complement

1043

ASCII Converter

Chương 1: Mô Tả Tổng Quan Đề Tài

1.1 Mô Tả Bài Toán

Đề tài này liên quan đến việc phát triển một ứng dụng giao diện đồ họa (GUI) trong Java để chuyển đổi văn bản ASCII thành các dạng khác nhau bao gồm Hex, Binary, Decimal và tính toán checksum. Dưới đây là mô tả chi tiết về bài toán:

- Mục tiêu: Phát triển một ứng dụng GUI Java cho phép người dùng nhập hoặc mở tệp văn bản chứa ký tự ASCII và sau đó chuyển đổi nó thành các định dạng khác nhau, bao gồm Hex (hệ số 16), Binary (hệ số 2), Decimal (hệ số 10) và tính toán checksum.
- Giao diện người dùng: Ứng dụng có một giao diện người dùng đơn giản với các phần sau:
 - Khung văn bản cho người dùng nhập hoặc hiển thị văn bản ASCII.
 - Các ô văn bản khác để hiển thị kết quả chuyển đổi Hex, Binary, Decimal và checksum.
 - Các tùy chọn cho người dùng như chọn dấu phân cách cho số, bật/tắt tiền tố "0x"/"0b" cho Hex và Binary, và chọn kích thước checksum.
- Chức năng:
 - Người dùng có thể nhập văn bản ASCII bằng cách gõ hoặc mở tệp văn bản.
 - Ứng dụng tự động chuyển đổi văn bản ASCII thành Hex, Binary và Decimal và hiển thị chúng trên giao diện.
 - Người dùng có thể chọn kích thước checksum và loại tính toán (Sum, 2's Complement, Xor), sau đó ứng dụng tính toán checksum và hiển thị kết quả.
 - Có chức năng đặt lại tất cả các trường để người dùng có thể bắt đầu lại từ đầu.

1.2 Công Nghệ Sử Dụng

Để phát triển ứng dụng này, người lập trình đã sử dụng công nghệ và thư viện sau đây:

- Ngôn ngữ lập trình: Java
- Thư viện GUI: Java Swing
- Công cụ phát triển: IDE Java (ví dụ: Eclipse, IntelliJ IDEA)
- Thư viện xử lý tệp và đọc ghi dữ liệu: Java File I/O

1.3 Các Yêu cầu

Các yêu cầu chức năng và phi chức năng chính của ứng dụng đã được mô tả trong đề tài:

➤ Yêu cầu Chức Năng:

- Cho phép người dùng nhập văn bản ASCII hoặc mở một tệp văn bản để chuyển đổi.
- Tự động chuyển đổi văn bản ASCII thành các định dạng Hex, Binary và Decimal và hiển thị chúng trên giao diện.
- Cho phép người dùng chọn kích thước checksum (8-bit, 16-bit, 32-bit).
- Tính toán checksum dựa trên văn bản ASCII và hiển thị kết quả.
- Cho phép người dùng đặt dấu phân cách cho số (None, Space, Comma, User defined).
- Cho phép người dùng bật/tắt tiền tố "0x" cho Hex và "0b" cho Binary.
- Có nút đặt lại để đặt lại tất cả các trường về giá trị mặc định.

➤ Yêu cầu Phi Chức Năng:

- Ứng dụng có giao diện người dùng đẹp và dễ sử dụng.
- Xử lý lỗi và thông báo lỗi cho người dùng khi có sự cố trong việc mở tệp hoặc tính toán checksum.
- Ứng dụng cần được triển khai dưới dạng tệp thực thi có thể chạy độc lập trên nhiều hệ điều hành.

Chương 2: Thiết Kế và Cài Đặt Giao Diện

Chương này trình bày về quá trình thiết kế và cài đặt giao diện của ứng dụng ASCII Converter. Giao diện người dùng là một phần quan trọng của ứng dụng, vì vậy nó cần được thiết kế sao cho dễ sử dụng và thân thiện với người dùng.

2.1 Thiết Kế Giao Diện

Giao diện của ứng dụng ASCII Converter được thiết kế để có các phần sau:

- Khung văn bản ASCII: Đây là nơi người dùng có thể nhập hoặc hiển thị văn bản ASCII.
- Các ô văn bản kết quả: Có bốn ô văn bản riêng biệt để hiển thị kết quả chuyển đổi văn bản thành Hex, Binary, Decimal và checksum.
- Tùy chọn: Người dùng có thể chọn dấu phân cách cho số, bật/tắt tiền tố "0x"/"0b" cho Hex và Binary, chọn kích thước checksum, và đặt lại tất cả các trường.
- Nút mở tệp và nút đặt lại: Có một nút để mở tệp văn bản và một nút để đặt lại tất cả các trường về giá trị mặc định.

2.2 Cài Đặt Giao Diện

Giao diện người dùng của ứng dụng ASCII Converter được cài đặt bằng sử dụng thư viện Java Swing. Dưới đây là mô tả chi tiết về cài đặt các thành phần giao diện chính:

- Khung văn bản ASCII: Sử dụng JTextArea để tạo khung văn bản cho văn bản ASCII. Đã thiết lập thuộc tính setLineWrap(true) để cho phép dòng tự động xuống khi vượt quá chiều rộng và đặt JScrollPane cho khung văn bản để cho phép cuộn dọc và ngang khi cần thiết.
- Các ô văn bản kết quả: Có bốn JTextArea tương ứng với Hex, Binary, Decimal và checksum. Các ô văn bản này cũng được thiết lập để tự động xuống dòng và có JScrollPane để cuộn khi cần.
- Tùy chọn: Sử dụng các thành phần như JComboBox để cho phép người dùng chọn dấu phân cách cho số và chọn kích thước checksum. Sử dụng JCheckBox để bật/tắt tiền tố "0x"/"0b" cho Hex và Binary. Đã cài đặt các ActionListener để theo dõi sự thay đổi trong các tùy chọn này và cập nhật kết quả diễn giải tương ứng.

- Nút mở tệp và nút đặt lại: Sử dụng JButton để tạo nút "Open File" và nút "Reset" để mở tệp và đặt lại tất cả các trường. Đã thiết lập ActionListener cho cả hai nút để xử lý sự kiện khi người dùng nhấn vào chúng.

2.3 Giao Diện Cuối Cùng

Giao diện người dùng cuối cùng của ứng dụng ASCII Converter có vẻ như sau:

The screenshot shows a Java Swing window titled "ASCII, Hex, Binary, Decimal, Base64 converter". The window has a yellow background and contains the following elements:

- At the top, there are two buttons: "Open File" (green) and "x Reset" (grey).
- Below the buttons is a "Number delimiter" section with a dropdown menu set to "None" and an adjacent empty text field.
- Underneath is a checkbox labeled "0x/0b prefix" which is currently unchecked.
- There are four large text input fields stacked vertically, each with a label to its left: "ASCII text:", "Hex (bytes):", "Binary (bytes):", and "Decimal (bytes):".
- At the bottom, there is a "Checksum" section with two dropdown menus. The first is set to "8-bit" and the second is set to "Sum". To the right of these is a text field containing "00".
- Below the checksum section is another "x Reset" button.

Giao diện này cho phép người dùng dễ dàng nhập hoặc mở tệp văn bản ASCII, chuyển đổi nó thành các định dạng khác nhau và tính toán checksum. Các tùy chọn cho phép người dùng tùy chỉnh cách chuyển đổi và tính toán được thực hiện.

Chương 3: Triển Khai và Kiểm Thử

Chương này tập trung vào quá trình triển khai ứng dụng ASCII Converter và thực hiện kiểm thử để đảm bảo tính ổn định và chức năng của ứng dụng.

3.1 Triển Khai Ứng Dụng

Trước khi triển khai, cần biên dịch mã nguồn của ứng dụng thành tệp thực thi (.jar hoặc .exe, tùy thuộc vào nền tảng) để có thể chạy độc lập trên các máy tính khác nhau. Sau đó, ứng dụng có thể triển khai thông qua các phương tiện sau:

- **Triển khai cục bộ:** Ứng dụng có thể chạy trên máy tính cá nhân hoặc máy chủ cục bộ. Người dùng chỉ cần thực hiện cài đặt bằng cách chạy tệp thực thi đã biên dịch.
- **Triển khai trực tuyến:** Ứng dụng có thể được đưa lên một máy chủ web và truy cập thông qua trình duyệt web. Điều này đòi hỏi quá trình triển khai máy chủ web và cấu hình phù hợp.

3.2 Kiểm Thử

Kiểm thử là một phần quan trọng để đảm bảo tính ổn định và chức năng của ứng dụng. Dưới đây là các loại kiểm thử quan trọng mà ứng dụng ASCII Converter cần trải qua:

- **Kiểm thử tích hợp:** Kiểm tra xem các thành phần của ứng dụng có làm việc cùng nhau một cách đúng đắn hay không. Ví dụ, kiểm tra xem việc chuyển đổi văn bản ASCII thành các định dạng khác có chính xác không.
- **Kiểm thử đơn vị:** Kiểm tra từng phần riêng lẻ của mã nguồn để đảm bảo chúng hoạt động đúng đắn. Ví dụ, kiểm tra chức năng chuyển đổi ASCII sang Hex.
- **Kiểm thử chấp nhận:** Thực hiện kiểm thử thực tế với dữ liệu thực tế để xác định xem ứng dụng hoạt động chính xác trong các tình huống thực tế. Ví dụ, kiểm tra việc nhập và chuyển đổi văn bản ASCII từ tệp văn bản.

- **Kiểm thử hiệu suất:** Đảm bảo rằng ứng dụng có thể xử lý lượng dữ liệu lớn mà không gây treo hoặc giảm hiệu suất đáng kể.
- **Kiểm thử bảo mật:** Đảm bảo rằng ứng dụng không có lỗ hổng bảo mật và thông tin của người dùng được bảo vệ đúng cách.
- **Kiểm thử tương thích:** Kiểm tra ứng dụng trên nhiều hệ điều hành và trình duyệt web để đảm bảo tính tương thích.
- **Kiểm thử giao diện người dùng:** Đảm bảo rằng giao diện người dùng có vẻ đẹp và dễ sử dụng trên mọi nền tảng.

3.3 Ghi Chú Về Quản Lý Phiên Bản

Trong quá trình triển khai và kiểm thử, quản lý phiên bản (version control) là một phần quan trọng để theo dõi và quản lý các phiên bản của ứng dụng. Sử dụng các hệ thống quản lý phiên bản như Git để theo dõi các thay đổi trong mã nguồn và quản lý phiên bản ứng dụng.

Chương 4: Thuật Toán và Cách Tính Toán

Chương này sẽ trình bày chi tiết về thuật toán được sử dụng trong ứng dụng ASCII Converter và cách tính toán các giá trị liên quan đến chuyển đổi ASCII và checksum.

4.1 Thuật Toán Chuyển Đổi ASCII

Thuật toán chuyển đổi ASCII trong ứng dụng ASCII Converter là quá trình chuyển đổi văn bản ASCII thành các định dạng khác như Hexadecimal, Binary và Decimal. Dưới đây là mô tả cụ thể của thuật toán này:

- **Chuyển đổi ASCII thành Hexadecimal:** Đối với mỗi ký tự trong văn bản ASCII, thuật toán chuyển đổi ký tự đó thành giá trị thập lục phân (hexadecimal) tương ứng và nối chúng lại để tạo chuỗi kết quả. Ký tự thập lục phân là một con số trong khoảng từ '0' đến '9' hoặc từ 'A' đến 'F'. Ví dụ: Ký tự 'A' trong ASCII tương ứng với giá trị thập lục phân là "41".
- **Chuyển đổi ASCII thành Binary:** Tương tự, thuật toán chuyển đổi mỗi ký tự trong văn bản ASCII thành biểu diễn nhị phân (binary) tương ứng và nối chúng lại để tạo chuỗi kết quả. Biểu diễn nhị phân bao gồm một chuỗi các số 0 và 1, và mỗi ký tự ASCII được chuyển đổi thành 8 bits. Ví dụ: Ký tự 'A' trong ASCII tương ứng với biểu diễn nhị phân "01000001".
- **Chuyển đổi ASCII thành Decimal:** Thuật toán chuyển đổi mỗi ký tự trong văn bản ASCII thành giá trị thập phân (decimal) tương ứng và nối chúng lại để tạo chuỗi kết quả. Giá trị thập phân là một số nguyên trong khoảng từ 0 đến 255, và mỗi ký tự ASCII tương ứng với một giá trị thập phân. Ví dụ: Ký tự 'A' trong ASCII tương ứng với giá trị thập phân "65".

4.2 Tính Toán Checksum

Ứng dụng ASCII Converter cũng cung cấp tính năng tính toán checksum của dữ liệu ASCII. Người dùng có thể chọn kích thước checksum (8-bit, 16-bit, hoặc 32-bit) và cách tính checksum (Sum, 2's Complement, hoặc Xor). Dưới đây là mô tả cách tính toán checksum:

- **Checksum Sum (Tổng):** Thuật toán tính tổng các giá trị ASCII tương ứng với các ký tự trong văn bản và sau đó lấy phần dư sau khi chia tổng cho 2^n (n là kích thước checksum được chọn). Kết quả checksum là một số nguyên trong khoảng từ 0 đến $2^n - 1$.
- **Checksum 2's Complement (Bù 2):** Thuật toán này tính tổng các giá trị ASCII tương ứng với các ký tự trong văn bản, sau đó áp dụng phép bù 2 (complement) cho tổng để tạo checksum. Kết quả checksum cũng là một số nguyên trong khoảng từ 0 đến $2^n - 1$.
- **Checksum Xor (XOR):** Thuật toán Xor tính toán XOR (hoặc phép XOR) của các giá trị ASCII tương ứng với các ký tự trong văn bản. Kết quả checksum là một số nguyên trong khoảng từ 0 đến $2^n - 1$.

Chương 5 : Giao Diện Người Dùng (User Interface)

Trong chương này, chúng tôi sẽ trình bày về giao diện người dùng của ứng dụng ASCII Converter, bao gồm mô tả, chức năng, hướng dẫn sử dụng, mục tiêu thiết kế, thách thức trong thiết kế, và hình ảnh minh họa.

5.1 Mô Tả Giao Diện Người Dùng

Giao diện người dùng của ASCII Converter được thiết kế để cung cấp một cách thuận tiện và trực quan để chuyển đổi dữ liệu ASCII và tính toán checksum. Dưới đây là mô tả chi tiết về giao diện người dùng:

Number Delimiter (Ký hiệu số): Đây là một hộp thả xuống cho phép người dùng chọn ký hiệu số được sử dụng khi chuyển đổi dữ liệu ASCII thành các định dạng khác như Hexadecimal, Binary và Decimal. Người dùng có thể lựa chọn giữa các tùy chọn sau:

- "None" (Không có ký hiệu): Không thêm ký hiệu nào sau mỗi số.
- "Space" (Dấu cách): Sử dụng dấu cách làm ký hiệu số.
- "Comma" (Dấu phẩy): Sử dụng dấu phẩy làm ký hiệu số.
- "User defined" (Người dùng tự định nghĩa): Cho phép người dùng tự định nghĩa một ký hiệu số.

0x/0b Prefix Checkbox (Hộp kiểm tiền tố 0x/0b): Đây là một hộp kiểm cho phép người dùng chọn có muốn thêm tiền tố "0x" hoặc "0b" vào số Hexadecimal và Binary khi chuyển đổi. Nếu được chọn, tiền tố sẽ được thêm vào kết quả.

ASCII Text (Văn bản ASCII): Khung văn bản này là nơi người dùng nhập hoặc dán dữ liệu ASCII mà họ muốn chuyển đổi hoặc tính toán checksum.

Hex (bytes), Binary (bytes), Decimal (bytes) (Hexadecimal, Nhị phân, Thập phân): Đây là các khung văn bản hiển thị kết quả chuyển đổi của dữ liệu ASCII thành các định dạng tương ứng. Kết quả sẽ tự động cập nhật khi người dùng nhập hoặc thay đổi dữ liệu ASCII.

Checksum (Kiểm tra tổng): Đây là một hộp thả xuống cho phép người dùng chọn kích thước checksum (8-bit, 16-bit, hoặc 32-bit) cho tính toán checksum.

Checksum Result (Kết quả kiểm tra tổng): Khung văn bản này hiển thị kết quả của tính toán checksum dựa trên dữ liệu ASCII đã nhập. Kết quả sẽ tự động cập nhật khi người dùng chọn kích thước checksum và cách tính checksum.

Open File Button (Nút Mở tệp): Nút này cho phép người dùng mở một tệp văn bản chứa dữ liệu ASCII để nạp vào ứng dụng. Dữ liệu từ tệp sẽ được hiển thị trong khung "ASCII Text."

Reset Buttons (Nút Đặt lại): Có hai nút "x Reset" tương ứng với các phần của giao diện: một nút dưới Number Delimiter và một nút dưới Checksum. Nhấp vào nút "x Reset" sẽ đặt lại tất cả các khung văn bản và tùy chọn về trạng thái ban đầu.

5.2 Các Chức Năng Của Giao Diện

Giao diện người dùng của ASCII Converter cung cấp các chức năng sau:

- Chuyển đổi dữ liệu ASCII thành các định dạng khác như Hexadecimal, Binary và Decimal.
- Tính toán checksum của dữ liệu ASCII dựa trên kích thước và cách tính checksum được chọn.
- Mở tệp văn bản để nạp dữ liệu ASCII.
- Đặt lại giao diện về trạng thái ban đầu để bắt đầu một chuyển đổi hoặc tính toán checksum mới.

5.3 Mục Tiêu Thiết Kế Giao Diện

Mục tiêu thiết kế giao diện của ASCII Converter bao gồm:

- Đảm bảo tính trực quan và dễ sử dụng để người dùng có thể dễ dàng thao tác với ứng dụng.
- Hiển thị kết quả chuyển đổi và tính toán checksum một cách rõ ràng và dễ đọc.
- Cho phép người dùng tùy chỉnh ký hiệu số và tiền tố để đáp ứng nhu cầu cụ thể của họ.
- Cung cấp tính năng mở tệp để thuận tiện cho việc nạp dữ liệu ASCII từ các tệp văn bản.
- Đảm bảo tính linh hoạt trong việc thay đổi kích thước checksum và cách tính checksum.

5.4 Thách Thức Trong Thiết Kế Giao Diện

Trong quá trình thiết kế giao diện người dùng cho ASCII Converter, có một số thách thức cần được xem xét, bao gồm:

- Đảm bảo giao diện gọn gàng và không quá nhiều yếu tố để tránh làm cho ứng dụng trở nên phức tạp và khó sử dụng.
- Xử lý các tùy chọn người dùng như ký hiệu số và tiền tố một cách trực quan và hiệu quả.
- Tích hợp tính năng mở tệp văn bản một cách sao cho nó phù hợp với luồng làm việc tự nhiên của người dùng.
- Kiểm tra và đảm bảo tính ổn định và hiệu suất của giao diện khi xử lý dữ liệu lớn hoặc phức tạp.

5.5 Hình Ảnh Minh Họa

Đây là một số hình ảnh minh họa cho giao diện người dùng của ứng dụng :

➤ Number delimiter is “None”

The screenshot shows a web-based application window titled "ASCII, Hex, Binary, Decimal, Base64 converter". The interface is yellow and contains the following elements:

- Buttons: "Open File" (green) and "x Reset" (grey).
- Number delimiter: A dropdown menu set to "None" and an empty text input field.
- 0x/0b prefix: An unchecked checkbox.
- ASCII text: A text area containing "Hello Monday".
- Hex (bytes): A text area containing "48656C6C6F204D6F6E646179".
- Binary (bytes): A text area containing the binary string "010010000110010101101100011011000110111100100000010011010110111101101110011000101111001".
- Decimal (bytes): A text area containing the decimal string "7210110810811132771111010097121".
- Checksum: Two dropdown menus, the first set to "8-bit" and the second set to "Sum", followed by a text area showing "7C".
- A final "x Reset" button at the bottom.

➤ Number delimiter is ‘Space’

The screenshot shows a web application titled "ASCII, Hex, Binary, Decimal, Base64 converter". It has a yellow background and a light blue header bar. At the top, there are two buttons: "Open File" (green) and "x Reset" (grey). Below these is a "Number delimiter" section with a dropdown menu set to "Space" and an empty text input field. A checkbox labeled "0x/0b prefix" is unchecked. The "ASCII text:" section contains a text area with "Hello Monday". The "Hex (bytes):" section shows the hex value "48 65 6C 6C 6F 20 4D 6F 6E 64 61 79". The "Binary (bytes):" section shows the binary value "01001000 01100101 01101100 01101100 01101111 00100000 01001101 01101111 0110 1110 01100100 01100001 01111001". The "Decimal (bytes):" section shows the decimal value "72 101 108 108 111 32 77 111 110 100 97 121". The "Checksum" section has two dropdown menus set to "8-bit" and "Sum", and a text input field containing "7C". At the bottom, there is another "x Reset" button.

➤ Number delimiter is “Comma”

The screenshot shows the same web application as the previous one, but with the "Number delimiter" dropdown menu set to "Comma". The text input field next to it now contains a comma ",". The "Hex (bytes):" section now shows the hex value "48,65,6C,6C,6F,20,4D,6F,6E,64,61,79". The "Binary (bytes):" section now shows the binary value "01001000,01100101,01101100,01101100,01101111,00100000,01001101,01101111,0110 1110,01100100,01100001,01111001". The "Decimal (bytes):" section now shows the decimal value "72,101,108,108,111,32,77,111,110,100,97,121". All other elements, including the "ASCII text:" section, the "Checksum" section, and the buttons, remain the same as in the previous screenshot.

➤ Number delimiter is “User Define”

The screenshot shows a web-based application window titled "ASCII, Hex, Binary, Decimal, Base64 converter". The interface has a yellow background and includes the following elements:

- Buttons:** A green "Open File" button and a grey "x Reset" button at the top.
- Number delimiter:** A dropdown menu set to "User defined" with a text input field containing ",0x".
- 0x/0b prefix:** An unchecked checkbox.
- ASCII text:** A text input field containing "Hello Monday".
- Hex (bytes):** A text output field displaying the hexadecimal representation: "48,0x65,0x6C,0x6F,0x20,0x4D,0x6F,0x6E,0x64,0x61,0x79".
- Binary (bytes):** A text output field displaying the binary representation: "01001000,0x01100101,0x01101100,0x01101100,0x01101111,0x00100000,0x01001101,0x01101111,0x01101110,0x01100100,0x01100001,0x011111001".
- Decimal (bytes):** A text output field displaying the decimal representation: "72,0x101,0x108,0x108,0x111,0x32,0x77,0x111,0x110,0x100,0x97,0x121".
- Checksum:** A section with two dropdown menus. The first is set to "8-bit" and the second to "Sum". To the right, a text field shows the result "7C".
- Reset Button:** A grey "x Reset" button at the bottom.

Chương: Kết Luận Chung

1. Kết Luận

Sau thời gian thực hiện đề tài, tôi đã đạt được những kết quả quan trọng sau đây:

- **Hoàn Thành Mục Tiêu Ban Đầu:** Mục tiêu ban đầu của dự án là phát triển một ứng dụng cho phép chuyển đổi giữa các hệ thống số và thực hiện chuyển đổi ASCII đã được đáp ứng thành công. Ứng dụng của chúng tôi đã đem lại giá trị thực sự cho người dùng, cho phép họ thực hiện các phép chuyển đổi này một cách dễ dàng.
- **Tích Lũy Kiến Thức Quan Trọng:** Quá trình thực hiện dự án này đã giúp em hiểu rõ hơn về các khái niệm liên quan đến hệ thống số, thuật toán chuyển đổi và thiết kế giao diện người dùng. Kiến thức này có giá trị đặc biệt trong lĩnh vực phát triển phần mềm và công nghệ thông tin.
- **Bước Tiến Quan Trọng:** Đề tài này đại diện cho một bước tiến quan trọng trong quá trình phát triển và ứng dụng các khái niệm liên quan đến hệ thống số và mã ký tự ASCII. Ứng dụng của em có khả năng giúp người dùng hiểu và sử dụng các hệ thống số khác nhau cũng như thực hiện chuyển đổi ASCII một cách hiệu quả.

2. References

- [Decimal to Hexadecimal Converter \(rapidtables.com\)](https://rapidtables.com/convert/decimal-to-hex/)
- [ASCII text, Hex, Binary, Decimal, Base64 converter \(rapidtables.com\)](https://rapidtables.com/convert/text-to-hex/)
- [Number System \(Definition, Types, Conversion & Examples\) \(byjus.com\)](https://byjus.com/number-system/)
- [ASCII table - Table of ASCII codes, characters and symbols \(ascii-code.com\)](https://ascii-code.com/)