baseNconverter

December 18, 2024

1 Base N Converter

1.0.1 Written By Ashar Adeel

Convert a positive integer between 0-2,000,000 into a different base, between 0-35. Encoded using 0-9 for first 10 values, and then a-z for the next 26. Command Line version (Version 1).

1.1 Program

[14]:

```
//WRITTEN BY ASHAR ADEEL
//17 DECEMBER 2024
//VERSION 1

//BASE N CONVERTER COMMAND LINE VERSION
//CONVERT ANY NUMBER (>2mill) TO ANY BASE (2<x<35)


import java.util.Scanner;
import java.io.*;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;

class data {
    int base;
    String outputValue;
    int inputValue;
    String extra;
}

class baseNConverter {
    //GLOBALS

    public static void main(String[] p) throws IOException{
        data user = createUser(-1,null,-1,null);
        System.out.println("BaseNConverter.");
        System.out.println("BASE: Must be between 2-35, will be encoded from␣
 ↪'0-9' and 'a-z' ");
        System.out.println("VALUEs: Must be positive, between 0 - 2 million.");
        System.out.println("Written by: ashar adeel");
        System.out.println("");
        System.out.println("");
        String askUse = input("Would you like to use BaseNConverter? (y/n)");
        while (!askUse.equalsIgnoreCase("y") && !askUse.equalsIgnoreCase("n"))␣
↪{ // If input is invalid
            askUse = input("Invalid input, choose between y/n.");
        }
        while (askUse.equalsIgnoreCase("y")) { // If yes, do this
            baseConverter(user);
            //SAVE CHECK
            String askSave = input("Would you like to save user data? (y/n)");{
                while(!askSave.equalsIgnoreCase("y") && !askSave.
↪equalsIgnoreCase("n")){
                    askSave = input("Invalid input, choose between y/n.");
                }
                if(askSave.equalsIgnoreCase("y")){
                    saveUserData(user);
                }
            }
            askUse = input("Would you like to convert another value? (y/n)");
        }
        if (askUse.equalsIgnoreCase("n")) { // If no, print this and close
            System.out.println("Thank you for using BaseNConverter");
        }

    }

    //SAVE SYSTEM
    public static void saveUserData(data user) throws IOException {
        String filename = input("Enter file name: ");
        File file = new File(filename + ".csv");

        // If the file exists, we append new data below the previous data
        boolean fileExists = file.exists();

        // Create or open the file for writing
        try (PrintWriter fileWrite = new PrintWriter(new FileWriter(filename +␣
↪".csv", true))) {
            // If the file doesn't exist, write the header
            if (!fileExists) {
                fileWrite.println("BaseNConverter, written by ashar adeel");
                fileWrite.println("BASE, INPUT VALUE, CONVERTED VALUE, STATUS,␣
↪TIMESTAMP");
            }

            // Write the new user data below the previous results
            fileWrite.println(getBase(user) + "," + getInput(user) + "," +␣
 ↪getOutput(user) + ",SAVED," + sysTime());
            System.out.println("User data saved successfully for " + filename);
        } catch (IOException e) {
            System.out.println("An error occurred while saving user data.");
            e.printStackTrace();
        }
    }

    //PARENT METHOD
    public static void baseConverter(data user){
        takeInputs(user);
        String value = assignedValue(getInput(user));
        System.out.println("VALUE = #" + convertToBase(user));
    }

    //CONVERSION CALCULATOR
    public static String convertToBase(data user){
        //DECLARE VARIABLES
        int base = getBase(user);
        int inputValue = getInput(user);
        int pointer = inputValue / base;
        String[] values = new String[pointer + 1];

        //CONVERSION
        while (inputValue != 0) {
            int remainder = inputValue % base;
            values[pointer] = assignedValue(remainder);
            inputValue = inputValue / base;
            pointer = pointer - 1;
        }

        //SET VALUES
        String output = buildOutput(values);
        setOutput(user,output);
        return output;
    }

    //BUILD FINAL RESULT
    public static String buildOutput(String[] values) {
        String result = "";
        for (String val : values) {
            if (val != null) {
                result += val; // Concatenate the non-null values
            }
        }
        return result;
    }

    public static String assignedValue(int value){
        String assVal = "";
        if(value>=0 && value <=9){
            assVal = String.valueOf(value);
        } else if(value == 10){
            assVal = "a";
        } else if(value == 11){
            assVal = "b";
        } else if(value == 12){
            assVal = "c";
        } else if(value == 13){
            assVal = "d";
        } else if(value == 14){
            assVal = "e";
        } else if(value == 15){
            assVal = "f";
        } else if (value == 16) {
            assVal = "g";
        } else if (value == 17) {
            assVal = "h";
        } else if (value == 18) {
            assVal = "i";
        } else if (value == 19) {
            assVal = "j";
        } else if (value == 20) {
            assVal = "k";
        } else if (value == 21) {
            assVal = "l";
        } else if (value == 22) {
            assVal = "m";
        } else if (value == 23) {
            assVal = "n";
        } else if (value == 24) {
            assVal = "o";
        } else if (value == 25) {
            assVal = "p";
        } else if (value == 26) {
            assVal = "q";
        } else if (value == 27) {
            assVal = "r";
        } else if (value == 28) {
            assVal = "s";
        } else if (value == 29) {
            assVal = "t";
        } else if (value == 30) {
            assVal = "u";
        } else if (value == 31) {
            assVal = "v";
        } else if (value == 32) {
            assVal = "w";
        } else if (value == 33) {
            assVal = "x";
        } else if (value == 34) {
            assVal = "y";
        } else if (value == 35) {
            assVal = "z";
        }
        return assVal;
    }

    //GATHER INPUTS
    public static void takeInputs(data user){
        //take base
        String userBase = input("Insert Base: "); //BASE GREATER THAN 2 AND␣
↪LESS THAN 35
        while(!isInteger(userBase) || Integer.parseInt(userBase)<2 || Integer.
↪parseInt(userBase)>35){
            userBase = input("Please enter a valid integer.");
        }
        //take value
        String userValue = input("Insert Value: "); //POSITIVE ONLY, must be␣
↪less than 20000
        while(!isInteger(userValue) || Integer.parseInt(userValue)<0 || Integer.
↪parseInt(userValue)>2000000){
            userValue = input("Please enter a valid integer.");
        }
        //set values
        setBase(user,Integer.parseInt(userBase));
        setInput(user,Integer.parseInt(userValue));
    }

    //CHECK IF INT
    public static boolean isInteger(String inputValue) {
        // Check if input is null or empty
        if (inputValue == null || inputValue.isEmpty())
        {
            return false;
        }

        // Check for the negative sign at the beginning
        int startIndex = 0;
        if (inputValue.charAt(0) == '-')
        {
            startIndex = 1;
        }

        // Check that every character after the negative sign (if present) is a␣
↪digit
        for (int i = startIndex; i < inputValue.length(); i++) {
            if (!Character.isDigit(inputValue.charAt(i)))
            {
                return false;  // If any character is not a digit, return false
            }
        }

        return inputValue.length() > startIndex;  // Ensure that the string has␣
↪at least one digit
    }

    //GATHER INPUTS
    public static String input(String text){
        System.out.println(text);
        Scanner scanner = new Scanner(System.in);
        String answer = scanner.nextLine();
        return answer;
    }

    //GET SYSTEM TIME FOR SAVE
    public static String sysTime() {
        LocalDateTime now = LocalDateTime.now();
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("hh:mm a,␣
↪dd-MM-yyyy");
        return now.format(formatter);
    }

    //  //  ADT METHODS
    //CREATE USER
    public static data createUser(int base, String out, int in, String x){
        data user = new data();
        user.base = base;
        user.outputValue = out;
        user.inputValue = in;
        user.extra = x;
        return user;
    }
    //GETTERS
    public static int getBase(data d) {
        return d.base;
    }
    public static String getOutput(data d) {
        return d.outputValue;
    }
    public static int getInput(data d) {
        return d.inputValue;
    }
    public static String getExtra(data d) {
        return d.extra;
    }
    //SETTERS
    public static void setBase(data d, int base) {
        d.base = base;
    }
    public static void setOutput(data d, String outputValue) {
        d.outputValue = outputValue;
    }
    public static void setInput(data d, int inputValue) {
        d.inputValue = inputValue;
    }
    public static void setExtra(data d, String extra) {
        d.extra = extra;
    }

}
```

1.2 Testing

[15]:

```
baseNConverter.main(null);
```

```
BaseNConverter.
BASE: Must be between 2-35, will be encoded from '0-9' and 'a-z'
VALUEs: Must be positive, between 0 - 2 million.
Written by: ashar adeel


Would you like to use BaseNConverter? (y/n)

 y

Insert Base:

 10

Insert Value:

 12351325

Please enter a valid integer.

 124
```

```
VALUE = #124
Would you like to save user data? (y/n)

 n

Would you like to convert another value? (y/n)

 y

Insert Base:

 2

Insert Value:

 321375

VALUE = #1001110011101011111
Would you like to save user data? (y/n)

 n

Would you like to convert another value? (y/n)

 y

Insert Base:

 16

Insert Value:

 193837

VALUE = #2f52d
Would you like to save user data? (y/n)

 n

Would you like to convert another value? (y/n)

 y

Insert Base:

 35

Insert Value:

 2000000

VALUE = #1bmmu
Would you like to save user data? (y/n)

 n

Would you like to convert another value? (y/n)

 y

Insert Base:
 0

Please enter a valid integer.

 1

Please enter a valid integer.

 36

Please enter a valid integer.

 35

Insert Value:

 -1

Please enter a valid integer.

 20000001

Please enter a valid integer.

 10

VALUE = #a
Would you like to save user data? (y/n)

 n

Would you like to convert another value? (y/n)

 n

Thank you for using BaseNConverter
```