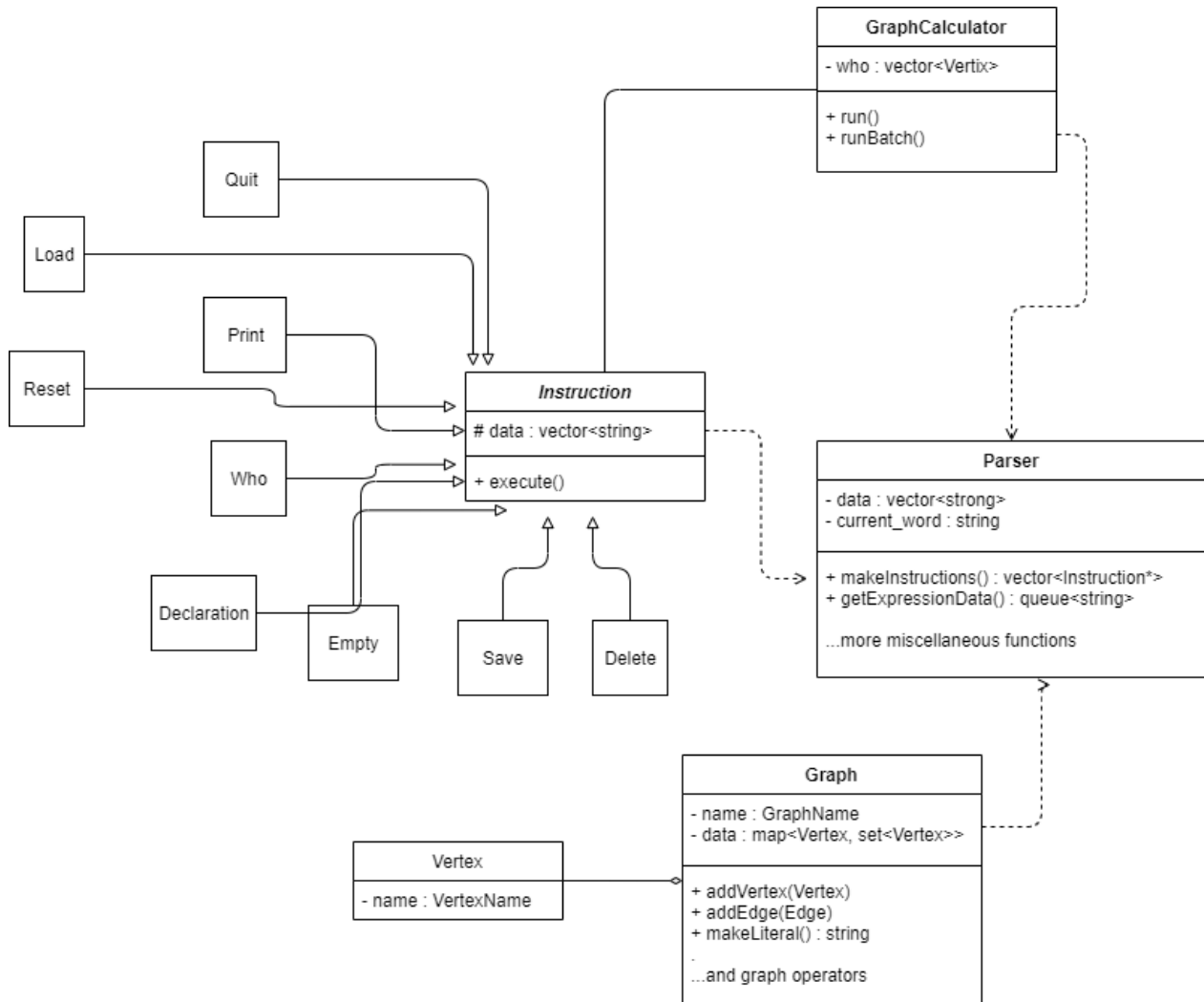


מסמך תיאור הפרויקט

א. דיאגרמת מחלקות:



ב.

המחלקה GraphCalculator מריצה את הפקודות שהתקבלו כקלט – Instruction, ושומרת מסד נתונים של כל הגרפים בזכרון. את הפקודות היא מקבלת בעזרת makeInstructions של Parser. בנוסף, GraphCalculator מטפל בכל השגיאות שנזכרות.

המחלקה Parser מעבדת את הקלט ומפרידה בין המילים בתחביר של המחשבון כך שהפקודות יבינו את הקלט. העיבוד נעשה באופן יבש במעבר על כל התווים בשורה ומפריד למקרים בהתאם. התפקיד של Parser הוא אף ורק לעבד את הקלט ולא להתמודד עם התחביר או לזרוק שגיאות.

המחלקה Instruction היא מחלקה אבסטרקטית שמשמשת כמחלקת בסיס לכל הפקודות האפשריות במחשבון כפי שמתואר בדיאגרמה (Empty היא פקודה ריקה שמתקבלת מקלט ריק). הפונקציה execute היא pure virtual וכל פקודה שירשת ממשת אותה בהתאם. כך GraphCalculator לא צריכה לדעת את סוג הפקודה ועדיין לקרוא לexecute בזמן ריצה. הפקודות שמכילים ביטויים נעזרות

בפונקציית עזר `evaluateExpression` שמתמודדת עם התחביר באופן רקורסיבי בשביל לשמור על סדר תקין תוך כדי אפשר שירשור פעולות. כמו כן, כל פקודה יודעת לזרוק שגיאה מתאימה.

המחלקה `Graph` היא מחלקה שמתארת גרף כפי שמוגדר במסמך הגדרת הפרוייקט וממשת את כל הפעולות שלה. בנוסף היא יודעת להמיר קבועי גרף לאובייקט ולהפך. היא שומרת על כל קודקוד במפה כך שהערך של כל קודקוד הוא אוסף שח קודודים המייצגים שקיימת קשת שנכנסת אליהם מהקודקוד המפתח.

ההמחלה `Vertex` ממשת קודקוד כמעטפת של שם הקודקוד. בנוסף שכחתי להוסיף בדיאגרמה `struct` שמייצג קשת `Edge` שמחזיק בתוכו 2 קודקודים מסודרים.