שם הפרוייקט

S-Emulator

מגישים

|  |  |  |
| --- | --- | --- |
| מייל ליצירת קשר | תז | שם |
|  |  |  |
| avshalomsh@mta.ac.il | 318799608 | אבשלום שהרבני |

בונוסים (אם מומשו)

|  |  |
| --- | --- |
| תיאור | שם |
| הוספתי את האפשרות להחליף סקינים למערכת, האפשרות להחליף נמצאת בכפתור בצד ימין בממשק המשתמש, ישנם שלושה סקינים שונים: dark,light and high contrast בנוסף מימשתי כפתור שנותן להגדיל את הגופן של הכתב. | אפשרות להחליף סקין |
| מימשתי 2 אנימציות, אחת מדגישה את הערך של משתנה הפלט במהלך ריצת הדיבאג, השני מגדיל את טבלת ההוראות בכל לחיצת  step over/step back יש צ'ק בוקס בצד ימין למטה, שמאפשר להוסיף/להוריד את האנימציות | מימוש 2-3 אנימציות |
| מימשתי את היכולת לבצע צעד לאחור בריצת דיבאג של היוזר. | Step back |

קישור לגיט:  
  
<https://github.com/AvshiSha/S-emulator>

# יכולות מערכת

## טעינת תוכניות

* תיבת דיאלוג לבחירת קובץ:

טעינת תוכניות מתבצעת באמצעות תיבת דיאלוג לבחירת קובץ (File Chooser) בממשק JavaFX. המשתמש יכול לבחור את הקובץ הרצוי לטעינה בצורה נוחה, כאשר המערכת פותחת את תיבת הדיאלוג המובנית של JavaFX ומאפשרת ניווט בין תיקיות ומסמכים.

במערכת יושם דגש על ניהול שגיאות יעיל, תוך הצגת הודעות מובנות, ברורות וידידותיות למשתמש. כאשר מתרחשות שגיאות בעת טעינת תוכניות, המשתמש מקבל פידבק מיידי המאפשר לו להבין את מהות הבעיה ולפעול בהתאם. למשל, במידה ומתגלה שגיאה במבנה קובץ ה-XML במהלך הטעינה, תוצג הודעה מפורטת המסבירה את מקור השגיאה ומציעה דרכי פתרון אפשריות.

## תמיכה בפקודות QUOTE ו-JUMP\_EQUAL\_FUNCTION

## פקודת QUOTE – שימוש בפונקציות כערכים ראשוניים עם העברת פרמטרים

המערכת כוללת תמיכה בפקודת QUOTE, המאפשרת להתייחס לפונקציות כאל ערכים ראשוניים במבנה התוכנית. המשמעות היא שניתן להעביר פונקציות כפרמטרים ולשלב אותן כחלק מהגדרת ערכים, מבלי להפעיל אותן מיידית.

באמצעות QUOTE ניתן גם להגדיר פרמטרים שישמשו את הפונקציה בעת הפעלתה, ומכאן לאפשר ביצוע של פעולות מתקדמות, דוגמת העברת פונקציות כפלט או קלט לפעולות אחרות, כל זאת תוך שמירה על מבנה קוד ברור ונוח לתחזוקה.

קושי גדול שהיה לי היה לשמר את משתני הקלט בהרכבה של פונקציות.  
פתרתי את זה באמצעות קריאה רקורסיבית על הארגומנטים של הפונקציה, בכדי למצוא את המקומות בהם נמצא משתני הקלט שנקראים בפונקציה פנימית.

## השוואת משתנים עם תוצאות פונקציות – פקודת JUMP\_EQUAL\_FUNCTION

המערכת תומכת בפקודת JUMP\_EQUAL\_FUNCTION, המאפשרת לבצע השוואת משתנים עם תוצאות של פונקציות. כאשר ערך משתנה שווה לתוצאה שמחזירה פונקציה, ניתן לבצע מעבר מותנה (קפיצה) במבנה התוכנית.

## קריאות פונקציה מקוננות

המערכת כוללת תמיכה מלאה בקריאות פונקציה מקוננות, כלומר, ניתן להעביר פונקציות כארגומנטים לפונקציות אחרות ולבנות שרשראות חישוב מורכבות.   
הלוגיקה שאיתה עשיתי את האלגוריתם לקריאה לפונקציות מקוננות היא לפי הנספח של אביעד.

## חישוב דרגה רקורסיבי

ביצעתי את חישוב הדרגה, בגלל שתי ההוראות החדשות, אלגוריתם לחישוב דרגה רקורסיבי, אשר מנתח את מורכבות התוכנית על פי אופן השימוש בפונקציות מקוננות והפרמטרים שלהן. בדרך זו ניתן להעריך את עומק ורמת המורכבות של קריאות הפונקציה, ולשלוט טוב יותר בתכנון ובתחזוקת הקוד, תוך שמירה על יעילות ויכולת הרחבה עתידית (נמצא בקובץ SprogramImpl).

## שימוש מתקדם בפקודות QUOTE ו-JUMP\_EQUAL\_FUNCTION

## מנגנון קומפוזיציה לפונקציות

המערכת מבוססת על מנגנון מתקדם לקומפוזיציה של פונקציות, כאשר כל פונקציה מוגדרת בשם ייחודי המאפשר זיהוי חד-משמעי במערכת. קיימת תמיכה בהעברת מספר פרמטרים בין פונקציות, כך שניתן להעביר מידע וגמישות בין תהליכים שונים בתוכנית. בנוסף, קיימת אפשרות לקריאות מקוננות, בהן פונקציות יכולות לקרוא לפונקציות אחרות תוך ניהול תחום הריצה והפרמטרים המועברים. המערכת תומכת ברקורסיה מלאה, ומאפשרת לפונקציה להפעיל את עצמה עם פרמטרים שונים ליצירת לולאות חישוב מתקדמות.

## חישוב התרחבות

במערכת מיושם אלגוריתם רקורסיבי לחישוב דרגה עבור קריאות פונקציה מקוננות. האלגוריתם מבצע ניתוח של גוף הפונקציה, כלומר בודק את רצף הפקודות המוגדר בפונקציה עצמה, וכן מבצע ניתוח על הארגומנטים ,כולל חישוב הדרגות של ארגומנטים מקוננים. בסופו של תהליך, מחושב הדרגה המקסימלית של התוכנית.

## חישוב מחזורים (Cycles)

לכל פקודה במערכת מוגדר מספר מחזורים, כאשר גם קריאות לפונקציה כוללות חישוב של מחזורי קריאה. המערכת מחשבת את השפעת ההתרחבות.  
כלומר, את השפעת קריאות הפונקציה והתרחבותן על סך כל מחזורי התוכנית. בעזרת אגרגציה מלאה מתבצע חישוב כולל של כל מחזורי התוכנית במטרה להעריך ביצועים וסיבוכיות.  
גם פה חישוב הסייקלים של QUOTE ו- JUMP EQUAL FUNCTION מתבצע עי חישוב של גוף הפונקציות בתוספת 5 או 6 (תלוי הוראה).

## הדגשת פקודות

המערכת תומכת במנגנון הדגשה מתקדם לפקודות, המבוסס על משתנים ועל תוויות.  
כך שניתן להדגיש פקודות לפי ערכי משתנים או לפי יעדי קפיצה ובקרת זרימה.   
ההדגשה היא בצבע צהוב, והיא מדגישה משתני קלט, עבודה, ופלט, בתוספת לייבלים.

## מערכת שרשרת היסטוריה

המערכת שומרת היסטוריה מלאה של ביצועי הפקודות, כולל מעקב אחר מצב משתנים בכל שלב בתהליך. קיימת אפשרות לעקוב אחר שרשראות קריאת פונקציה, ולהציג הדמיה חזותית של שרשראות הביצוע.

## מנגנון דיבוג

מערכת הדיבוג כוללת אפשרויות מתקדמות כמו עצירה בנקודה נוכחית, ביצוע פקודה אחת (Step Over) וחזרה לאחור בהיסטוריית ההרצה (Step Back). ניתן להמשיך את ההרצה מהנקודה הנוכחית, לקבוע נקודות עצירה (Breakpoints) בפקודות ספציפיות, ולבדוק ערכי משתנים בזמן אמת.

## הרצות חוזרות

קיימת תמיכה באיפוס מצב התוכנית להרצה חדשה, כולל אתחול ערכי משתנים, ניקוי היסטוריית ההרצה והחזרת רכיבי ממשק המשתמש למצב ראשוני.

## רכיבי ממשק משתמש וארכיטקטורת רכיבים

המערכת בנויה מארכיטקטורת רכיבים ברורה: התצוגה הראשית מהווה מכולה מרכזית המשלבת את כל הרכיבים הדרושים. רכיב הכותרת מאפשר טעינת קבצים, הצגת מדדי התקדמות ושליטה על פעולות. טבלת הפקודות מציגה את מבנה התוכנית בצורה טבלאית עם הדגשות והתרחבות. שרשרת ההיסטוריה מוצגת בצורה חזותית, לצד סטטיסטיקות ונתונים על הרצת התוכנית הנוכחית. רכיב הדיבאג מספק בקרה להרצה צעד אחר צעד ותצוגת משתנים.

## עיצוב רספונסיבי עם Scroll Panes

המערכת תומכת בגלילה אופקית לטבלאות פקודות רחבות ובגלילה אנכית לרשימות פקודות ארוכות. שינוי גודל דינמי מאפשר התאמה אוטומטית לפי תוכן, כולל שינוי רספונסיבי של רכיבים וסקיילינג דינמי בהתאם לגודל התוכן, תוך שמירה על גודל מינימלי והימנעות מהקטנה מוגזמת של רכיבים.

## שימוש בסיסי

1. לחץ על 'Load Program' ובחר קובץ XML של תוכנית S.
2. השתמש בטבלת הפקודות להצגת מבנה התוכנית.
3. בחר משתנים או תוויות להדגשה.
4. השתמש בכלי הדיבאג להרצה צעד אחר צעד.
5. הצג שרשראות היסטוריה וסטטיסטיקות לניתוח התנהגות התוכנית.