

# תכנות מתקדם

## תרגיל 3

### parser

מועד אחרון להגשה: 21.5.2021

בתרגיל זה נבנה parser שפועל לפי טבלה כפי שראינו במצגת.

המימוש הוא עבור הדקדוק שבמצגת.

- ההגשה ביחיד לכתובת:

<http://submit.org.il/ariel>

לבחור קורס cpp38510

- צריך להעלות ארבעה קבצים:

`symbol.h`, `symbol.cpp`, `parser.h`, `parser.cpp`

בנוסף אפשר להעלות קבצים נוספים ולהכליל אותם עם `#include`

(בדפדפן, אפשר לבחור את כולם או להעלות קובץ זיפ שמכיל אותם)

בשרת הבדיקות נמצאים הקבצים `main.cpp` ו-`makefile` (המצורפים) ואי אפשר להעלותם.

מצורפים הקבצים `symbol.h` ו-`parser.h` ששמשו לפתרון התרגיל.

אפשר להעלות אותם כפי שהם, לשנות אותם או לכתוב אחרים.

## תאור ה- parser

ה- parser מבוסס על שתי מחלקות:

### 1. Symbol

המחלקה (כפי שמומשה) מכילה סמל שהוא או מילה (token, terminal) או משתנה (nonterminal) ומיוצג על ידי int.

מאחר שכללי הדקדוק הם תערובת של מילים ומשתנים יש צורך באובייקט שהוא או מילה או משתנה, מימוש יותר נכון הוא סמל כמחלקה אבסטרקטית, ושתי מחלקות יורשות לסמל, ושימוש ב- enum במקום int.

המחלקה מכילה מתודה להדפסת הסמל.

### 2. Parser

קובץ הכותרת מכיל את כללי הדקדוק ואת הטבלה.

- כללי הדקדוק (לפי המימוש) מכילים את הצד הימני של כל כלל דקדוק בסדר הפוך.
- בטבלה מופיע האינדקס של כלל הדקדוק, אם שגיאה אז -1.
- המחסנית ממומשת על ידי vector כדי לאפשר גישה לאיברים שבתוך המחסנית.

לצורך המימוש נעשה שימוש במתודות הבאות של vector:

begin, end, rbegin, rend, push\_back, pop\_back, back.

- המתודה printLM מדפיסה את הגזירה השמאלית של מחרוזת הקלט.
- משמאל לימין מודפסים המלים שכבר נגזרו, תוו | מפריד, ומשתנים ומילים שבתהליך גזירה.
- ההדפסה דומה לדוגמה שבמצגת.

- במקום E' מודפס E\_
- במקרה של שגיאה, יודפס "syntax error\n" וה- parser יעצור.

עבור הקלט:

id + id \* id

הפלט הנכון הוא:

| E

| T E\_

| F T\_ E\_

| id T\_ E\_

id | T\_ E\_ //id|T\_E\_\n

id | E\_

id | + T E\_

id + | T E\_

id + | F T\_ E\_

id + | id T\_ E\_

id + id | T\_ E\_

id + id | \* F T\_ E\_

id + id \* | F T\_ E\_

id + id \* | id T\_ E\_

id + id \* id | T\_ E\_

id + id \* id | E\_

id + id \* id | //id+id\*id|\n

**בהצלחה**