* המבנה הכללי הוא של עץ (האם צריך מצביעים כלפי מטה?).
* נבנה מחלקות של node-ים: Global, Method, If, While (לא ברור עדיין מה הקשר ביניהם – ירושה?)
* לכל node יהיה מבנה נתונים שיאכסן את הנתונים המוגדרים בו (אופציה: hash-map עם מפתחות שהם השמות של המשתנים וערך שהוא הפנייה לאובייקט משתנה המייצג אותם).
* מתודה שיודעת לנתח שורה של השמת משתנה (?) וזורקת exception אם היא לא תקינה.
* מתודה שיודעת לקבוע מה הסוג של הערך (?) וזורקת exception אם הערך אסור.
* בגלובל – מבנה נתונים נוסף שמאכסן את כל המתודות (שם ואולי גם הפניה?)
* בתוך כל אובייקט של משתנה:

- אינדיקציה האם הוא מאותחל

- אינדיקציה האם הוא final

- "רשימה" של סוגים שמותר להשים בהם.

-getters ו- setters כראות עינינו.

-מתודת compatible שקובעת האם סוג הערך תואם את סוג האובייקט וזורק exception אם לא.

-isArgument

* בכל אובייקט של scope יהיו שמורים איכשהו כל הפורמטים של שורות שה-scope יודע לקבל (הגדרת משתנה, השמה של משתנה, השמה והגדרה, קריאה למתודה, פתיחת מתודה, פתיחת if, פתיחת while, הערה, סגירה)

מחלקת מתודה:

* צריכה לאחסן איזה ארגומנטים מותר לה לקבל (אולי באמצעות מבנה נתונים של אובייקטים שייצגו את האובייקטים איתם היא הוגדרה – טוב ל- getNumberOfArguments) .
* הארגומנטים יכנסו גם למבנה הנתונים שמכיל משתנים **אבל** המחלקה צריכה לדעת לזהות מי מהמשתנים הוא ארגומנט.
* מתודת getNumberOfArguments
* ארגומנט שהוא final – לא מקבל פה ערך, זה תקין! לאחר מכן בתוך המתודה בלבד צריך להתייחס אליו כ final לכל דבר.
* משתנה גלובלי שלא אותחל – יכנס להאש מאפ של המתודה כשתהיה לו השמה (אם יהיה שימוש לפני כן זה אקספשן). העבודה מול הרפרנס שלו תהיה מול המתודה ולא בסקופ הגלובלי (כדי שישאר לא מאותחל)

If ו-while

* נגדיר להם הורה אבל ם לא מוגדרים כבנים של אף אחד.
* בדיקת תקינות התנאי – מותר Boolean/int/double . (שליפת הפרמטרים ע"י מתודה מlineAnalyzer)

סריקת המסמך

ריצה ראשונה על הסקופ הגלובלי בלבד (לפי return? או לפי ספירת סוגריים).

ריצה שניה – לדלג על כל מה שנמצא בסקופ הגלובלי ישירות.