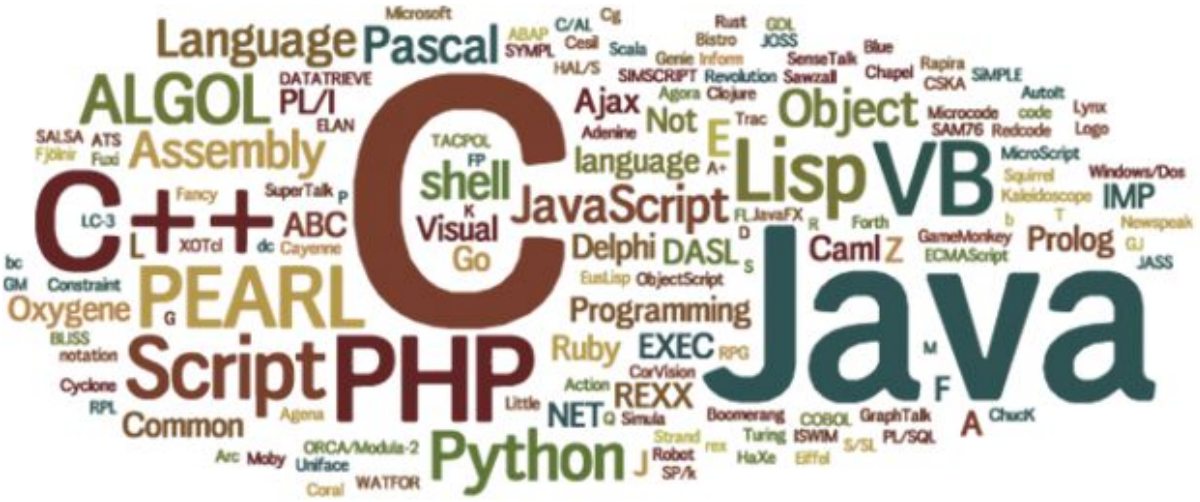
שפות תכנות, 234319

חורף 2018-2019



תרגיל בית 1

תאריך פרסום: 26/10/2018
מועד אחרון להגשה:08/11/2018
מועד אחרון להגשה מאוחרת:11/11/2018
מתרגל אחראי: יוסי גורשומוב
אי-מייל: yossigor@campus.technion.ac.il
בפניה בדוא"ל, נושא ההודעה (subject) יהיה "PL-EX1" (ללא המירכאות).

תרגיל בית זה מורכב משני חלקים, חלק יבש וחלק רטוב.
לפני ההגשה, ודאו שההגשה שלכם תואמת את הנחיות ההגשה בסוף התרגיל.
תיקונים והבהרות יפורסמו בסוף מסמך זה, אנא הקפידו להתעדכן לעתים תכופות.

חלק יבש

משימה 1

א. קרא את הפרק צעדים ראשונים, וחזור על סיכומי ההרצאות.

נתח את שפת התכנות RUST, באמצעות תיבת הכלים שהוצגה בשיעור ובסיכומים, ועל פי הסעיפים הבאים:

1. מהו ה"חזון" או ה"מטרה" שהביאה להשקעת המשאבים לפיתוח השפה? מנה 10-3 עקרונות שקבעו לעצמם מתכנני השפה.
2. אילו מהדורות יש לשפה? מה השינויים העיקריים שהיו בהן? מה תוכל להסיק, אם בכלל, מתכיפות מהדורות אלו ומהשינויים בשפה בין המהדורות, על הקשר בין השפה וקהל המשתמשים.
3. מהם הישויות שהן nameable בשפה? האם ניתן ליצר מופעים אנונימיים מישויות אלו? ממי מהן בדיוק?
4. אילו מנגנונים מספקת השפה לניהול מרחב השמות ולמתן אפשרות לשימוש חוזר בשמות טובים.
5. מהן הישויות בשפה שהן RECURSIVELY DEFINED?
6. מהן המילים השמורות בשפה? אילו מהן הן מזהים שמורים? אילו מהוות סימני פיסוק?
7. כיצד נקבעים גבולות התכנית בשפה? האם השפה אוטרקית? האם יש לה ספריה סטנדרטית?

משימה 2

חפש ומצא מהו המונח המקובל לתכנית שמדפיסה את עצמה. למד את שפת התכנות AWK עלי ידי קריאת דף המנואל (נסה להמנע מכל דרך אחרת) לשם כך, בלינוקס, הדפס MAN AWK בשורת הפקודה.

1. כתוב או מצא תכנית ב AWK שמדפיסה את עצמה. אם מצאת, צרף את הקישור.
2. כתוב תכנית ב AWK המדפיסה את עצמה שלוש פעמים.
3. האם השפה אוטרקית? הסבר.
4. ישנו ואריאנט של AWK, שמכיל הוראת INCLUDE. איזו בעיה מנסה הוואריאנט לפתור.

משימה 3

שפת AWK תומכת בביטויים רגולריים. מצא את התיעוד לדקדוק של ביטויים אלו (צרף קישור), והסבר כיצד קבוצה זו מוגדרת רקורסיבית

חלק רטוב

משימה 1 - סוגריים מאוזנים

כתבו את הפונקציה balance אשר בודקת איזון של סוגריים במשפט.

```
balance = fn : string -> bool
```

הפונקציה תקבל מחרוזת ותבדוק שהסוגריים במחרוזת מאוזנים, משמע יש מספר זהה של ')' ו'(' וגם לא ייתכן מצב בו ברישא של המחרוזת יופיע יותר סוגרים "סוגרים" מסוגרים "פותחים".

<u>דוגמאות הרצה</u>:

```
- balance "()";
val it = true : bool
- balance "if(true) then (foo(5))";
val it = true : bool

- balance ":(";
val it = false : bool

- balance ")(";
val it = false : bool
```

משימה 2 - atoi

כתבו את הפונקציה atoi המקבלת מחרוזת של מספר אי שלילי (בטווח החוקי של int) ומחזירה את המספר השלם שהיא מייצגת.

```
atoi = fn : string -> int
```

<u>דוגמאות הרצה:</u>

```
- atoi "123";
val it = 123: int
- atoi "9784";
val it = 9784: int
```

משימה 3 - reverseString

כתבו את הפונקציה

```
reverseString : string -> string
```

הפונקציה תקבל מחרוזת ותחזיר את המחרוזת כשהיא הפוכה.

<u>דוגמאות הרצה:</u>

```
- reverseString("hello world");
val it = "dlrow olleh" : string
- reverseString("");
val it = "" : string
```

משימה 4 - אלגוריתם אוקלידס המורחב

<u>תזכורת:</u>

לכל a ו b שלמים, קיימים שני מספרים שלמים x וy כך ש: $a \cdot x + b \cdot y = gcd(a,b)$ *, כלומר,*
המחלק המשותף הגדול ביותר של a ו b הוא צירוף לינארי של a ו b.
למשל עבור a=5 ו b=64 מתקיים:

gcd(5,64) = 1 , x = 13, y = -1

ואכן:

*13*5 + (-1)*64 = 1*

את gcd(a,b) , x , y מחשבים בעזרת [אלגוריתם אוקלידס המורחב](#) .

כתבו פונקציה המחשבת את אלגוריתם אוקלידס המורחב:

```
extended = fn : int * int -> int * int * int
```

הפונקציה extended מקבלת tuple שבו a ו b שני מספרים שלמים. ומחזירה tuple בעל שלושה איברים:

(gcd(a,b),x,y)

כך ש $a \cdot x + b \cdot y = gcd(a,b)$.

<u>דוגמאות הרצה:</u>

```
- extended(5,64);
val it = (1,13,~1) : int * int * int
- extended(64,5);
val it = (1,~1,13) : int * int * int
- extended(~7,80);
val it = (1,~23,~2) : int * int * int
- extended(3,9);
val it = (3,1,0) : int * int * int
```

הדרכה:

- שפת ML היא שפה מהפרדיגמה הפונקציונלית. עבור רבים ממכם ML היא השפה הפונקציונלית הראשונה שאתם פוגשים. לכן, עליכם לסגל לעצמכם את דרך החשיבה שנדרשת בתכנות בשפה פונקציונלית. למשל, בתרגול ראינו שני מימושים שונים עבור אלגוריתם אוקלידס (חישוב gcd). אחד בשפה מהפרדיגמה האימפרטיבית (פאסקל) והשני בשפה מהפרדיגמה הפונקציונלית (ML). ודאו שאתם מבינים את הדוגמא.
- קראו את האלגוריתם המוצג בדף הויקיפדיה של אלגוריתם אוקלידס המורחב, הוא מוצג בפסאודו קוד המזכיר תכנות אימפרטיבי. הבינו אותו, וודאו שאתם מסוגלים לכתוב תוכנית בשפה אימפרטיבית כמו שפת C הממשת אותו.
- וודאו שאתם מסוגלים לכתוב תוכנית הממשת אותו בשפת C, אך הפעם ללא לולאות (רמז: ניחשתם נכון, רקורסיה).
- כעת אתם בהחלט מוכנים לכתוב את הקוד בML הממש את הפונקציה extended.

הנחיות

- בתרגיל זה ניתן להשתמש רק בחומר שנלמד בשפת ML עד תרגול 2. אין להשתמש באף פונקציה או תכונה של השפה שלא נלמדה בתרגולים.

- רשימת הקבצים שצריכים להופיע בתוך קובץ ה-zip היא:

```
dry.pdf, sol.sml
```

- לכל קובץ שמכיל קוד הוסיפו בשורה הראשונה הערה המכילה את השם, מספר ת.ז. וכתובת המייל של המגישים מופרדים באמצעות רווח. וודאו שהקבצים נטענים/מתקמפלים ללא שגיאות גם לאחר הוספת ההערה שלעיל (**ובמיוחד לפני ההגשה!**)

- יש להגיש את הקבצים דחוסים בתוך קובץ zip. הקבצים יהיו בשורש קובץ ה-zip ולא בתוך ספרייה. שם הקובץ יהיה EX1_ID1_ID2.zip כאשר ID1,ID2 הם מספרי ת.ז. של המגישים.

- שימו לב שהבדיקה של החלק הרטוב היא אוטומטית, ולכן הקפידו על מילוי כל ההוראות בשביל למנוע בעיות מיותרות.

בהצלחה!

תיקונים והבהרות

לענין מיון האסימונים, המזהים, והמילים השמורות

1. אסימון -או token הוא יחידה סינטקטית, שהיא סימן סופי ב bnf שמגדיר שפת תכנות. סוגי האסימונים העיקריים הם
   a. מזהים, identifiers, או שמות, הם אסימונים שמציינים ישות. ישות היאו פרגמנט אחר של שפת התכנות שניתן לתת לו שם. משתנה, קבוע, , כגון משתנה, טיפוס, פונקציה, פרוצדורה, קבוע, כתובת (כזו שניתן לקפוץ אליה9, מחלקה, וכו. לא כל פרגמנט של שפת התכנות הוא ישות. למשל, במרבית שפות התכנות לא ניתן לתת שם לביטוי, או לפקודת For במקום מסויים בתכנית.
   b. ליטרלים, או מילולונים, הם אסימונים שמציינים את עצמם,
   c. סימני פיסוק, כגון נקודה ופסיק, פסיק, סוגריים וכו'.
   d. אופרטורים, כגון ++, *, וכו'. ניתן לחשוב על אופרטורים כשמות של פונקציות הבנויות בשפת התכנה. כך למשל, * הוא השם של הפונקציה המבצעת את הכפל.
   e. מילה שמורה, נקראת גם reserved word או keyword, היא אסימון שסינטקטית נראה כמו מזהה, אך המתכנת אינו רשאי להשתמש במילה שמורה כמזהה.
2. הסוגים העיקריים של מזהים הם-
   a. מזהה שמור מראש
   b. מזהה סיפריה
   c. מזהה המוגדר על ידי המשתמש.
   d. מזהה שהוא מילה שמורה, reserved identifier
3. הסוגים העיקריים של המילים השמורות
   a. מזהה שמור, כגון המילה int בשפת C (מזהה של הטיפוס של מספרים שלמים) או המילה
   b. סימן פיסוק, כגון begin, end אבל גם if ו for ואפילו גם class ו struct. מילים אלו הן עזר תחבירי בלבד.
   c. שמות של אופרטורים, כגון new ו delete בשפת ג'אווה ובשפת C++