בשביל מימוש הקומפיילר השלם של חלק שלישי השתמשנו בשני החלקים הקודמים: המנתל הלקסיקלי מחלק 1 והמנתח התחבירי (עם מודיפיקציות מתאימות) מחלק 2.

השימוש היה בשפת c++ על ידי יצירת אובייקטים עם תכונות ומשתנים שנדרשים לצמתים השונות בעץ וכן הוספנו טבצאות סמלים רלוונטיות להחזקת משתנים, איפה הם קיימים, קישורים לפונקציות וכל מה שנדרש לכתיבת קוב הביניים.

נעשה שימוש ברגיסטרים עם תפקידים יעודיים לצורך קריאות וחזרות מפונקציות (RA שומר את כתובת הקפיצה בחזרה מפונקציה, FP הוא מצביע לשומת ההפעלה, RT שומר את ערך החזרה, SP מצביע לכתובת הבאה במחסנית) ושאר הרגיסטרים משמשים למשתנים השונים.

*נקודת הנחה היא שיש מספיק רגיסטרים לכמות המשתנים הנדרשת.

* בגלל שפקודות ASM כוללות פעולות שמכילות גם רגיסטרים של INT וגם של FLOAT (וכתובת במחסנית היא INT בלבד) יש לבצע התמרה של רגיסטר INT המכיל את הכתובת לרגיסטר FLOAT.

לצורך מימוש העץ הוגדרו שתי מחלקות: מחלקת סימבולים (SymNode) ומחלקת אסימונים (TokenNode) היורשות מאותה מחלקת אב. כל מחלקה מכילה את כל התכונות שנדרשות למימוש הצומת המתאימה כאשר נעשה שימוש רק בתכונות הנדרשות לצומת.

לצורך מימוש רשומות המשתנים יצרנו טבלה שתכיל את המשתנים הקיימים בכל סקופ וכל תא בטבלה מנהל את המשתנים וההקצאות השונות עבור הסקופ שרלוונטי אליו.

במהלך הריצה יש שימוש בשני סוגי משתנים:

1) משתנים זמניים אשר מוקצים לרגיסטר הפנוי הבא ברשימת הרגיסטרים הפנויים.

2) משתנים קבועים אשר דורשים מיפוי (גם כן בעזרת רשימת map) בין שם המשתנה למיקום שהוקצה במחסנית (היסט במחסנית).

לצורך מימוש הפונקציות בקבצים השונים יצרנו טבלה מטיפוס map אשר ממפה function name לfuncEntry (טיפוס של אובייקט שמכיל את פרטי הפונקציה הנדרשים).