

הפקולטה למדעי ההנדסה

המחלקה להנדסת תעשייה וניהול

### **תרגיל בית 1 בנושא Git**

קורס אוטומציה

364.1.3321

מגישה:

315474205

30.03.2023

I. מטרות הפרויקט –

- א. היכרות עם מערכת Git ומערכת Github.
- ב. תכנות מחשבון להמרת מספרים מבסיס עשרוני לבסיס הקסדצימלי ולהיפך.

II. הנחות יסוד בפיתוח המערכת –

- א. טווח המספרים שיוזן בקלט להמרת מספר מבסיס עשרוני ויודפס בפלט בהמרת מספר לבסיס 10 לא יהיה גדול מ-2147483647 (הערך המקסימלי שמשתנה מסוג int יכול לקבל ב-java).
- ב. יש להזין רק מספרים שלמים (ללא שברים).
- ג. יש להזין רק מספרים חיוביים (ללא מספרים שליליים).

III. תיאור מצבי הקיצון ושיטות הפתרון –

- א. מצב הקיצון: חריגת הטווח העליון של משתנה מסוג int בפלט בעת המרה לבסיס עשרוני.  
פתרון: הדפסת שגיאה למשתמש על חריגת הטווח העליון.
- ב. מצב הקיצון: חריגת הטווח העליון של משתנה מסוג int בקלט בעת המרה מבסיס עשרוני.  
פתרון: מתקבלת שגיאה אוטומטית מהמערכת על חריגה מהטווח העליון עבור סוג המשתנה. במידה והשגיאה לא "נתפסת" על ידי המערכת, תודפס שגיאה למשתמש בדומה לשגיאה בסעיף א'.
- ג. מצב הקיצון: כתיבת אותיות אנגליות בקלט בהמרה לבסיס עשרוני.  
פתרון: קליטת הקלט כמשתנה מסוג String והמרת ה-char-ים המרכיבים את ה-String למספרים (מספרים ואותיות כאחד) על מנת לחשב את ערך הקלט בבסיס עשרוני.
- ד. מצב הקיצון: הדפסת אותיות אנגליות בפלט בהמרה לבסיס הקסדצימלי.  
פתרון: שימוש במשתנה מסוג String לכתיבת הפלט. המספרים מומרים ל-String בעזרת פונקציות קיימות ב-java והמספרים 10-15 מומרים "ידינית" לאותיות.
- ה. מצב הקיצון: בחירת בסיס להמרה שאינו 10 או 16.  
פתרון: כצעד מונע, בתחילת התוכנית מודפס למשתמש סוגי הבסיסים והמשתנים שלהם. במידה והמשתמש בחר סוג בסיס שאינו 10 או 16, תודפס לו הודעת שגיאה.
- ו. מצב הקיצון: כתיבת מספר קלט שגוי בהמרה מבסיס 16 לבסיס 10.  
פתרון: הדפסת הודעת שגיאה למשתמש והפסקת המרת המספר.
- ז. מצב הקיצון: כתיבת מספר קלט שגוי בהמרה מבסיס 10 לבסיס 16.  
פתרון: המערכת מחזירה שגיאה אוטומטית ומפסיקה את ריצת התוכנית.

## IV. תיעוד קוד התוכנית –

ייבוא חבילות לטובת מימוש פעולות שונות ב-java, קליטת בסיס ההמרה הרצוי מהמשתמש ואתחול משתנה isExceed המעיד לאלגוריתם האם התרחשה חריגת הטווח העליון של משתנה מסוג int.

```
import java.util.Scanner;
import java.lang.Math;
import java.io.*;

public class Main
{
    public static void main(String[] args)
    {
        Scanner scan = new Scanner (System.in);
        System.out.println("Please select the conversion base:");
        System.out.println("16 = Convert to Hexadecimal base, 10 = Convert to Decimal base");
        int base = scan.nextInt();
        boolean isExceed = false;
```

עבור המרה לבסיס 10, המספר המבוקש להמרה נקלט כ-String בשל טווח הספרות והאותיות. שיטת ההמרה מבסיס עשרוני:  $x_0 * 16^n + x_1 * 16^{n-1} + \dots + x_n * 16^0$ . sum יהווה את המספר הסופי בבסיס עשרוני (בכל איטרציה sum מקבל את הערך של ה-char במקום ה-i בחזקת המשתנה pow, משתנה יורד מאורך הקלט פחות אחת ועד אפס). עבור כל char בקלט מבוצעת בדיקה שלא התרחשה חריגת הטווח העליון של משתנה מסוג int (במידה וכן יודפס פלט מתאים למשתמש וההמרה תופסק).

```
if (base == 10)
{
    System.out.println("Enter the requested number:");
    Scanner stringScanner = new Scanner(System.in);
    String requestedNumber = stringScanner.nextLine();
    int sum = 0;
    int pow = requestedNumber.length() - 1;
    for (int i = 0; i < requestedNumber.length(); i++)
    {
        if (requestedNumber.charAt(i) == '1')
        {
            if (Math.pow(16, pow) > 2147483647)
            {
                System.out.println("The requested number to convert exceeds the upper range of an int variable (2,147,483,647), please select a new number");
                isExceed = true;
                break;
            }
            else
            {
                sum += Math.pow(16, pow);
            }
        }
        else if (requestedNumber.charAt(i) == '2')
        {
            if (Math.pow(16, pow) > 2147483647)
            {
                System.out.println("The requested number to convert exceeds the upper range of an int variable (2,147,483,647), please select a new number");
                isExceed = true;
                break;
            }
        }
    }
}
```

```

else
{
    sum += 2 * (Math.pow(16, pow));
}
}
else if (requestedNumber.charAt(i) == '3')
{
    if (Math.pow(16, pow) > 2147483647)
    {
        System.out.println("The requested number to convert exceeds the upper range of an int variable (2,147,483,647), please select a new number");
        isExceed = true;
        break;
    }
    else
    {
        sum += 3 * (Math.pow(16, pow));
    }
}
else if (requestedNumber.charAt(i) == '4')
{
    if (Math.pow(16, pow) > 2147483647)
    {
        System.out.println("The requested number to convert exceeds the upper range of an int variable (2,147,483,647), please select a new number");
        isExceed = true;
        break;
    }
    else
    {
        sum += 4 * (Math.pow(16, pow));
    }
}
else if (requestedNumber.charAt(i) == '5')
{
    if (Math.pow(16, pow) > 2147483647)
    {
        System.out.println("The requested number to convert exceeds the upper range of an int variable (2,147,483,647), please select a new number");
        isExceed = true;
        break;
    }
    else
    {
        sum += 5 * (Math.pow(16, pow));
    }
}
else if (requestedNumber.charAt(i) == '6')
{
    if (Math.pow(16, pow) > 2147483647)
    {
        System.out.println("The requested number to convert exceeds the upper range of an int variable (2,147,483,647), please select a new number");
        isExceed = true;
        break;
    }
    else
    {
        sum += 6 * (Math.pow(16, pow));
    }
}
else if (requestedNumber.charAt(i) == '7')
{
    if (Math.pow(16, pow) > 2147483647)
    {
        System.out.println("The requested number to convert exceeds the upper range of an int variable (2,147,483,647), please select a new number");
    }
}

```

```
        isExceed = true;
        break;
    }
    else
    {
        sum += 7 * (Math.pow(16, pow));
    }
}
else if (requestedNumber.charAt(i) == '8')
{
    if (Math.pow(16, pow) > 2147483647)
    {
        System.out.println("The requested number to convert exceeds the upper range of an int variable (2,147,483,647), please select a new number");
        isExceed = true;
        break;
    }
    else
    {
        sum += 8 * (Math.pow(16, pow));
    }
}
else if (requestedNumber.charAt(i) == '9')
{
    if (Math.pow(16, pow) > 2147483647)
    {
        System.out.println("The requested number to convert exceeds the upper range of an int variable (2,147,483,647), please select a new number");
        isExceed = true;
        break;
    }
    else
    {
        sum += 9 * (Math.pow(16, pow));
    }
}
else if (requestedNumber.charAt(i) == 'A')
{
    if (Math.pow(16, pow) > 2147483647)
    {
        System.out.println("The requested number to convert exceeds the upper range of an int variable (2,147,483,647), please select a new number");
        isExceed = true;
        break;
    }
    else
    {
        sum += 10 * (Math.pow(16, pow));
    }
}
else if (requestedNumber.charAt(i) == 'B')
{
    if (Math.pow(16, pow) > 2147483647)
    {
        System.out.println("The requested number to convert exceeds the upper range of an int variable (2,147,483,647), please select a new number");
        isExceed = true;
        break;
    }
    else
    {
        sum += 11 * (Math.pow(16, pow));
    }
}
else if (requestedNumber.charAt(i) == 'C')
{

```

```

if (Math.pow(16, pow) > 2147483647)
{
    System.out.println("The requested number to convert exceeds the upper range of an int variable (2,147,483,647), please select a new number");
    isExceed = true;
    break;
}
else
{
    sum += 12 * (Math.pow(16, pow));
}
}
else if (requestedNumber.charAt(i) == 'D')
{
    if (Math.pow(16, pow) > 2147483647)
    {
        System.out.println("The requested number to convert exceeds the upper range of an int variable (2,147,483,647), please select a new number");
        isExceed = true;
        break;
    }
    else
    {
        sum += 13 * (Math.pow(16, pow));
    }
}
}
else if (requestedNumber.charAt(i) == 'E')
{
    if (Math.pow(16, pow) > 2147483647)
    {
        System.out.println("The requested number to convert exceeds the upper range of an int variable (2,147,483,647), please select a new number");
        isExceed = true;
        break;
    }
    else
    {
        sum += 14 * (Math.pow(16, pow));
    }
}
else if (requestedNumber.charAt(i) == 'F')
{
    if (Math.pow(16, pow) > 2147483647)
    {
        System.out.println("The requested number to convert exceeds the upper range of an int variable (2,147,483,647), please select a new number");
        isExceed = true;
        break;
    }
    else
    {
        sum += 15 * (Math.pow(16, pow));
    }
}
}
}

```

במידה ויש שגיאה בקלט מודפסת שגיאה למשתמש וההמרה מסתיימת.  
במידה ולא צצו שגיאות קלט (חריגת טווח או שגיאה בכתיבת הקלט), מודפס המספר המתקבל.

```

else
{
    System.out.println("The requested number to convert is not in 16 base range");
    isExceed = true;
    break;
}
pow--;
}
if (isExceed == false)
{
    System.out.println("The number in base 10 is " + sum);
}
}

```

עבור המרה לבסיס 16, הקלט ייקלט לתוך משתנה מסוג int בשל אופי הבסיס ממנו מתבצעת ההמרה.  
מתבצעת בדיקה שהמספר שהוזן אינו חורג את הטווח העליון של משתנה מסוג int (במידה והמערכת לא  
זיהתה את החריגה).

```

else if (base == 16)
{
    System.out.println("Enter the requested number:");
    Scanner intScanner = new Scanner(System.in);
    int requestedNumber = intScanner.nextInt();
    if (requestedNumber > 2147483647)
    {
        System.out.println("The requested number to convert exceeds the upper range of an int variable (2,147,483,647), please select a new number");
        isExceed = true;
    }
}
}

```

במידה ולא בוצעה חריגה, מתחילה ההמרה. ההמרה מתבצעת באמצעות חילוק המספר המקורי (ובהמשך תוצאת החילוק) ב-16, והוספת שאריות החילוק כספרות למשתנה מסוג String מימין לשמאל (מקביל לקריאת המספר מלמטה למעלה כאשר כותבים את החישובים בעמודה). במידה והשארית שווה למספר בין 10-15, היא תכתב במספר הסופי שיוצג כאות המתאימה.

```
if (isExceed == false)
{
    int prevResult = 0;
    int result = requestedNumber / 16;
    int remnant = requestedNumber - (result * 16);
    String num = null;
    String finalNum = String.valueOf(remnant);
    while (result != 0)
    {
        prevResult = result;
        result = prevResult / 16;
        remnant = prevResult - (result * 16);
        num = finalNum;
        if (remnant == 10)
        {
            finalNum = 'A' + num;
        }
        else if (remnant == 11)
        {
            finalNum = 'B' + num;
        }
        else if (remnant == 12)
        {
            finalNum = 'C' + num;
        }
        else if (remnant == 13)
        {
            finalNum = 'D' + num;
        }
        else if (remnant == 14)
        {
            finalNum = 'E' + num;
        }
    }
}
```

בסיום, מודפס המספר המתקבל (בהנחה שלא בוצעה חריגה של הטווח העליון של משתנה מסוג int).

```
    }
    else if (remnant == 15)
    {
        finalNum = 'F' + num;
    }
    else
    {
        finalNum = String.valueOf(remnant) + num;
    }
}
System.out.println("The number in base 16 is " + finalNum);
}
```

במידה ונבחר בסיס שאינו 10 או 16 להמרה, תודפס הודעת שגיאה למשתמש.

```
else
{
    System.out.println("Error in base type, please close the program and try again");
}
}
```

V. סיכום ומסקנות – במהלך הכנת העבודה תרגלתי ורעננתי תכנות בשפת java, ביצוע המרות מבסיס עשרוני לבסיס הקסדצימלי ולהיפך ואת העבודה עם המערכות Git ו-Github. העבודה חידדה לי את החשיבות של יצירת סכמת פתרון טרם כתיבת הפתרון. כתיבת סכמת הפתרון סייעה לי למצוא מקרי קיצון בקוד ואת פתרונם, להגדיר את הנחות העבודה (מה שהקל על כתיבת הפתרון, כיוון שנקודות המוצא ברורה) ולכתוב פתרון מסודר יותר לוגית.