‏22/02/2023

תרגיל בית 1: Git

* **מטרת הפרויקט**:

יצירת מחשבון המרה בין בסיס בינארי לבסיס הקסדצימלי (בסיס 16) ומבסיס הקסדצימלי לבסיס בינארי.

* **הנחות יסוד בפיתוח המערכת:**

1. הקוד יוכל להחזיר מספר בינארי או הקסדצימלי (בסיס 16), במידה והמשתמש יכניס מספר שאינו אחד מהם, יתקבל פלט "לא תקין"
2. אם המשתמש לא יבחר איזו המרה הוא בוחר (האם מבינארי להקסדצימלי או להיפך), לא ניתן יהיה להריץ את הקוד.

* **תיאור מצבי הקיצון ושיטת הפתרון:**

1. עבור מספר בינארי שכמות הספרות שלו לא מתחלקת ב-4 תתבצע השלמה לכמות ספרות שמתחלקות ב-4 משמאל למספר (יתווספו אפסים על מנת שנוכל לבצע המרה).

* האלגוריתם יודע להתמודד עם קלט באותיות קטנות וגדולות.
* **תיעוד קוד התוכנית**:

**import** java.util.Scanner;

**public** **class** project1 {

**public** **static** **void** main(String[] args) {

Scanner scanner = **new** Scanner(System.***in***);

**while** (**true**) {

System.***out***.print("Enter a number or X to exit: ");

String num = scanner.nextLine();

**if** (num.equalsIgnoreCase("x")) {

System.***out***.println("Good Bye.");

**break**;

}

System.***out***.print("Choose the base of the number(2 or 16): ");

**int** base = scanner.nextInt();

**if** (base == 2) {

*binaryToHex*(num);

} **else** **if** (base == 16) {

*hexToBinary*(num);

} **else** {

System.***out***.println("Invalid base");

}

scanner.nextLine();

}

//scanner.close();

}

//method to convert Hexadecimal to Binary

**private** **static** **void** hexToBinary(String hexString) {

**int** i;

**char** ch;

String returnVal = "";

hexString = hexString.toUpperCase();

// loop through each char in the hexString and convert it to decimal equivalent

// and convert that decimal to binary

**for** (i = 0; i < hexString.length(); i++) {

ch = hexString.charAt(i);

// Check if the character is a valid Hexadecimal character

**if** (!Character.*isDigit*(ch) && !((**int**) ch >= 65 && (**int**) ch <= 70)) {

// It's not a valid Hexadecimal character, return error

System.***out***.println("Invalid Hexadecimal String");

**return**;

} **else** **if** ((**int**) ch >= 65 && (**int**) ch <= 70)

// It's not digit but a valid Hexadecimal character

// Convert alphabet to decimal equivalent using ASCII code

returnVal += *decimalToBinary*((**int**) ch - 55);

**else**

// It's a digit

returnVal += *decimalToBinary*(Integer.*parseInt*(String.*valueOf*(ch)));

}

System.***out***.println("Binary of Hex string " + hexString + " is " + returnVal);

}

// Function to convert Decimal to Binary

**private** **static** String decimalToBinary(**int** decimal) {

String returnVal = "";

// loop through until decimal becomes 0

**while** (decimal != 0) {

// Get the remainder on dividing the decimal integer by 2

// and keep on concatenation to the returnVal variable

returnVal = (decimal % 2) + returnVal;

// update the decimal integer by dividing by 2

decimal /= 2;

}

// If required pad '0' to make the returnedVal string multiple of 4

**while** (returnVal.length() % 4 != 0) {

returnVal = "0" + returnVal;

}

**return** returnVal;

}

**private** **static** **void** binaryToHex(String bitStr) {

// check if it's valid binary string

**int** i = 0;

**while** (i < bitStr.length()) {

**if** (bitStr.charAt(i) != '0' && bitStr.charAt(i) != '1') {

System.***out***.println("Invalid input string");

**return**;

}

i++;

}

// If required pad '0' to make input string multiple of 4

**while** (bitStr.length() % 4 != 0) {

bitStr = "0" + bitStr;

// System.out.println(bitStr);

}

**int** startPos = 0, bitPos = 0;

String hexString = "";

**int** decimalVal = 0;

// Get every 4 digits starting from last position and convert to Hex

**while** (startPos < bitStr.length()) {

// Convert 4 binary digits to decimal

**while** (bitPos < 4) {

decimalVal = (**int**) (decimalVal

+ Integer.*parseInt*("" + bitStr.charAt(bitStr.length() - startPos - 1)) \* Math.*pow*(2, bitPos));

bitPos++;

startPos++;

}

**if** (decimalVal < 10)

hexString = Integer.*toString*(decimalVal) + hexString;

**else**

hexString = (**char**) (decimalVal + 55) + hexString;

bitPos = 0;

decimalVal = 0;

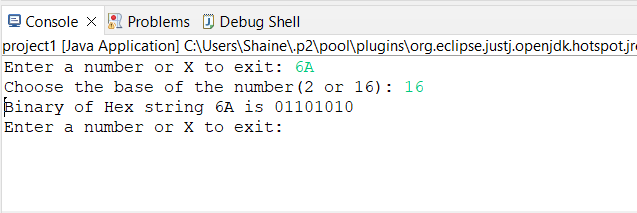
}

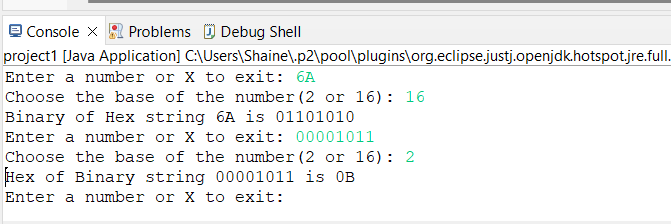
System.***out***.println("Hex of Binary string " + bitStr + " is " + hexString);

}

}

דוגמאות:





* **סיכום ומסקנות:**

בפרוייקט זה למדנו כיצד ליישם מעבר בין מספר בינארי למספר הקסדצימלי ולהיפך.

את לוגיקת ההמרה יישמנו בעצמנו במקום להשתמש בפונקציות מובנות, ולכן לימד אותנו בצורה מעמיקה יותר את המושגים הבסיסיים שבקורס.