

**תכנות פונקציונלי ומבוזר - 046273**

**תרגיל בית 1: תכנות טורי**

תאריך הגשה מפורסם ב-moodle

## תרגיל ראשון – תכנון מערכת לחישובים של צורות גיאומטריות

### תיאור הבעיה:

בתרגיל זה נבצע מניפולציות על מבנה מתונים מסוג shapes. מבנה הנתונים הנ"ל מוגדר בצורה הבאה:  
{shapes, [Shape1, Shape2,...]}

כלומר, tuple שהאיבר הראשון שלו הוא ה-atom "shapes" והאיבר השני שלו הוא רשימה של איברים שיוגדרו בהמשך. הרשימה יכולה להיות ריקה או להכיל מספר כלשהוא של איברים.

איברי הרשימה הנתמכים במערכת הינם מלבנים (כולל ריבועים כמקרה פרטי), משולשים ואליפסות (כולל עיגולים כמקרה פרטי).

מלבן נשמר במבנה הבא:

{rectangle, {dim, Width, Height}}

אם הגובה והרוחב זהים, זהו ריבוע. פרט לגובה ורוחב זהים, אין סימון מיוחד לריבועים.

משולש נשמר במבנה הבא:

{triangle, {dim, Base, Height}}

אליפסה נשמרת במבנה הבא:

{ellipse, {radius, Radius1, Radius2}}

אם שני הרדיוסים זהים, זהו עיגול. פרט לרדיוסים זהים, אין סימון מיוחד לעיגולים.

שימו לב כי עבור כלל הצורות, פרמטרים גיאומטריים חוקיים הינם מספרים חיוביים בלבד. למשל,

{triangle, {dim, 0, 13}}, {ellipse, {radius, -7, 3}}

אינם חוקיים, בעוד שהאיבר

{rectangle, {dim, 1.5, 7.5}}

הינו חוקי.

מבנה נתונים מסוג shapes ייחשב כחוקי רק אם כל איברי הרשימה חוקיים.

## מימוש נדרש:

עליכם לממש מודול בשם **shapes** אשר יחשוף את הפונקציות הבאות:

shapesArea/1

פונקציה המקבלת מבנה נתונים מסוג shapes ומחזירה את סכום שטחי הצורות הכלולות בו.

squaresArea/1

פונקציה המקבלת מבנה נתונים מסוג shapes ומחזירה את סכום שטחי הריבועים הכלולים בו.

trianglesArea/1

פונקציה המקבלת מבנה נתונים מסוג shapes ומחזירה את סכום שטחי המשולשים הכלולים בו.

shapesFilter/1

פונקציה המקבלת סוג צורה (rectangle או ellipse או triangle) כ-atom ומחזירה פונקציה אחרת. הפונקציה המוחזרת מקבלת מבנה נתונים מסוג shapes ומחזירה מבנה נתונים מסוג shapes הכולל רק את הצורה זאת.

shapesFilter2/1

פונקציה המקבלת סוג צורה (rectangle או ellipse או square או triangle או circle) כ-atom ומחזירה פונקציה אחרת. הפונקציה המוחזרת מקבלת מבנה נתונים מסוג shapes ומחזירה מבנה נתונים מסוג shapes הכולל רק את הצורה זאת.

סה"כ 5 פונקציות למימוש. כל הפונקציות המקבלות מבנה נתונים מסוג shapes צריכות לוודא כי הפרמטר המועבר הוא אכן מבנה נתונים חוקי מסוג shapes.

הסבר על השימוש בערך `?` ניתן למצוא בקישור הבא:

<http://erlang.org/doc/man/math.html>

## תרגיל שני – משחק גפרורים לשני שחקנים

תיאור הבעיה:

נתון משחק שני שחקנים המתנהל בצורה הבאה:  
שמים על השולחן מספר גפרורים מסודרים בשורה.  
כל שחקן לוקח בתורו גפרור אחד או שניים.  
המנצח הוא השחקן אשר לוקח את הגפרור האחרון.

מימוש נדרש:

עליכם לממש מודול בשם **game** אשר יחשוף את הפונקציות הבאות:

`canWin(N) -> true | false`

הפונקציה `canWin/1` מקבלת את מספר הגפרורים ההתחלתי `N` ומחזירה ערך האם השחקן הראשון יכול לנצח (ה-`atom` `true`) או שלא (ה-`atom` `false`).

`nextMove(N) -> {true,1} | {true,2} | false`

הפונקציה `nextMove/1` מקבלת את מספר הגפרורים ההתחלתי `N` ומחזירה ערך האם השחקן הראשון יכול ואם כן, כמה גפרורים לקחת בתור הראשון (אם יש שתי אפשרויות, יש להודיע רק על אחת מהן). אם אינו יכול לנצח, מחזירה את ה-`atom` `false`. שימו לב לקיום/אי קיום הסוגריים עבור האפשרויות השונות.

`explanation() -> { Answer }`

פונקציה `explanation/0` אשר לא מקבלת קלט ומחזירה tuple המכיל את התשובה לשאלה הבאה – "מהו הקושי במימוש הפתרון למשחק בעזרת רקורסיית זנב?". שימו לב – אין צורך לספק מימוש כזה.

**סה"כ 3 פונקציות למימוש. לשתי הפונקציות הראשונות יש לממש פתרון רקורסיבי.**

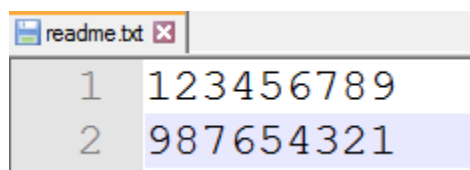
## הוראות הגשה:

- הגשה בזוגות בלבד. ניתן להיעזר בפורום באתר הקורס ע"מ למצוא שותף.
- הגשה אלקטרונית דרך מערכת ה-Moodle. הגשה אחת בלבד עבור שני בני הזוג. שאלות על תוכן התרגיל – דרך הפורום המתאים ב-Moodle.
- עליכם להגיש את כל קבצי המקור (כלומר, .erl). הכוללים תיעוד ומימוש. אין להגיש קבצים מקומפלים (.beam) או תתי-תיקיות.
- לא ניתן להגיש את התרגיל באיחור למעט מקרים פרטניים (מילואים וכו') – בתיאום **מראש** עם המרצה.

בדיקת התרגילים מתבצעת באופן אוטומטי, ולכן עליכם לוודא לפני ההגשה:

- שמות המודולים, הפונקציות והקבועים תואמים לשמות המצוינים בהנחיות התרגיל.
- לא מתבצעות הדפסות לא רצויות למסך.
- הקוד מתקמפל ורץ כהלכה.
- טיפול במקרי קצה – מספרים שליליים ואפס, רשימות ריקות וכו'. מדיניות הטיפול בשגיאות הינה זו שהוגדרה במהלך הקורס - let it crash.

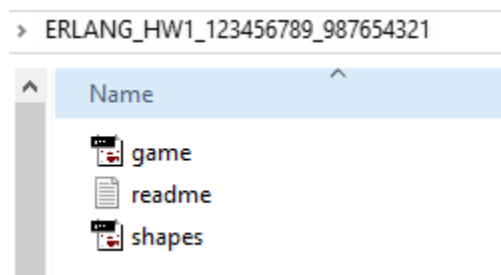
צרו קובץ בשם readme.txt המכיל את ת"ז של שני המגישים בשתי שורות נפרדות. לדוגמא:



את כל הקבצים יש לארוז בקובץ ZIP (בלבד). שם קובץ ההגשה יהיה

ERLANG\_HW1\_<id1>\_<id2>.zip

כאשר id1 ו-id2 הינם מספרי תעודות הזהות של המגישים. לדוגמא:



ERLANG\_HW1\_123456789\_987654321.zip

לאחר הגשת הקבצים ב-Moodle, אתם נדרשים להוריד את הקבצים שהגשתם ולראות כי הם מתקמפלים ורצים בסביבה נקייה.

הבהרה – אי עמידה בהוראות ההגשה שקולה לאי הגשת התרגיל. אתם נדרשים לוודא כי קיימתם את כל ההוראות המפורטות. לא תהיינה הגשות חוזרות.

## בהצלחה!