

מכללת הדסה, החוג למדעי המחשב
תכנות מונחה עצמים ופיתוח משחקים
סמסטר ב', תשע"ט

תרגיל 1

תאריך אחרון להגשה:

קמפוס שטראוס גברים / נשים – יום ג' י"ב אדר ב' 19.3.2019 עד השעה 23:59

מטרות התרגיל:

רענון וחזרה על העקרונות שנלמדו בסמסטר א', ובכללם תכנון ממשק, העמסת אופרטורים, ירושה ופולימורפיזם וכן שימוש בכלים מהספרייה הסטנדרטית שנלמדו כולל מצביעים חכמים.

תיאור כללי:

בתרגיל זה נממש מחשבון צורות הניתן לתכנות. המחשבון מחשב את שטח הצורות שנבחר על ידי המשתמש. המשתמש יכול גם להוסיף וליצור צורות נוספות למחשבון מהצורות הראשוניות ואחר כך לבקש חישוב של השטחים והנפחים שלהן. בתרגיל זה נחזור לעבוד בטרמינל, לשם הפשטות.

פירוט הדרישות:

המחשבון מסוגל להחזיק רשימת צורות בסיסיות שעליהן הוא מסוגל לבצע מספר פעולות כפי שיבואר בהמשך. שלוש הצורות הבסיסיות הן:

- משולש - triangle
- מלבן - rectangle
- מעגל - circle

מהלך ריצת התוכנית:

בכל פעם המחשבון מדפיס למסך את רשימת הצורות הנוכחית, מזכיר את הפקודה לקבלת עזרה, ומחכה לקבלת פקודה מהמשתמש. כשהמשתמש מסיים להכניס את הפקודה, המחשבון מבצע אותה ושוב מדפיס את רשימת הפונקציות העדכנית וכן הלאה.

לכל צורה ברשימה יש מספר סידורי (המספרים תמיד רציפים) שמודפס לידה בהדפסת רשימת הפונקציות, ובעזרתה המשתמש יכול להתייחס אליה בפקודות שהוא מקליד.

רשימת הפקודות האפשריות:

הערה: הפקודה שהמשתמש מקליד היא תחילת המילה. בסוגריים מוצגת השלמת המילה המלאה כדי לעזור בהבנת המשמעות. גם במסך העזרה נדפיס כך.

create #shape x [y] – יוצרת אחת מהצורות הבסיסיות (R – מלבן, C – מעגל, T – משולש) עם הממדים $x [y]$ לצורך הכנת הצורה (ראו הדגמה בהמשך)

area num – מחשבת את שטח הצורה שמספרה num, במשולש נניח לשם הפשטות שבסיסו מקביל לציר ה-X והוא שווה שוקיים.

peri(meter) num – מחשבת את היקף הצורה שמספרה num,

mul(tiply) num X – מוסיפה לרשימת הצורות צורה שמספרה num, איקס פעמים. כמובן שחישוב השטח יתייחס לכל הצורות ביחד בהיבחר הצורה החדשה. הערה: מספר הפעמים X הינו מטיפוס unsigned int

add num1 num2 – מוסיפה לרשימת הצורה חיבור שתי הצורות שמספרן num1 ו-num2. כמובן שחישוב השטח יתייחס לכל הצורות ביחד בהיבחר הצורה החדשה

sub num1 num2 – מוסיפה לרשימת הצורה חיסור הצורה num2 מהצורה num1. כמובן שחישוב השטח יתייחס לכל הצורות ביחד בהיבחר הצורה החדשה.

draw num – תצייר את הצורה למסך. אין צורך להתחשב במידות הצורה, הציור יכול להיות אחיד לכל העיגולים, למשל. אם הצורה מורכבת מכפל צורות תוצג הצורה בתוספת X מספר הפעמים. אם הצורה הינה חיבור 2 צורות יציורו שתיהן בתוספת הסימן +. ובחיסור תוצגנה שתיהן עם סימן ה-"-" ביניהן.

max n #num1 #... numN – מדפיסה את ערך השטח הגדול מבין הצורות שיושאו. n - מספר הצורות שיושאו. num1-numN - מספרי הצורות המצויים כעת במחשבון

min n #num1 #... numN – מדפיסה את ערך השטח הקטן מבין הצורות שיושאו. n - מספר הצורות שיושאו. num1-numN - מספרי הצורות המצויים כעת במחשבון

same num – מחזירה את הצורות ששטחן והיקפן שווה לצורה #num ומדפיסה תת רשימה שלהם למסך (אפשרי עם האינדקס המקורי שלהם).

del(ete) num – מוחקת את הצורה שמספרה num מרשימת הצורות. שימו לב שאם יש צורות שנשמכות על הצורה הזו (צורות שיצרנו תוך שימוש בצורה הזו כאבן בניין) הצורות הללו עדיין צריכות להמשיך לעבוד כרגיל. מצד שני, מספרי הצורות שנשארו ברשימה צריכים להישאר רציפים, כמו שהוזכר לעיל.

help – מדפיסה מסך עזרה עם רשימת הפקודות האפשריות והסבר קצר עליהן.

exit – מדפיסה למסך "Goodbye" ויוצאת מהתוכנית.

דוגמה לריצת התוכנית:

הערה: אין צורך לעקוב בצורה מדויקת אחרי הניסוח כאן, אבל כן לעקוב אחרי הקו הכללי.

הקלט מהמשתמש מסומן ברקע צהוב.

הפלט ההתחלתי:

```
This is the shapes list:
```

```
Please enter a command ("help" for command list):
```

```
help
```

```
Following is the list of the calculator's available commands:
```

```
cre(ate) #shape <R - rectangle | T-triangle | C- circle>
```

```
area num - Computes the area of shape #num
```

```
per(imeter) num - Computes the perimeter of shape #num
```

```
draw num - Draw the shape #num
```

```
mul(tiply) num x - Creates a function that is the multiplication  
of shape #num x times
```

```
add num1 num2 - Creates a function that is the sum of function  
#num1 and function #num2
```

```
sub num1 num2 - Creates a function that is the subtraction of  
function #num1 and function #num2
```

```
min n #num1-#numN - Returns the smallest area in the chosen  
shapes. n - the requested shapes: #num1...
```

```
max n #num1-#numN - Returns the biggest area in the chosen shapes.  
n - the requested shapes: #num1...
```

```
same num - Returns shapes with which have the same area and  
perimeter to the shape #num
```

```
del(ete) num - Deletes function #num from function list
```

```
help - Prints this help screen
```

```
exit - Exits the program
```

```
This is the shapes list:
```

```
Please enter a command ("help" for command list):
```

יצירת הצורות:

```
Please enter a command ("help" for command list): cre r
```

Please enter length of the width:

10

Please enter length of the height:

10

This is the shapes list:

0: Rectangle(w: 10, h: 10)

Please enter a command ("help" for command list):

Please enter a command ("help" for command list): cre t

Please enter length of the width:

10

Please enter length of the height:

10

This is the shapes list:

0: Rectangle(w: 10, h: 10)

1: Triangle(w: 10, h: 10)

Please enter a command ("help" for command list): cre c

Please enter length of the radius:

10

This is the shapes list:

0: Rectangle(w: 10, h: 10)

1: Triangle(w: 10, h: 10)

2: Circle(r:10)

חישוב שטחים

Please enter a command ("help" for command list): area 0

Area Rectangle(w: 10, h: 10) = 100

This is the shapes list:

0: Rectangle(w: 10, h: 10)

1: Triangle(w: 10, h: 10)

2: Circle(r:10)

Please enter a command ("help" for command list): area 1

Area Triangle(w: 10, h: 10) = 50

This is the shapes list:

0: Rectangle(w: 10, h: 10)

ציור הצורות

```
This is the shapes list:
0: Rectangle( w: 10, h: 10)
1: Triangle( w: 10, h: 10)
2: Circle( r:10)

Please enter a command ("help" for command list): draw 1
Triangle( w: 10, h: 10)
      *
    * *
  *     *
*       *
*       *
*       *
*       *
*****
```

This is the shapes list:

0: Rectangle(w: 10, h: 10)

1: Triangle(w: 10, h: 10)

2: Circle(r:10)

Please enter a command ("help" for command list): draw 2

Circle(r:10)

```
      ****
    **      **
  **        **
 *          *
 *          *
 *          *
 *          *
**         **
 **        **
      ****
```

This is the shapes list:

0: Rectangle(w: 10, h: 10)

1: Triangle(w: 10, h: 10)

2: Circle(r:10)

חיבור צורות

Please enter a command ("help" for command list): add 0 1

This is the shapes list:

0: Rectangle(w: 10, h: 10)

1: Triangle(w: 10, h: 10)

2: Circle(r:10)

3: (Rectangle(w: 10, h: 10)) + (Triangle(w: 10, h: 10))

Please enter a command ("help" for command list): add 2 3

This is the shapes list:

0: Rectangle(w: 10, h: 10)

1: Triangle(w: 10, h: 10)

2: Circle(r:10)

```

3: (Rectangle( w: 10, h: 10)) + (Triangle( w: 10, h: 10))
4: (Circle( r:10)) + ((Rectangle( w: 10, h: 10)) + (Triangle( w:
10, h: 10)))

```

Please enter a command ("help" for command list): **sub 2 0**

This is the shapes list:

```

0: Rectangle( w: 10, h: 10)
1: Triangle( w: 10, h: 10)
2: Circle( r:10)
3: (Rectangle( w: 10, h: 10)) + (Triangle( w: 10, h: 10))
4: (Circle( r:10)) + ((Rectangle( w: 10, h: 10)) + (Triangle( w:
10, h: 10)))
5: (Circle( r:10)) - (Rectangle( w: 10, h: 10))

```

Please enter a command ("help" for command list): **draw 5**

Circle(r:10)

```

      ****
    **      **
  **          **
*              *
*              *
*              *
*              *
**            **
  **          **
      ****

```

- Rectangle(w: 10, h: 10)

```

*****
*      *
*      *
*      *
*      *
*      *
*      *
*      *

```

sub 2 0

Invalid operation: The minuend is smaller than subtrahend

This is the shapes list:

0: Rectangle()

1: Triangle()

2: Circle()

3: (Rectangle()) + (Triangle())

Please enter a command ("help" for command list):

Please enter a command ("help" for command list): mul 2 3

This is the shapes list:

0: Rectangle(w: 10, h: 10)

1: Triangle(w: 10, h: 10)

2: Circle(r:10)

3: (Rectangle(w: 10, h: 10)) + (Triangle(w: 10, h: 10))

4: (Circle(r:10)) + ((Rectangle(w: 10, h: 10)) + (Triangle(w: 10, h: 10)))

5: (Circle(r:10)) - (Rectangle(w: 10, h: 10))

6: (Circle(r:10)) * (3) times

Please enter a command ("help" for command list):

del 4

This is the shapes list:

0: Rectangle()

1: Triangle()

2: Circle()

3: (Rectangle()) + (Triangle())

4: (Circle()) * (3) times

Please enter a command ("help" for command list):

area 4 10 10

Area (Circle(10, 10)) * (3) times = 235.619

This is the shapes list:

```
0: Rectangle()
1: Triangle()
2: Circle()
3: (Rectangle()) + (Triangle())
4: (Circle()) * (3) times
```

מחיקת צורות

Please enter a command ("help" for command list): **del 2**

This is the shapes list:

```
0: Rectangle( w: 10, h: 10)
1: Triangle( w: 10, h: 10)
2: (Rectangle( w: 10, h: 10)) + (Triangle( w: 10, h: 10))
3: (Circle( r:10)) + ((Rectangle( w: 10, h: 10)) + (Triangle( w:
10, h: 10)))
4: (Circle( r:10)) - (Rectangle( w: 10, h: 10))
5: (Circle( r:10)) * (3) times
```

Please enter a command ("help" for command list): **area 1**

Area Triangle(w: 10, h: 10) = 50

This is the shapes list:

```
0: Rectangle( w: 10, h: 10)
1: Triangle( w: 10, h: 10)
2: (Rectangle( w: 10, h: 10)) + (Triangle( w: 10, h: 10))
3: (Circle( r:10)) + ((Rectangle( w: 10, h: 10)) + (Triangle( w:
10, h: 10)))
4: (Circle( r:10)) - (Rectangle( w: 10, h: 10))
5: (Circle( r:10)) * (3) times
```

exit

Goodbye.

הערות לגבי צורת המימוש:

- כדאי להשקיע זמן בתיכון התוכנית. עם תיכון נכון, הקוד יהיה די מינימלי ופשוט לכתובה. בפרט, אל תחששו להשתמש בכלים של הספרייה הסטנדרטית כמו וקטור ומצביעים חכמים, הם יהפכו את הקוד שלכם להרבה יותר נקי, יעיל וקצר ויקלו עליכם בכתיבת קוד שעובד נכון.

- כל המספרים בפונקציות בתוכנית הם double למעט מקרים שצויין בפירוש אחרת.
- ניתן להניח שהקלט תקין מהבחינה שבמקום שאנחנו מצפים למספר אכן קיבלנו מספר. כן צריך לבדוק את שם הפקודה ואת מספרי הצורות.

קובץ ה-README:

יש לכלול קובץ README שיקרא README.doc, README.docx או README.txt (ולא בשם אחר).

קובץ זה יכיל לכל הפחות:

1. כותרת.
 2. פרטי הסטודנט: שם מלא כפי שהוא מופיע ברשימות המכללה, ת"ז.
 3. הסבר כללי של התרגיל.
 4. תיכון (design): הסבר קצר מהם האובייקטים השונים בתוכנית, מה התפקיד של כל אחד מהם וחלוקת האחריות ביניהם ואיך מתבצעת האינטראקציה בין האובייקטים השונים.
 5. רשימה של הקבצים שנוצרו ע"י הסטודנט, עם הסבר קצר (לרוב לא יותר משורה או שתיים) לגבי תפקיד הקובץ.
 6. מבני נתונים עיקריים ותפקידיהם.
 7. אלגוריתמים הראויים לציון.
 8. באגים ידועים.
 9. הערות אחרות.
- יש לתמצת ככל שניתן אך לא לוותר על אף חלק. אם אין מה להגיד בנושא מסוים יש להשאיר את הכותרת ומתחתיו פסקה ריקה. תכתבו ב-README כל דבר שרצוי שהבודק ידע כשהוא בודק את התרגיל.

אופן ההגשה:

הקובץ להגשה: יש לדחוס כל קובץ הקשור לתרגיל, למעט מה שיצוין להלן, לקובץ ששמו exN_name.zip, כאשר N הוא מספר התרגיל ו-name הוא השם המלא. במקרה של הגשה בזוג, שם הקובץ יהיה לפי התבנית exN_name1_name2.zip, עם שמות המגישים בהתאמה (ללא רווחים; גם בשמות עצמם יש להחליף רווחים בקו תחתי או להצמיד את שני חלקי השם).

לפני דחיסת תיקיית ה-Solution שלכם יש למחוק את הפריטים הבאים:

- תיקיות בשם debug, release או x64 (לרוב יש יותר מתיקייה אחת בשם זה, אחת בתיקיית ה-Solution ואחת בתיקיית כל פרויקט).
- תיקייה (לפעמים מוסתרת!) בשם vs. שאמורה להיות בתיקייה עם ה-Solution.

ככלל אצבע, אם קובץ ה-zip שוקל יותר ממ"ב אחד או שניים, כנראה שלא מחקתם חלק מהקבצים הבינאריים המוזכרים.

את הקובץ יש להעלות ל-Moodle של הקורס למשימה המתאימה.

הגשה חוזרת: אם מסיבה כלשהי סטודנט מחליט להגיש הגשה חוזרת יש לוודא ששם הקובץ זהה לחלוטין לשם הקובץ המקורי. אחרת, אין הבודק אחראי לבדוק את הקובץ האחרון שיוגש.

כל שינוי ממה שמוגדר פה לגבי צורת ההגשה ומבנה ה-README עלול לגרום הורדת נקודות בציון.

מספר הערות:

1. שימו לב לשם הקובץ שאכן יכלול את שמות המגשים.

2. שימו לב שעליכם לשלוח את תיקיית ה-solution כולה, לא רק את קובצי הקוד שיצרתם. עקבו אחרי ההסבר המפורט באתר, במקרה שאתם לא בטוחים איך למצוא את התיקייה. תרגיל שלא יכלול את ה-solution, לא יתקבל וידרוש הגשה חוזרת (עם כללי האיחור הרגילים).

המלצה כללית: אחרי שהכנסתם את הקובץ להגשה, העתיקו אותו לתיקייה חדשה, חלצו את הקבצים שבתוכו ובדקו אם אתם מצליחים לפתוח את ה-solution שבתוכו ולקמפל את הקוד. הרבה טעויות של שכחת קבצים יכולות להימנע על ידי בדיקה כזו.

בהצלחה!