



Greek++ Compiler

ΜΥΥ802 – ΜΕΤΑΦΡΑΣΤΕΣ

ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2024 – 2025

Κουτουλής Χρήστος (A.M.: 5064)
Κουτσονικολής Νικόλαος (A.M.: 5108)

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ	3
ΛΕΚΤΙΚΗ ΚΑΙ ΣΥΝΤΑΚΤΙΚΗ ΑΝΑΛΥΣΗ.....	4
Λεκτική ανάλυση	4
Συντακτική ανάλυση.....	8
Παράδειγμα.....	10
ΠΑΡΑΓΩΓΗ ΕΝΔΙΑΜΕΣΟΥ ΚΩΔΙΚΑ ΚΑΙ ΠΙΝΑΚΑ ΣΥΜΒΟΛΩΝ.....	12
Ενδιάμεσος κώδικας.....	12
Πίνακας Συμβόλων.....	14
Παραγωγή.....	18
Αναθέσεις:	18
Πράξεις:.....	19
IF-THEN:.....	21
WHILES:.....	25
FOR LOOPS:.....	27
ΚΛΗΣΕΙΣ ΣΥΝΑΡΤΗΣΕΩΝ ΚΑΙ ΔΙΑΔΙΚΑΣΙΩΝ:.....	29
ΠΑΡΑΓΩΓΗ ΤΕΛΙΚΟΥ ΚΩΔΙΚΑ	33
TESTS ΚΑΙ ΠΑΡΑΔΕΙΓΜΑΤΑ ΟΡΘΗΣ ΛΕΙΤΟΥΡΓΙΑΣ	44
TEST1	44
TEST2	50
TEST3	57
TEST DEFAULT INTERMEDIATE CODE.....	74

ΕΙΣΑΓΩΓΗ

Σε αυτήν την άσκηση θα υλοποιήσουμε έναν compiler για την γλώσσα Greek++. Οι προδιαγραφές της γλώσσας και η γραμματική της αναλύονται από τις εκφωνήσεις που μας δόθηκαν στο μάθημα. Σε αυτή την αναφορά, θα αναλύσουμε πως φτιάξαμε ένα python πρόγραμμα `compiler.py` που δέχεται αρχεία τύπου `.gr` και εκτελεί λεκτική και συντακτική ανάλυση, παραγωγή ενδιάμεσου κώδικα και πίνακα συμβόλων και καταλήγει με μετατροπή του κώδικα σε γλώσσα RISC-V assembly.

Η πορεία της εργασίας χωρίστηκε σε 3 στάδια:

- Λεκτική και συντακτική ανάλυση
- Παραγωγή ενδιάμεσου κώδικα και πίνακα συμβόλων
- Παραγωγή τελικού κώδικα

ΛΕΚΤΙΚΗ ΚΑΙ ΣΥΝΤΑΚΤΙΚΗ ΑΝΑΛΥΣΗ

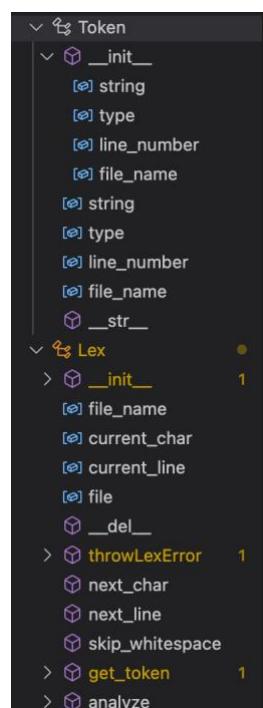
Λεκτική ανάλυση

Για την λεκτική ανάλυση χρησιμοποιείται η Lex κλάση. Όταν καλείται ανοίγει το .gr αρχείο, αρχικοποιεί τους μετρητές current_char και current_line και χρησιμοποιεί την next_char() για να διαβάσει τον πρώτο χαρακτήρα του αρχείου.

Χρησιμοποιεί την κλάση TokenFamily για να κατηγοριοποιήσει τα Tokens ανάλογα τον τύπο τους και χρησιμοποιεί τους πίνακες ALLOWED_CHARS, KEYWORDS, OPS_AND_SYMBOLS και WHITESPACES για να ελέγχει και να παράγει κατάλληλα Tokens.

To Token ορίζεται ως αντικείμενο με την κλάση Token:

```
class Token:  
  
    def __init__(self, string, type, line_number, file_name=None):  
        self.string = string  
        self.type = type  
        self.line_number = line_number  
        self.file_name = file_name  
  
    def __str__(self):  
        return f'{self.string}\ttype:{self.type.name}\tline: {self.line_number}'
```



Η λεκτική ανάλυση γίνεται μέσα στην συνάρτηση analyze() όπου όλα τα tokens γίνονται extracted από την get_token() και αποθηκεύονται σε μια λίστα μέχρι να εμφανιστεί Token τύπου EOF:

```
def analyze(self):  
    tokens = []  
  
    while True:  
        token = self.get_token()  
        if token.type == TokenFamily.EOF:  
            break  
        tokens.append(token)  
    print(tokens)
```

Figure 1 Token and Lex Class Outlines

Η συνάρτηση `get_token()` υλοποιεί την εξαγωγή των `Tokens`. Αρχικά ελέγχει για whitespaces και τα αγνοεί:

```
self.skip_whitespace()
```

Έπειτα, αν δεν διαβάσει χαρακτήρα επιστρέφει EOF:

```
if not self.current_char:
```

```
    return Token("", TokenFamily.EOF, self.current_line, self.file_name)
```

Μετά κάνει αναγνώριση αριθμών. Αναδρομικά όσο διαβάζει αριθμό τον προσθέτει στην λέξη και ελέγχει εάν ξεπερνάει τους 30 χαρακτήρες:

```
# AKERAIOI
if self.current_char.isdigit():
    word = self.current_char
    self.next_char()
    while self.current_char and self.current_char.isdigit():
        word += self.current_char
        self.next_char()
        if(len(word) >= MAX_WORD_SIZE):
            break
    # Tsekare an o epomenos xaraktiras einai psifio h whitespace
    if self.current_char and (self.current_char in ALLOWED_CHARS): #.isalpha() or self.current_char == '_'):
        word += self.current_char
        self.next_char()
        while self.current_char and (self.current_char in ALLOWED_CHARS): #.isalnum() or self.current_char == '_'):
            word += self.current_char
            self.next_char()
            self.throwLexError(INVALID_TOKEN_ERROR, self.current_line, word)
    return Token(word, TokenFamily.ERROR, self.current_line, self.file_name)
return Token(word, TokenFamily.NUMBER, self.current_line, self.file_name)
```

Υστερα κάνει έλεγχο για identifiers ή keywords. Αν η λέξη που διαβαστεί είναι μέσα στον πίνακα των KEYWORDS τότε το token είναι τύπου KEYWORD, αλλιώς είναι αναγνωριστικό:

```
# IDENTIFIERS KAI KEYWORDS
if self.current_char in ALLOWED_CHARS and self.current_char != '_': # Prevent identifiers from starting with underscore
    word = self.current_char
    self.next_char()
    while self.current_char and (self.current_char in ALLOWED_CHARS): #.isalnum() or self.current_char == '_'):
        word += self.current_char
        self.next_char()
        if(len(word) >= MAX_WORD_SIZE):
            break
    token_type = TokenFamily.KEYWORD if word in KEYWORDS else TokenFamily.IDENTIFIER
    return Token(word, token_type, self.current_line, self.file_name)
```

Έλεγχος για relational operators:

```
# RELATIONAL OPS: <, >, <=, =, >=, <>
if self.current_char in {'<', '>', '='}:
    word = self.current_char
    line = self.current_line
    next_word = self.next_char() # epomeni leksi
    if (word in {'<', '>'} and next_word == '=') or (word == '<' and next_word == '>'): # <=, >=
        word += next_word
        self.next_char()
return Token(word, TokenFamily.RELATIONAL_OPERATOR, line, self.file_name)
```

Έλεγχος για ανάθεση, όπου πετάει error αν το σύμβολο ':' είναι χωρίς το '=' γιατί δεν υποστηρίζεται σαν χαρακτήρας απ' την γλώσσα μόνο του:

```
# ANATHESI
if self.current_char == ':':
    word = self.current_char
    next_word = self.next_char()
    if next_word == '=':
        assignment = word + next_word
        self.next_char()
        return Token(assignment, TokenFamily.ASSIGNMENT, self.current_line, self.file_name)
    self.throwLexError(INVALID_ASSIGNMENT_ERROR, self.current_line, word+next_word)
    return Token(word+next_word, TokenFamily.ERROR, self.current_line, self.file_name) # Invalid ':'
```

Έλεγχος για πέρασμα με αναφορά και αριθμητικούς τελεστές:

```
# PERASMA ME ANAFORA
if self.current_char == '%':
    self.next_char()
    return Token("%", TokenFamily.PASSBYREFERENCE, self.current_line, self.file_name)

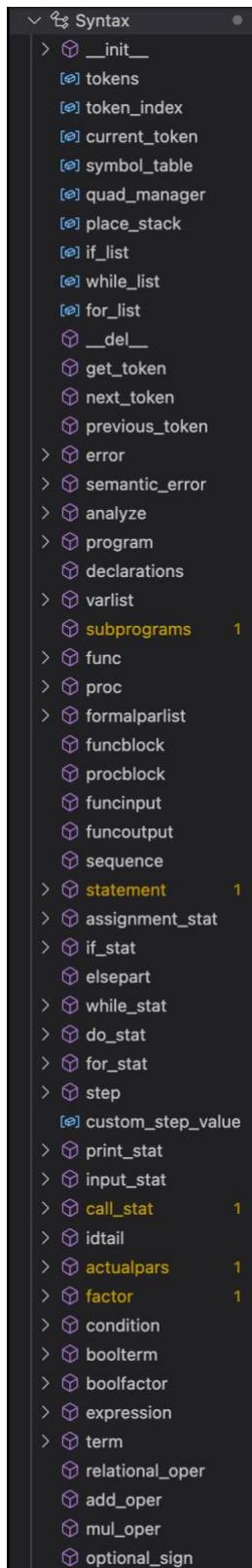
# ARITHMITIKA OPS KAI SYMVOLA
if self.current_char in OPS_AND_SYMBOLS:
    word = self.current_char
    token_type = OPS_AND_SYMBOLS.get(self.current_char)
    self.next_char()
    return Token(word, token_type, self.current_line, self.file_name)
```

Αγνόηση σχολίων που περικλείονται από {...} και error αν η get_token() δεν καταφέρει να εξάγει Token:

```
# SXOLIA
if self.current_char == '{':
    while self.current_char and self.current_char != '}':
        if self.current_char == "\n": # se periptwsi pou to sxolio einai pollaples grammes prepei na prosmetrountai
            self.skip_whitespace()
        self.next_char()
    self.next_char()
    return self.get_token()

# MI DEKTOI XARAKTIRES
error_char = self.current_char
self.next_char()
self.throwLexError(INVALID_TOKEN_ERROR, self.current_line, error_char)
return Token(error_char, TokenFamily.ERROR, self.current_line, self.file_name)
```

Συντακτική ανάλυση



Κατά τη συντακτική ανάλυση ελέγχεται εάν η ακολουθία των λεκτικών μονάδων (Tokens) που σχηματίζεται από τον λεκτικό αναλυτή, αποτελεί μία νόμιμη ακολουθία, με βάση τη γραμματική της γλώσσας. Όποια ακολουθία δεν αναγνωρίζεται από τη γραμματική, αποτελεί μη επιτρεπτό κώδικα και οδηγεί στον εντοπισμό συντακτικού σφάλματος. Έτσι, ο συντακτικός αναλυτής παίρνει ως είσοδο μία ακολουθία από Tokens, προϊόν της κλήσης της Lex.analyze().

```
# Main  
if __name__ == "__main__":  
    if len(sys.argv) != 2:  
        print("Error: Usage: 'python compiler.py <file_name>.gr'")  
        print("Exiting...")  
        sys.exit(1)  
  
    if not sys.argv[1].endswith('.gr'):  
        print("Error: File must be of .gr type")  
        print("Exiting...")  
        sys.exit(1)  
  
    filename = sys.argv[1]  
  
    lexer = Lex(filename)  
    token_list = lexer.analyze()  
    del lexer  
  
    parser = Syntax(token_list)  
    parser.analyze()
```

Ο συντακτικός αναλυτής υλοποιείται μέσω της κλάσης Syntax. Κάθε κανόνας της γραμματικής έχει υλοποιηθεί με μια αντίστοιχη μέθοδο.

Η αρχικοποίηση του συντακτικού αναλυτή γίνεται ως εξής:

```
class Syntax:  
    def __init__(self, tokens):  
        self.tokens = tokens  
        self.token_index = 0  
        self.current_token = self.tokens[self.token_index]  
        self.symbol_table = SymbolTable()  
        self.quad_manager = QuadManager()  
  
        # For expression evaluation  
        self.place_stack = []  
        # For control flow  
        self.if_list = []  
        self.while_list = []  
        self.for_list = []  
  
        print("\n-- Syntax Analyzer --")  
        print("Beginning syntactical analysis...\n")
```

Figure 2 Syntax Class Outline

Οι βοηθητικές συναρτήσεις για την συντακτική ανάλυση:

```
def get_token(self):
    self.token_index += 1
    if self.token_index < len(self.tokens):
        self.current_token = self.tokens[self.token_index]
    return self.current_token

def next_token(self):
    if self.token_index + 1 < len(self.tokens):
        return self.tokens[self.token_index + 1]
    return None

def previous_token(self):
    self.token_index -= 1
    self.current_token = self.tokens[self.token_index]
    return self.current_token

def error(self, message, expected=None):
    if expected:
        print(f"Syntax Error at line {self.current_token.line_number}: Expected '{expected}', but got '{self.current_token.string}'")
    else:
        print(f"Syntax Error at line {self.current_token.line_number}: {message}")
    sys.exit(1)

def semantic_error(self, message):
    print(f"Semantic Error at line {self.current_token.line_number}: {message}")
    sys.exit(1)

def analyze(self):
    for token in self.tokens:
        if token.type.name == "ERROR":
            self.error(f"Got invalid phrase '{token.string}'")

    self.program()
    print("--No errors--")
    self.quad_manager.print_intermediate_code()
```

- Η `get_token()` θα κάνει update το πεδίο `current_token` με το επόμενο `token` από την λίστα που πήρε είσοδο.
- Η `next_token()` θα επιστρέψει το επόμενο `token` χωρίς όμως να κάνει update το `current_token`. Την χρησιμοποιούμε για να κάνουμε προεπισκόπηση του επομένου `token` ώστε να ακολουθηθεί ο κατάλληλος κανόνας της γραμματικής.
- Η `previous_token()` θα κάνει backtracking επιστρέφοντας στο `token` πριν από το `current_token`.
- Η `error()` διαχειρίζεται τα συντακτικά λάθη στον κώδικα και εκτυπώνει το κατάλληλο μήνυμα πρώτου σταματήσει την συντακτική ανάλυση.
- Η `semantic_error()` διαχειρίζεται τα σημασιολογικά λάθη στον κώδικα όπως αδήλωτες μεταβλητές κλπ. Εκτυπώνει το κατάλληλο μήνυμα πρώτου σταματήσει την συντακτική ανάλυση.
- Η `analyze()` είναι η συνάρτηση που θα ξεκινήσει την συντακτική ανάλυση. Θα ελέγξει πρώτα την λίστα όλων των `tokens` και αφού βεβαιωθεί ότι δεν υπάρχει μη έγκυρο `token` θα καλέσει τον κανόνα `program()` που θα ξεκινήσει την αναδρομική κάθοδο στους κανόνες της γραμματικής.

Παράδειγμα

Οι μέθοδοι των κανόνων της γραμματικής είναι εμπλουτισμένοι και συνδυασμένοι με στοιχεία από την επόμενη φάση της παραγωγής ενδιάμεσου κώδικα και πίνακα συμβολών. Για αυτό θα αναλύσουμε αναλυτικά τις μεθόδους/κανόνες παρακάτω.

Θα παραθέσουμε όμως ένα ενδεικτικό παράδειγμα της λογικής μετάφρασης των κανόνων σε μεθόδους:

```
program           : 'πρόγραμμα' ID programblock
;
programblock     : declarations subprograms
                  'αρχή_προγράμματος' sequence 'τέλος_προγράμματος'
```

Ο κανόνας επιτάσσει πως ένα πρόγραμμα της Greek++ πρέπει να ξεκινάει με την keyword πρόγραμμα και να ακολουθεί ένα αναγνωριστικό του προγράμματος. Έπειτα πρέπει να ακολουθούν δηλώσεις μεταβλητών και υποπρογραμμάτων. Ο κορμός του προγράμματος πρέπει να περιέχεται μεταξύ των λέξεων αρχή_προγράμματος και τέλος_προγράμματος. Αυτή την επιταγή υλοποιούμε με την παρακάτω λογική:

```
def program(self):
    if self.current_token.string != "πρόγραμμα":
        self.error("Program should start with 'πρόγραμμα'")

    self.get_token()
    if self.current_token.type != TokenFamily.IDENTIFIER:
        self.error("'πρόγραμμα' should be followed by <PROGRAM_NAME> of type <IDENTIFIER>")

    self.declarations()
    next_token = self.next_token()
    if next_token and (next_token.string == "συνάρτηση" or next_token.string == "διαδικασία"):
        self.subprograms()

    self.get_token()
    if self.current_token.string != "αρχή_προγράμματος":
        self.error("'αρχή_προγράμματος' not found")

    self.sequence()

    self.get_token()
    if self.current_token.string != "τέλος_προγράμματος":
        self.error("τέλος_προγράμματος", "τέλος_προγράμματος")
```

Κρυμμένες με γκρι πλαίσιο είναι εντολές που αφορούν την επόμενη φάση.
Υπογραμμισμένες με κίτρινο είναι οι κλήσεις των κανόνων της γραμματικής.
Βλέπουμε πως εκτελούνται έλεγχοι για τον τύπο και περιεχόμενο των tokens
πριν προχωρήσουμε στην κλήση του επομένου κανόνα. Ας δούμε και τον
κανόνα της subprograms:

```
subprograms      :   ( func | proc )*
                      ;
def subprograms(self):
    if self.next_token().string == "συνάρτηση":
        self.func()
        self.subprograms()
        return
    elif self.next_token().string == "διαδικασία":
        self.proc()
        self.subprograms()
        return
    return
```

Καλεί την func () ή proc () ανάλογα με την keyword συνάρτηση ή
διαδικασία. Το * στον κανόνα δηλώνει πως μπορούμε να έχουμε ακολουθία
από συναρτήσεις και διαδικασίες. Για να το πετύχουμε αυτό καλούμε
αναδρομικά την subprograms (), η οποία θα επιστρέψει στην program ()
μόνο όταν δεν ακολουθεί κάποιο από τα 2 keywords.

ΠΑΡΑΓΩΓΗ ΕΝΔΙΑΜΕΣΟΥ ΚΩΔΙΚΑ ΚΑΙ ΠΙΝΑΚΑ ΣΥΜΒΟΛΩΝ

Ενδιάμεσος κώδικας

Ο ενδιάμεσος κώδικας είναι μία ενδιάμεση αναπαράσταση της αρχικής γλώσσας πριν φτάσει στην τελική της μορφή. Ένα πρόγραμμα συμβολισμένο στην ενδιάμεση γλώσσα αποτελείται από μία σειρά από τετράδες, οι οποίες είναι αριθμημένες έτσι ώστε σε κάθε τετράδα να μπορούμε να αναφερθούμε χρησιμοποιώντας τον αριθμό της ως ετικέτα. Η διαχείριση και παραγωγή των τετράδων υλοποιείται από την κλάση QuadManager. Η κλάση έχει πεδία μια λίστα από τις τετράδες (πλειάδες) quads, έναν μετρητή ετικετών, το όνομα του προγράμματος και έναν μετρητή προσωρινών μεταβλητών.

```
class QuadManager:
    def __init__(self):
        self.quads = []
        self.next_label = 1
        self.program_name = ""
        self.temp_counter = 0

    def gen_quad(self, op, arg1, arg2, result):
        #generate tetrada
        quad = (self.next_label, op, arg1, arg2, result)
        self.quads.append(quad)
        self.next_label += 1
        return self.next_label - 1

    def next_quad(self):
        #epistrofi etiketas tis epomenis tetradas
        return self.next_label

    def backpatch(self, list_of_quads, label):
        for quad_index in list_of_quads:
            if 0 < quad_index <= len(self.quads):
                quad_num, op, arg1, arg2, _ = self.quads[quad_index - 1]
                self.quads[quad_index - 1] = (quad_num, op, arg1, arg2, str(label))

    def merge_lists(self, list1, list2):
        return list1 + list2

    def make_list(self, quad_index):
        return [quad_index]
```

Δημιουργήθηκαν οι παρακάτω βοηθητικές συναρτήσεις:

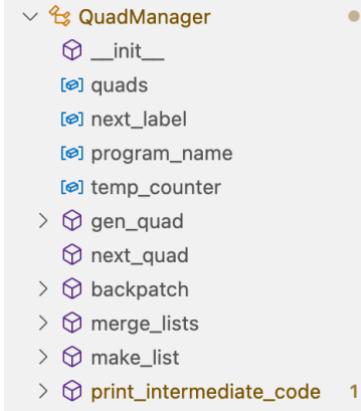


Figure 3 :QuadManager Outline

- Η `gen_quad()` δέχεται 4 ορίσματα:
 - `op`: ο τελεστής της τετράδας.
 - `arg1`: το πρώτο κατηγορημα της τετράδας.
 - `arg2`: το δεύτερο κατηγορημα της τετράδας.
 - `result`: εκεί που θα εκχωρηθεί το αποτέλεσμα.
- Η τετράδα/πλειάδα δημιουργείται και προστίθεται στην λίστα όλων των `quads`. Αυξάνεται επίσης και ο μετρητής των `labels`.

- Η `next_quad()` επιστρέφει την ετικέτα της επομένης τετράδας που θα δημιουργηθεί όταν κάνουμε κλήση της `next_quad()`
- Η `backpatch()` δέχεται μια λίστα από τετράδες και συμπληρώνει τα κενά πεδία στο `result` με το όρισμα `label`.
- Η `merge_list()` δέχεται 2 λίστες και παράγει την συνένωση αυτών.
- Η `make_list()` δέχεται ένα `index` και το επιστρέφει ως λίστα.

```
def print_intermediate_code(self, output_file=None):

    if output_file:
        with open(output_file, 'w', encoding='utf-8') as f:
            for quad_num, op, arg1, arg2, result in self.quads:
                # format arguments
                arg1 = arg1 if arg1 != "_" else "_"
                arg2 = arg2 if arg2 != "_" else "_"
                result = result if result != "_" else "_"
                f.write(f"{quad_num} : {op} , {arg1} , {arg2} , {result}\n")
```

Η `print_intermediate_code()` θα δημιουργήσει ένα αρχείο και θα γράψει τις τετράδες μέσα.

Πίνακας Συμβόλων

Στον πίνακα συμβόλων αποθηκεύουμε, κατά τη φάση της συντακτικής ανάλυσης και της παραγωγής ενδιάμεσου κώδικα, πληροφορίες σχετικά με τα αναγνωριστικά που χρησιμοποιούμε σε ένα πρόγραμμα: μεταβλητές, συναρτήσεις, παράμετροι, κλπ. Η πληροφορία αυτή χρησιμοποιείται στη φάση της σημασιολογικής ανάλυσης αλλά και της παραγωγής του τελικού κώδικα.

Υλοποιήσαμε 3 κλάσεις για την παραγωγή του πίνακα συμβόλων:

- 1) SymbolType: Κρατάει τους τύπους των συμβόλων όπως κρατούσε και η TokenFamily τους τύπους των Tokens.

```
class SymbolType(Enum):
    VARIABLE = 0
    FUNCTION = auto()
    PROCEDURE = auto()
    PARAMETER = auto()
    TEMPORARY = auto()
```

- 2) Symbol: Δημιουργία αντικειμένου για τα σύμβολα.

```
class Symbol:
    def __init__(self, name, symbol_type, scope="global", offset=0, parameter_mode=None):
        self.name = name
        self.symbol_type = symbol_type
        self.scope = scope
        self.offset = offset
        self.parameter_mode = parameter_mode
        self.parameters = []
        self.nesting_level = 0
        self.entry_label = 0
        self.frame_length = 0
```

- i) name: το όνομα του συμβόλου.
- ii) symbol_type: ο τύπος του συμβόλου (μεταβλητή, συνάρτηση, διαδικασία, προσωρινή μεταβλητή).
- iii) scope: η εμβέλεια του συμβόλου. Είτε θα είναι global είτε θα ανήκει σε κάποια συνάρτηση ή διαδικασία.
- iv) offset: Η θέση του συμβόλου στο activation record σε bytes
- v) parameter_mode: Για αν η μεταβλητή περνιέται με αναφορά ή τιμή
- vi) parameters: Λίστα από τις τοπικές μεταβλητές μιας συνάρτησης ή διαδικασίας.
- vii) nesting_level: για τις εμφωλιασμένες συναρτήσεις ή διαδικασίες.
- viii) entry_label: η ετικέτα από όπου ξεκινάει μια συνάρτηση ή διαδικασία.
- ix) frame_length: Το συνολικό μέγεθος του activation record μιας συνάρτησης.

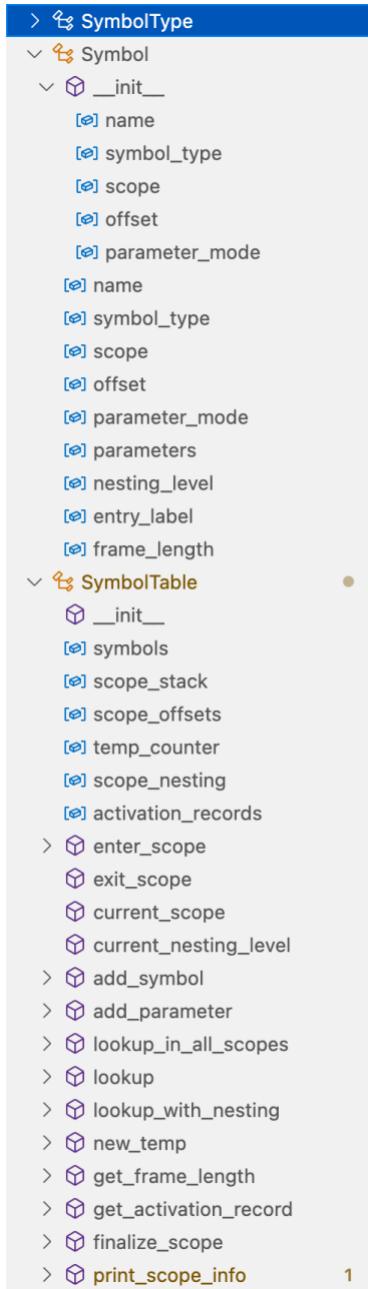


Figure 4 SymbolType, Symbol, SymbolTable Outline

3) SymbolTable:

```

class SymbolTable:
    def __init__(self):
        self.symbols = {}
        self.scope_stack = ["global"]
        self.scope_offsets = {"global": 12} # ksekinaei apo 12 bytes
        self.temp_counter = 0
        self.scope_nesting = {"global": 0}
        self.activation_records = {
            "global": {
                'parent': None,
                'frame_length': 12,
                'parameters': [],
                'local_vars': [],
                'temp_vars': []
            }
        }

```

Symbols: Η λίστα με τα σύμβολα

scope_stack: Η στοίβα των εμβελειών

scope_offsets: Τα offsets κάθε εμβέλειας

temp_counter: ο μετρητής προσωρινών μεταβλητών

scope_nesting: κρατά σε ποιο επίπεδο βρίσκεται
κάθε εμβέλεια

activation_records: στοιχεία για το activation record
και την διάταξη της μνήμης της κάθε εμβέλειας.

```

def enter_scope(self, scope_name):
    parent_scope = self.current_scope
    self.scope_stack.append(scope_name)
    parent_nesting = self.scope_nesting.get(parent_scope, 0)
    self.scope_nesting[scope_name] = parent_nesting + 1
    self.scope_offsets[scope_name] = 12

    # create activation record for this scope
    self.activation_records[scope_name] = {
        'parent': parent_scope,
        'frame_length': 12,
        'parameters': [],
        'local_vars': [],
        'temp_vars': []
    }

```

Κρατάει το parent scope και προσθέτει το νέο scope
στην στοίβα. Αυξάνει το nesting_level και
αρχικοποιει το offset στα 12 bytes. Επίσης,
δημιουργεί το activation record της εμβέλειας.

```

def exit_scope(self):
    if len(self.scope_stack) > 1:
        return self.scope_stack.pop()
    return "global"

```

Επιστρέφει το προηγούμενο scope χωρίς να βγαίνει ποτέ από το global.

```

def current_scope(self):
    return self.scope_stack[-1]

```

Επιστρέφει το τελευταίο στοιχείο του scope_stack δηλαδή το scope όπου βρισκόμαστε.

```

def current_nesting_level(self):
    return self.scope_nesting.get(self.current_scope, 0)

```

Επιστρέφει το επίπεδο nesting που βρισκόμαστε.

```

def add_symbol(self, name, symbol_type, parameter_mode=None):
    current_scope = self.current_scope()
    key = (name, current_scope)

    if key in self.symbols:
        return False

    current_offset = self.scope_offsets.get(current_scope, 12)
    symbol = Symbol(name, symbol_type, current_scope, current_offset, parameter_mode)
    symbol.nesting_level = self.current_nesting_level()

    self.symbols[key] = symbol
    self.scope_offsets[current_scope] = current_offset + 4 # κατε symbol 4 bytes

    #enimerosi activation record
    if current_scope in self.activation_records:
        ar = self.activation_records[current_scope]
        ar['frame_length'] += 4

        if symbol_type == SymbolType.PARAMETER:
            ar['parameters'].append(name)
        elif symbol_type == SymbolType.TEMPORARY:
            ar['temp_vars'].append(name)
        elif symbol_type == SymbolType.VARIABLE:
            ar['local_vars'].append(name)

    if symbol_type in [SymbolType.FUNCTION, SymbolType.PROCEDURE]:
        symbol.frame_length = 12

    return True

```

H add_symbol ελέγχει για διπλότυπα σύμβολα, υπολογίζει το offset, δημιουργεί και αποθηκεύει το σύμβολο.

H lookup_in_all_scopes αναζητά σε όλες τις εμβέλειες για το σύμβολο που ψάχνουμε. H new_temp επιστρέφει νέα προσωρινή μεταβλητή και την προσθέτει στον πίνακα συμβολών.

Παραγωγή

Αναθέσεις:

```
def assignment_stat(self):
    self.get_token()
    if self.current_token.type != TokenFamily.IDENTIFIER:
        self.error(f"Expected type {TokenFamily.IDENTIFIER.name}")

    target_var = self.current_token.string

    if not self.symbol_table.lookup(target_var) and not self.symbol_table.lookup_with_nesting(target_var):
        self.error(f"Undeclared variable '{target_var}'")

    self.get_token()
    if self.current_token.type != TokenFamily.ASSIGNMENT:
        self.error(":=", ";=")

    result = self.expression()
    self.quad_manager.gen_quad(":=", result, "_", target_var)

    if target_var == self.symbol_table.current_scope:
        func_symbol = self.symbol_table.lookup_in_all_scopes(target_var)
        if func_symbol and func_symbol.symbol_type == SymbolType.FUNCTION:
            self.quad_manager.gen_quad(":=", result, "_", "@RET")
```

πρόγραμμα example

δήλωση x, y

αρχή_προγράμματος

x := 5;

y := x

τέλος_προγράμματος

1 : begin_block , example , _ , _

2 : := , 5 , _ , x

3 : := , x , _ , y

4 : halt , _ , _ , _

5 : end_block , example , _ , _

Global Scope:

Name: x, Type: VARIABLE, Offset: 12

Name: y, Type: VARIABLE, Offset: 16

Πράξεις:

```
def expression(self):
    # Χειρισμός optional sign
    sign = None
    if (self.next_token().string == "+") or (self.next_token().string == "-"):
        sign = self.add_oper()

    # Κρατάω αποτέλεσμα από term
    term_result = self.term()

    # Επεξεργασία αρνητικού sign αν υπάρχει
    if sign == "-":
        # Δημιουργία temp
        neg_temp = self.symbol_table.new_temp()
        self.quad_manager.gen_quad("-", "0", term_result, neg_temp)
        term_result = neg_temp

    # Επεξεργασία περισσότερων όρων
    while (self.next_token().string == "+") or (self.next_token().string == "-"):
        op = self.add_oper()
        next_term = self.term()

        # Δημιουργία temp για το αποτέλεσμα
        result_temp = self.symbol_table.new_temp()

        # Δημιουργία τετράδας
        self.quad_manager.gen_quad(op, term_result, next_term, result_temp)

        # Ανανέωση του τρέχοντος αποτελέσματος
        term_result = result_temp

    return term_result

def term(self):
    # Κρατάω το αποτέλεσμα του factor
    factor_result = self.factor()

    # Επεξεργασία περισσότερων παραγόντων
    while (self.next_token().string == "*") or (self.next_token().string == "/"):
        op = self.mul_oper()
        next_factor = self.factor()

        # Δημιουργία temp για το αποτέλεσμα
        result_temp = self.symbol_table.new_temp()

        # Δημιουργία τετράδας
        self.quad_manager.gen_quad(op, factor_result, next_factor, result_temp)

        # Ανανέωση του τρέχοντος αποτελέσματος
        factor_result = result_temp

    return factor_result
```
def factor(self):
 self.get_token()
 if self.current_token.type == TokenFamily.NUMBER:
 if int(self.current_token.string) < ACCEPTED_NUMBER_RANGE[0] or int(self.current_token.string) > ACCEPTED_NUMBER_RANGE[1]:
 self.syntax_error(f"Got number out of range '{self.current_token.string}'")
 return self.current_token.string
 elif self.current_token.string == "(":
 expr_result = self.expression()
 self.get_token()
 if self.current_token.string != ")":
 self.syntax_error(")")
 return expr_result
 elif self.current_token.type == TokenFamily.IDENTIFIER:
 var_name = self.current_token.string
 func_call_result = self.idetail(var_name)
 if func_call_result is not None:
 return func_call_result
 else:
 # Check if variable exists in current scope or outer scopes
 if not self.symbol_table.lookup(var_name) and not self.symbol_table.lookup_with_nesting(var_name):
 self.syntax_error(f"Undeclared variable '{var_name}'")
 return var_name
 else:
 self.syntax_error("ID or (Expression) or NUMBER")
 return None
```

πρόγραμμα example

δήλωση a, b, c

αρχή\_προγράμματος

a := 10;

b := 5;

c := a + b \* 2

τέλος\_προγράμματος

1 : begin\_block , example , \_ , \_

2 ::= , 10 , \_ , a

3 ::= , 5 , \_ , b

4 : \* , b , 2 , T@0

5 : + , a , T@0 , T@1

6 ::= , T@1 , \_ , c

7 : halt , \_ , \_ , \_

8 : end\_block , example , \_ , \_

Global Scope:

Name: a, Type: VARIABLE, Offset: 12

Name: b, Type: VARIABLE, Offset: 16

Name: c, Type: VARIABLE, Offset: 20

Name: T@0, Type: TEMPORARY, Offset: 24

Name: T@1, Type: TEMPORARY, Offset: 28

## IF-THEN:

```
def if_stat(self):
 self.get_token()
 if self.current_token.string != "εάν":
 self.error("εάν", "εάν")

 # Κρατάω αποτελέσματα συνθηκών και true/false jump listes
 true_list, false_list = self.condition()

 self.get_token()
 if self.current_token.string != "τότε":
 self.error("τότε", "τότε")

 # Εκτελώ backpatch της true list στην τρέχουσα τετράδα
 self.quad_manager.backpatch(true_list, self.quad_manager.next_quad())

 self.sequence()

 # Δημιουργία jump τετράδας για το τέλος της if
 jump_quad = self.quad_manager.gen_quad("jump", "_", "_", "_")
 end_if_list = self.quad_manager.make_list(jump_quad)

 # Εκτελώ backpatch της false list στην τρέχουσα τετράδα
 self.quad_manager.backpatch(false_list, self.quad_manager.next_quad())

 self.elsepart()

 # Εκτελώ backpatch της end_if_list στην τρέχουσα τετράδα
 self.quad_manager.backpatch(end_if_list, self.quad_manager.next_quad())

 self.get_token()
 if self.current_token.string != "εάν_τέλος":
 self.error("εάν_τέλος", "εάν_τέλος")
```

```

def condition(self):
 # Krataw boolterm apotelesma kai true/false jump listes
 true_list, false_list = self.boolterm()

 # Epeksergasia OR operator
 while self.next_token().string == "ή":
 self.get_token()

 # Ektelw backpatch tis false list stin epomeni tetrada
 self.quad_manager.backpatch(false_list, self.quad_manager.next_quad())

 # Krataw epomeno boolterm apotelesma kai true/false jump listes
 next_true_list, next_false_list = self.boolterm()

 # Sinenwsi twn true lists
 true_list = self.quad_manager.merge_lists(true_list, next_true_list)

 # H nea false list einai h epomeni false list
 false_list = next_false_list

 # Prosarmogi twn jump stoxwn gia tis tetrades sygkrisis
 for quad_index in true_list:
 if 0 < quad_index <= len(self.quad_manager.quads):
 quad_num, op, arg1, arg2, _ = self.quad_manager.quads[quad_index - 1]
 if op in {'<', '>', '<=', '>=', '=', '≠'}:
 # Gia tous comparison operators, to target einai h epomeni tetrada
 self.quad_manager.quads[quad_index - 1] = (quad_num, op, arg1, arg2, str(self.quad_manager.next_quad()))

 return true_list, false_list

def boolterm(self):
 # Krataw boolfactor apotelesma kai true/false jump listes
 true_list, false_list = self.boolfactor()

 # Epeksergasia AND operator
 while self.next_token().string == "και":
 self.get_token()

 # Ektelw backpatch tis true list stin epomeni tetrada
 self.quad_manager.backpatch(true_list, self.quad_manager.next_quad())

 # Krataw epomeno boolfactor apotelesma kai true/false jump listes
 next_true_list, next_false_list = self.boolfactor()

 # Sinenwsi twn false lists
 false_list = self.quad_manager.merge_lists(false_list, next_false_list)

 # H nea true list einai h epomeni true list
 true_list = next_true_list

 return true_list, false_list

```

```

def boolfactor(self):
 # Elegxos gia NOT operator
 not_flag = False
 if self.next_token().string == "όχι":
 self.get_token()
 not_flag = True

 # Xeirismos synthikis se parenthesi
 if self.next_token().string == "[":
 self.get_token()
 true_list, false_list = self.condition()
 self.get_token()
 if self.current_token.string != "]":
 self.syntax_error("]")

 # Ean yparxei NOT, kanw swap ta true kai false lists
 if not_flag:
 true_list, false_list = false_list, true_list

 return true_list, false_list

 # Xeirismos relational expression
 left_expr = self.expression()
 rel_op = self.relational_oper()
 right_expr = self.expression()

 # Dhamiourgia tetradas sygkrisis
 comp_quad = self.quad_manager.gen_quad(rel_op, left_expr, right_expr, "_")

 # Dhamiourgia listwn gia true kai false periptwseis
 if not_flag:
 # An yparxei NOT, kanw swap ta true kai false lists
 false_list = self.quad_manager.make_list(comp_quad)
 true_list = self.quad_manager.make_list(self.quad_manager.next_quad())
 # Prosthetiki jump gia na skiparei to false case
 self.quad_manager.gen_quad("jump", "_", "_", "_")
 else:
 true_list = self.quad_manager.make_list(comp_quad)
 false_list = self.quad_manager.make_list(self.quad_manager.next_quad())
 # Prothetiki jump gia na skiparei to true case
 self.quad_manager.gen_quad("jump", "_", "_", "_")

 return true_list, false_list

```

πρόγραμμα

δήλωση x, y

αρχή\_προγράμματος

x := 10;

εάν x > 5 τότε

y := 1

αλλιώς

y := 0

εάν\_τέλος

τέλος\_προγράμματος

1 : begin\_block , example , , ,

2 ::= , 10 , , , x

3 : > , x , 5 , 6

4 : jump , , , , 8

5 : jump , , , , 10

6 ::= , 1 , , , y

7 : jump , , , , 9

8 ::= , 0 , , , y

9 : halt , , , ,

10 : end\_block , example , , ,

## WHILES:

```
def while_stat(self):
 self.get_token()
 if self.current_token.string != "όσο":
 self.error("όσο", "όσο")

 # Κρατάω αρχή συνθηκών για να επιστρέψει στην επανάληψη
 while_start = self.quad_manager.next_quad()

 # Κρατάω αποτελέσματα συνθηκών και true/false jump listes
 true_list, false_list = self.condition()

 self.get_token()
 if self.current_token.string != "επανάλαβε":
 self.error("επανάλαβε", "επανάλαβε")

 # Εκτελώ backpatch της true list στην τρέχουσα τετράδα (loop body)
 self.quad_manager.backpatch(true_list, self.quad_manager.next_quad())

 self.sequence()

 # Παραγωγή jump τετράδας για να επιστρέψει στη συνθήκη
 self.quad_manager.gen_quad("jump", "_", "_", while_start)

 # Εκτελώ backpatch της false list στην επόμενη τετράδα (exit loop)
 self.quad_manager.backpatch(false_list, self.quad_manager.next_quad())

 self.get_token()
 if self.current_token.string != "όσο_τέλος":
 self.error("όσο_τέλος", "όσο_τέλος")
```

πρόγραμμα example

δήλωση i, sum

αρχή\_προγράμματος

i := 1;

sum := 0;

όσο i <= 5 επανάλαβε

sum := sum + i;

i := i + 1

όσο\_τέλος

τέλος\_προγράμματος

1 : begin\_block , example , \_ , \_

2 : := , 1 , \_ , i

3 : := , 0 , \_ , sum

4 : <= , i , 5 , 7

5 : jump , \_ , \_ , 12

6 : jump , \_ , \_ , 4

7 : + , sum , i , T@0

8 : := , T@0 , \_ , sum

9 : + , i , 1 , T@1

10 : := , T@1 , \_ , i

11 : jump , \_ , \_ , 4

12 : halt , \_ , \_ , \_

13 : end\_block , example , \_ , \_

## FOR LOOPS:

```
def for_stat(self):
 self.get_token()
 if self.current_token.string != "για":
 self.syntax_error("για")

 self.get_token()
 if self.current_token.type != TokenFamily.IDENTIFIER:
 self.syntax_error(f"Expected type {TokenFamily.IDENTIFIER.name}")

 counter_var = self.current_token.string

 if not self.symbol_table.lookup(counter_var) and not self.symbol_table.lookup_with_nesting(counter_var):
 self.syntax_error(f"Undeclared variable '{counter_var}'")

 self.get_token()
 if self.current_token.type != TokenFamily.ASSIGNMENT:
 self.syntax_error(f"Expected type {TokenFamily.ASSIGNMENT.name}")

 initial_value = self.expression()
 self.quad_manager.gen_quad(":=", initial_value, "_", counter_var)

 self.get_token()
 if self.current_token.string != "έως":
 self.syntax_error("έως")

 final_value = self.expression()
 final_temp = self.symbol_table.new_temp()
 self.quad_manager.gen_quad(":=", final_value, "_", final_temp)

 step_value = "1"
 self.step()

 if hasattr(self, 'custom_step_value'):
 step_value = self.custom_step_value
 delattr(self, 'custom_step_value')

 step_temp = self.symbol_table.new_temp()
 self.quad_manager.gen_quad(":=", step_value, "_", step_temp)

 for_start = self.quad_manager.next_quad()
 loop_body_quad = self.quad_manager.next_quad() + 2

 if step_value.startswith("-"):
 self.quad_manager.gen_quad(">=", counter_var, final_temp, str(loop_body_quad))

 else:
 self.quad_manager.gen_quad("<=", counter_var, final_temp, str(loop_body_quad))

 exit_quad = self.quad_manager.gen_quad("jump", "_", "_", "_")
 false_list = self.quad_manager.make_list(exit_quad)

 self.get_token()
 if self.current_token.string != "επανάλαβε":
 self.syntax_error("επανάλαβε")

 self.sequence()

 inc_temp = self.symbol_table.new_temp()
 self.quad_manager.gen_quad("+", counter_var, step_temp, inc_temp)
 self.quad_manager.gen_quad(":=", inc_temp, "_", counter_var)

 self.quad_manager.gen_quad("jump", "_", "_", for_start)
 self.quad_manager.backpatch(false_list, self.quad_manager.next_quad())

 self.get_token()
 if self.current_token.string != "για_τέλος":
 self.syntax_error("για_τέλος")
```

πρόγραμμα example

δήλωση i, sum

αρχή\_προγράμματος

sum := 0;

για i := 1 έως 5 επανάλαβε

    sum := sum + i

για\_τέλος

τέλος\_προγράμματος

1 : begin\_block , example , \_ , \_

2 ::= , 0 , \_ , sum

3 ::= , 1 , \_ , i

4 ::= , 5 , \_ , T@0

5 ::= , 1 , \_ , T@1

6 :<= , i , T@0 , 9

7 : jump , \_ , \_ , 14

8 : jump , \_ , \_ , 6

9 : + , sum , i , T@2

10 ::= , T@2 , \_ , sum

11 : + , i , T@1 , T@3

12 ::= , T@3 , \_ , i

13 : jump , \_ , \_ , 6

14 : halt , \_ , \_ , \_

15 : end\_block , example , \_ , \_

## ΚΛΗΣΕΙΣ ΣΥΝΑΡΤΗΣΕΩΝ ΚΑΙ ΔΙΑΔΙΚΑΣΙΩΝ:

```
def func(self):
 self.get_token()
 if self.current_token.string != "συνάρτηση":
 self.error("συνάρτηση", "συνάρτηση")

 self.get_token()
 if self.current_token.type != TokenFamily.IDENTIFIER:
 self.error("'συνάρτηση' should be followed by <FUNCTION_NAME> of type <IDENTIFIER>")

 # Προσθήκη συνάρτησης στον πίνακα συμβόλων
 func_name = self.current_token.string
 self.symbol_table.add_symbol(func_name, SymbolType.FUNCTION)

 # Ρύθμιση nesting level
 func_symbol = self.symbol_table.lookup(func_name)
 func_symbol.nesting_level = self.symbol_table.current_nesting_level()

 # Είσοδος σε νέα εμβέλεια
 self.symbol_table.enter_scope(func_name)

 # Παραγωγή begin_block τετράδας
 entry_label = self.quad_manager.next_quad()
 func_symbol.entry_label = entry_label
 self.quad_manager.gen_quad("begin_block", func_name, "_", "_")

 self.get_token()
 if self.current_token.string != "(":
 self.error("(", "(")

 self.formalparlist()

 self.get_token()
 if self.current_token.string != ")":
 self.error(")", ")"

 self.funcblock()

 # Οριστικοποίηση εμβέλειας
 self.symbol_table.finalize_scope(func_name)

 # Ρύθμιση frame length
 func_symbol.frame_length = self.symbol_table.get_frame_length(func_name)

 # Παραγωγή end_block τετράδας
 self.quad_manager.gen_quad("end_block", func_name, "_", "_")

 # Έξοδος από εμβέλεια
 self.symbol_table.exit_scope()
```

```

def call_stat(self):
 self.get_token()
 if self.current_token.string != "ΕΚΤΈΛΕΣΕ":
 self.syntax_error("ΕΚΤΈΛΕΣΕ")

 self.get_token()
 if self.current_token.type != TokenFamily.IDENTIFIER:
 self.syntax_error("'ΕΚΤΈΛΕΣΕ' should be followed by type <IDENTIFIER>'")

 proc_name = self.current_token.string

 # Έλεγχος αν η διαδικασία υπάρχει στον πίνακα συμβόλων
 proc_symbol = self.symbol_table.lookup(proc_name) or self.symbol_table.lookup_with_nesting(proc_name)
 if not proc_symbol or proc_symbol.symbol_type not in [SymbolType.PROCEDURE, SymbolType.FUNCTION]:
 self.syntax_error(f"Undefined procedure/function '{proc_name}'")

 # Λίστα παραμέτρων
 param_list = []

 # Έλεγχος αν υπάρχουν παράμετροι
 if self.next_token().string == "(":
 self.get_token() # κατανάλωση '('

 # Ανάλυση παραμέτρων
 if self.next_token().string != ")":
 while True:
 # Χειρισμός παραμέτρων με αναφορά
 if self.next_token().type == TokenFamily.PASSBYREFERENCE:
 self.get_token() # κατανάλωση '%'
 self.get_token() # λήψη ονόματος παραμέτρου
 if self.current_token.type != TokenFamily.IDENTIFIER:
 self.syntax_error("Expected identifier after '%'")
 param_list.append(("REF", self.current_token.string))
 else:
 # Χειρισμός παραμέτρων με τιμή
 param_value = self.expression()
 param_list.append(("CV", param_value))

 if self.next_token().string != ",":
 break
 self.get_token() # κατανάλωση ','

 self.get_token() # κατανάλωση ')'

 # Έλεγχος αριθμού παραμέτρων
 if len(param_list) != len(proc_symbol.parameters):
 self.semantic_error(f"Function/procedure '{proc_name}' expects {len(proc_symbol.parameters)} parameters but got {len(param_list)}")

 # Παραγωγή τετράδων παραμέτρων
 for mode, param in param_list:
 self.quad_manager.gen_quad("par", param, mode, "_")

 # Εάν είναι συνάρτηση δημιουργία temp για το αποτέλεσμα
 if proc_symbol and proc_symbol.symbol_type == SymbolType.FUNCTION:
 result_temp = self.symbol_table.new_temp()
 self.quad_manager.gen_quad("par", result_temp, "RET", "_")

 # Παραγωγή τετράδας κλήσης
 self.quad_manager.gen_quad("call", proc_name, "_", "_")

 # Εάν είναι συνάρτηση επιστροφή του temp με το αποτέλεσμα
 if proc_symbol and proc_symbol.symbol_type == SymbolType.FUNCTION:
 return result_temp

```

πρόγραμμα example

δήλωση x

διαδικασία print\_number(num)

διαπροσωπεία

είσοδος num

αρχή\_διαδικασίας

γράψε num

τέλος\_διαδικασίας

αρχή\_προγράμματος

x := 42;

εκτέλεσε print\_number(x)

τέλος\_προγράμματος

1 : jump ,\_,\_,5

2 : begin\_block , print\_number ,\_,\_

3 : out , num ,\_,\_

4 : end\_block , print\_number ,\_,\_

5 : begin\_block , example ,\_,\_

6 : := , 42 ,\_,x

7 : par , x , CV ,\_

8 : call , print\_number ,\_,\_

9 : halt ,\_,\_,\_

10 : end\_block , example ,\_,\_

πρόγραμμα example

δήλωση result

συνάρτηση add(a, b)

διαπροσωπεία

είσοδος a, b

έξοδος add

αρχή\_συνάρτησης

add := a + b

τέλος\_συνάρτησης

αρχή\_προγράμματος

result := add(5, 3)

τέλος\_προγράμματος

1 : jump ,\_,\_,7

2 : begin\_block , add ,\_,\_

3 : + , a , b , T@0

4 : := , T@0 ,\_, add

5 : := , T@0 ,\_, @RET

6 : end\_block , add ,\_,\_

7 : begin\_block , example ,\_,\_

8 : par , 5 , CV ,\_

9 : par , 3 , CV ,\_

10 : par , T@1 , RET ,\_

11 : call , add ,\_,\_

12 : := , T@1 ,\_, result

13 : halt ,\_,\_,\_

14 : end\_block , example ,\_,\_

Global Scope:

Name: result, Type: VARIABLE, Offset: 12

Name: add, Type: FUNCTION, Entry Label: 2, Frame Length: 20

Add Scope:

Name: a, Type: PARAMETER, Parameter Mode: CV, Offset: 12

Name: b, Type: PARAMETER, Parameter Mode: CV, Offset: 16

## READ/WRITE:

```

def print_stat(self):
 self.get_token()
 if self.current_token.string != "γράψε":
 self.syntax_error("γράψε")

 output_value = self.expression()

 # Παραγωγή τετράδας εκτύπωσης
 self.quad_manager.gen_quad("out", output_value, "_", "_")

def input_stat(self):
 self.get_token()
 if self.current_token.string != "διάβασε":
 self.syntax_error("διάβασε")

 self.get_token()
 if self.current_token.type != TokenFamily.IDENTIFIER:
 self.syntax_error("'διάβασε' should be followed by type <IDENTIFIER>'")

 var_name = self.current_token.string

 # Έλεγχος αν η μεταβλητή υπάρχει στον πίνακα συμβόλων
 if not self.symbol_table.lookup(var_name) and not self.symbol_table.lookup_with_nesting(var_name):
 self.syntax_error(f"Undeclared variable '{var_name}'")

 # Παραγωγή τετράδας εισόδου
 self.quad_manager.gen_quad("in", var_name, "_", "_")

```

πρόγραμμα example  
δήλωση x, y  
αρχή\_προγράμματος  
διάβασε x;  
διάβασε y  
τέλος\_προγράμματος

1 : begin\_block , example , \_ , \_  
2 : in , x , \_ , \_  
3 : in , y , \_ , \_  
4 : halt , \_ , \_ , \_  
5 : end\_block , example , \_ , \_

πρόγραμμα example  
δήλωση x, y  
αρχή\_προγράμματος  
x := 10;  
y := 20;  
γράψε x;  
γράψε x + y  
τέλος\_προγράμματος

1 : begin\_block , example , \_ , \_  
2 : := , 10 , \_ , x  
3 : := , 20 , \_ , y  
4 : out , x , \_ , \_  
5 : + , x , y , T@0  
6 : out , T@0 , \_ , \_  
7 : halt , \_ , \_ , \_  
8 : end\_block , example , \_ , \_

# ΠΑΡΑΓΩΓΗ ΤΕΛΙΚΟΥ ΚΩΔΙΚΑ

Η παραγωγή τελικού κώδικα υλοποιείται από την κλάση `RISCVGenerator`, η οποία μετατρέπει τις τετράδες ενδιάμεσου κώδικα σε RISC-V assembly κώδικα.

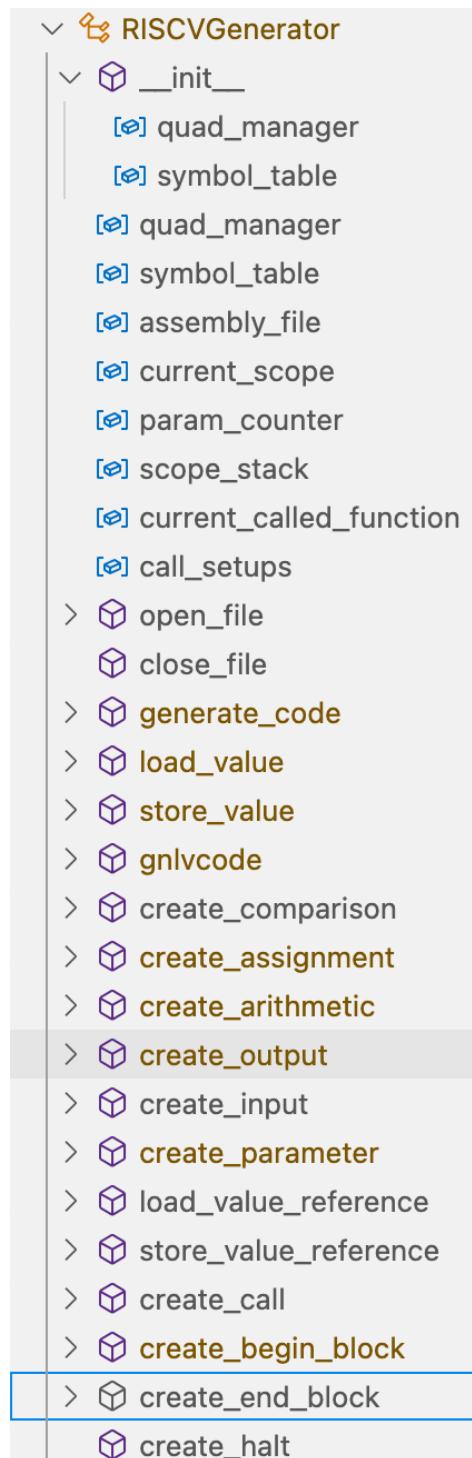


Figure 5 Class `RISCVGenerator` Outline

```
def __init__(self, quad_manager, symbol_table):
 self.quad_manager = quad_manager
 self.symbol_table = symbol_table
 self.assembly_file = None
 self.current_scope = "global"
 self.param_counter = -1
 self.scope_stack = ["global"]
 self.current_called_function = None
 self.call_setups = {}
```

- `quad_manager`: Αναφορά στον διαχειριστή τετράδων
- `symbol_table`: Αναφορά στον πίνακα συμβόλων
- `assembly_file`: το assembly αρχείο
- `current_scope`: Τρέχουσα εμβέλεια (ξεκινά από "global")
- `param_counter`: Μετρητής παραμέτρων (-1 = αδρανής,  $>=0$  = ενεργός)
- `scope_stack`: Stack για διαχείριση nested scopes
- `current_called_function`: Η συνάρτηση που καλείται
- `call_setups`: Dictionary για αντιστοίχιση call labels με συναρτήσεις

```
def open_file(self, filename):
 self.assembly_file = open(filename, 'w', encoding='utf-8')

def close_file(self):
 self.assembly_file.close()
```

Για άνοιγμα και κλείσιμο του assembly αρχείου.

Η `generate_code` εκτελεί το main flow παραγωγής του τελικού κώδικα διαβάζοντας τις παραγόμενες τετράδες του ενδιάμεσου. Θέτει frame pointer = stack pointer, βρίσκει το begin\_block του main προγράμματος και κάνει jump σε αυτό. Επίσης αντιστοιχεί ένα label L{} σε κάθε τετράδα.

```

def generate_code(self, output_file):
 self.open_file(output_file)

 # Αρχικοποίηση
 self.assembly_file.write('# Initialize \n')
 self.assembly_file.write(' mv fp, sp\n\n')

 # Εύρεση entry point του main προγράμματος
 main_entry_label = None
 for i, quad in enumerate(self.quad_manager.quads):
 if quad[1] == "begin_block" and quad[2] == self.quad_manager.program_name:
 main_entry_label = quad[0]
 break

 # Jump στο main
 if main_entry_label:
 self.assembly_file.write(f'# jump stin main \n')
 self.assembly_file.write(f'j L{main_entry_label}\n\n')

 # Μετάφραση κάθε τετράδας
 for quad in self.quad_manager.quads:
 quad_num, op, arg1, arg2, result = quad

 # Label για κάθε τετράδα
 self.assembly_file.write(f'L{quad_num}: \n')

 # Dispatcher για κάθε τύπο operation
 if op == "jump":
 self.assembly_file.write(f' b L{result}\n')
 elif op in ["=", "<>", "<", ">", "<=", ">="]:
 self.create_comparison(op, arg1, arg2, result)
 elif op == ":=":
 self.create_assignment(arg1, result)
 elif op in ["+", "-", "*", "/"]:
 self.create_arithmetic(op, arg1, arg2, result)
 elif op == "out":
 self.create_output(arg1)
 elif op == "in":
 self.create_input(arg1)
 elif op == "par":
 self.create_parameter(arg1, arg2)
 elif op == "call":
 self.create_call(arg1)
 elif op == "begin_block":
 pass
 elif op == "end_block":
 self.create_end_block(arg1)
 elif op == "halt":
 self.create_halt()
 else:
 self.assembly_file.write(f' # Unhandled operation: {op}\n')

 self.close_file()

```

H load\_value φορτώνει τιμές:

```
def load_value(self, value, register):
 if value.isdigit() or (value[0] == '-' and value[1:].isdigit()):
 # Σταθερά (αριθμός)
 self.assembly_file.write(f' li t{register}, {value}\n')
 elif value.startswith("T@"):
 # Προσωρινή μεταβλητή
 self.assembly_file.write(f' lw t{register}, -{int(value[2:])*4+40}(gp)\n')
 else:
 # Μεταβλητή - αναζήτηση στον πίνακα συμβόλων
 symbol = self.symbol_table.lookup(value, self.current_scope)
 if not symbol:
 symbol = self.symbol_table.lookup(value)
 if not symbol:
 symbol = self.symbol_table.lookup_with_nesting(value)
 if not symbol:
 # Έλεγχος για παραμέτρους συνάρτησης
 func_symbol = self.symbol_table.lookup_in_all_scopes(self.current_scope)
 if func_symbol and func_symbol.symbol_type in [SymbolType.FUNCTION, SymbolType.PROCEDURE]:
 for i, (param_name, param_mode) in enumerate(func_symbol.parameters):
 if param_name == value:
 if param_mode == "CV":
 self.assembly_file.write(f' # fortosi {value} me timi\n')
 self.assembly_file.write(f' lw t{register}, -{12+4*i}(sp)\n')
 return
 elif param_mode == "REF":
 self.assembly_file.write(f' # fortosi {value} me anafora\n')
 self.load_value_reference(f"-{12+4*i}", register)
 return
 return

 # Διαχείριση βάσει scope και parameter mode
 if symbol.scope == "global":
 self.assembly_file.write(f' lw t{register}, -{symbol.offset}(gp)\n')
 elif symbol.scope == self.current_scope:
 if symbol.symbol_type == SymbolType.PARAMETER:
 if symbol.parameter_mode == "CV":
 self.assembly_file.write(f' lw t{register}, -{symbol.offset}(sp)\n')
 elif symbol.parameter_mode == "REF":
 self.assembly_file.write(f' # Loading parameter {value} in current scope by reference\n')
 self.load_value_reference(f"-{symbol.offset}", register)
 else:
 self.assembly_file.write(f' lw t{register}, -{symbol.offset}(sp)\n')
 else:
 self.assembly_file.write(f' lw t{register}, -{symbol.offset}(sp)\n')

 elif symbol.parameter_mode == "CV":
 self.assembly_file.write(f' lw t{register}, -{symbol.offset}(sp)\n')
 elif symbol.parameter_mode == "REF":
 self.load_value_reference(f"-{symbol.offset}", register)
 else:
 self.gnlvcode(value, register)
```

Όταν η τιμή είναι μια σταθερά (αριθμός), χρησιμοποιείται η εντολή li (load immediate) για την άμεση φόρτωση της τιμής στο register.

Για τις προσωρινές μεταβλητές "T@", υπολογίζεται μια ειδική διεύθυνση μνήμης από τον global pointer. Ο τύπος υπολογισμού int (value[2:]) \* 4 + 40 δημιουργεί μια περιοχή μνήμης για προσωρινές μεταβλητές, με κάθε μεταβλητή να καταλαμβάνει 4 bytes.

Για τις κανονικές μεταβλητές, η διαδικασία είναι πιο περίπλοκη. Πρώτα γίνεται αναζήτηση στην τρέχουσα εμβέλεια, στη συνέχεια στην καθολική εμβέλεια, και τέλος σε όλες τις nested εμβέλειες. Αν δεν βρεθεί το σύμβολο, γίνεται έλεγχος αν πρόκειται για παράμετρο της τρέχουσας συνάρτησης. Οι παράμετροι CV φορτώνονται απευθείας από τη θέση τους στη στοίβα, ενώ οι παράμετροι REF απαιτούν έμμεση πρόσβαση μέσω της διεύθυνσης που είναι αποθηκευμένη στη στοίβα.

```

def gnlvcode(self, name, register):
 symbol = self.symbol_table.lookup_with_nesting(name)
 if not symbol:
 self.assembly_file.write(f' # Warning: Symbol {name} not found in symbol table\n')
 return

 self.assembly_file.write(f' lw t0, -4(sp)\n')

 current_level = self.symbol_table.current_nesting_level()
 target_level = symbol.nesting_level

 level_diff = current_level - target_level - 1
 for i in range(level_diff):
 self.assembly_file.write(f' lw t0, -4(t0)\n')

 if register != 0:
 self.assembly_file.write(f' lw t{register}, -{symbol.offset}(t0)\n')
 else:
 self.assembly_file.write(f' addi t0, t0, -{symbol.offset}\n')

```

```

def store_value(self, register, target):

 if target.startswith("T@"):
 self.assembly_file.write(f' sw t{register}, -{int(target[2:])*4+40}(gp)\n')
 return

 symbol = self.symbol_table.lookup(target)
 if not symbol:
 symbol = self.symbol_table.lookup_with_nesting(target)
 if not symbol:
 func_symbol = self.symbol_table.lookup_in_all_scopes(self.current_scope)
 if func_symbol and func_symbol.symbol_type in [SymbolType.FUNCTION, SymbolType.PROCEDURE]:
 for i, (param_name, param_mode) in enumerate(func_symbol.parameters):
 if param_name == target:
 if param_mode == "CV":
 self.assembly_file.write(f' sw t{register}, -{12+4*i}(sp)\n')
 return
 elif param_mode == "REF":
 self.store_value_reference(register, f"-{12+4*i}")
 return

 self.assembly_file.write(f' # Warning: Symbol {target} not found in symbol table (current scope: {self.current_scope})\n')
 return

 if symbol.scope == "global":
 self.assembly_file.write(f' sw t{register}, -{symbol.offset}(gp)\n')
 elif symbol.scope == self.current_scope:
 if symbol.symbol_type == SymbolType.PARAMETER:
 if symbol.parameter_mode == "CV":
 self.assembly_file.write(f' sw t{register}, -{symbol.offset}(sp)\n')
 elif symbol.parameter_mode == "REF":
 self.store_value_reference(register, f"-{symbol.offset}")
 else:
 self.assembly_file.write(f' sw t{register}, -{symbol.offset}(sp)\n')
 else:
 self.assembly_file.write(f' sw t{register}, -{symbol.offset}(sp)\n')
 elif symbol.parameter_mode == "CV":
 self.assembly_file.write(f' sw t{register}, -{symbol.offset}(sp)\n')
 elif symbol.parameter_mode == "REF":
 self.store_value_reference(register, f"-{symbol.offset}")
 else:

```

Η μέθοδος `store_value` είναι όμοια με την `load_value`. Όταν μια μεταβλητή είναι παράμετρος REF, η αποθήκευση πρέπει να γίνει έμμεσα, χρησιμοποιώντας τη διεύθυνση που είναι αποθηκευμένη στη στοίβα.

```

def create_comparison(self, op, arg1, arg2, result):
 self.load_value(arg1, 1)
 self.load_value(arg2, 2)

 if op == "=":
 self.assembly_file.write(f' beq t1, t2, L{result}\n')
 elif op == "<>":
 self.assembly_file.write(f' bne t1, t2, L{result}\n')
 elif op == "<":
 self.assembly_file.write(f' blt t1, t2, L{result}\n')
 elif op == ">":
 self.assembly_file.write(f' bgt t1, t2, L{result}\n')
 elif op == "<=":
 self.assembly_file.write(f' ble t1, t2, L{result}\n')
 elif op == ">=":
 self.assembly_file.write(f' bge t1, t2, L{result}\n')

```

Φορτώνονται οι δύο τιμές προς σύγκριση στους registers t1 και t2, και στη συνέχεια χρησιμοποιείται η κατάλληλη εντολή conditional branch.

```

def create_arithmetic(self, op, arg1, arg2, result):
 self.load_value(arg1, 1)
 self.load_value(arg2, 2)

 if op == "+":
 self.assembly_file.write(f' add t1, t1, t2\n')
 elif op == "-":
 self.assembly_file.write(f' sub t1, t1, t2\n')
 elif op == "*":
 self.assembly_file.write(f' mul t1, t1, t2\n')
 elif op == "/":
 self.assembly_file.write(f' div t1, t1, t2\n')

 self.store_value(1, result)

```

Φορτώνονται οι δύο τελεστέοι, εκτελείται η πράξη με την αντίστοιχη RISC-V εντολή, και το αποτέλεσμα αποθηκεύεται στη μεταβλητή προορισμού.

```

def create_output(self, arg):
 self.load_value(arg, 1)
 self.assembly_file.write(' mv a0, t1\n')
 self.assembly_file.write(' li a7, 1\n') # print integer
 self.assembly_file.write(' ecall\n')

 self.assembly_file.write(' li a0, 10\n') # ASCII code for newline
 self.assembly_file.write(' li a7, 11\n') # print character
 self.assembly_file.write(' ecall\n')

def create_input(self, arg):
 self.assembly_file.write(' li a7, 5\n') # read integer
 self.assembly_file.write(' ecall\n')
 self.assembly_file.write(' mv t1, a0\n')
 self.store_value(1, arg)

```

Η είσοδος/έξοδος βασίζεται στις system calls του RISC-V. Για την έξοδο, φορτώνεται η τιμή στο register a0, θετουμε τον κωδικός της system call στο register a7, και καλείται η ecall. Βαζουμε και αυτόματα έναν χαρακτήρα νέας γραμμής για καλυτερη οπτικοποιηση στην εξόδο. Για την είσοδο, καλείται η κατάλληλη system call, το αποτέλεσμα μεταφέρεται από το register a0 σε έναν temporary register, και στη συνέχεια αποθηκεύεται στη μεταβλητή προορισμού.

```

def create_parameter(self, arg1, arg2):
 if self.param_counter == -1:
 # Αρχικοποίηση για νέα κλήση
 self.param_counter = 0

 # Εύρεση της συνάρτησης που θα κληθεί
 func_name = None
 current_quad_index = None
 next_call_label = None

 for i, quad in enumerate(self.quad_manager.quads):
 if quad[1] == "par" and quad[2] == arg1 and quad[3] == arg2:
 current_quad_index = i
 break

 if current_quad_index is not None:
 for i in range(current_quad_index, len(self.quad_manager.quads)):
 if self.quad_manager.quads[i][1] == "call":
 func_name = self.quad_manager.quads[i][2]
 next_call_label = self.quad_manager.quads[i][0]
 break

 if func_name:
 self.current_called_function = func_name
 if next_call_label is not None:
 self.call_setups[next_call_label] = func_name

 func_symbol = self.symbol_table.lookup(func_name, self.current_scope)
 if not func_symbol:
 func_symbol = self.symbol_table.lookup(func_name) or self.symbol_table.lookup_with_nesting(func_name)

 if func_symbol:
 self.assembly_file.write(f' # Setting up frame pointer for {func_name}\n')
 self.assembly_file.write(f' addi fp, sp, {func_symbol.frame_length}\n')

 if arg2 == "CV":
 # Call by Value
 self.load_value(arg1, 0)
 self.assembly_file.write(f' sw t0, -(12+4*self.param_counter)(fp)\n')
 self.param_counter += 1
 elif arg2 == "REF":
 # Call by Reference
 symbol = self.symbol_table.lookup(arg1, self.current_scope)
 if not symbol:
 symbol = self.symbol_table.lookup(arg1)
 if not symbol:
 symbol = self.symbol_table.lookup_with_nesting(arg1)

 if symbol:
 if symbol.scope == self.current_scope:
 self.assembly_file.write(f' addi t0, sp, -{symbol.offset}\n')
 self.assembly_file.write(f' sw t0, -(12+4*self.param_counter)(fp)\n')
 elif symbol.scope == "global":
 self.assembly_file.write(f' addi t0, gp, -{symbol.offset}\n')
 self.assembly_file.write(f' sw t0, -(12+4*self.param_counter)(fp)\n')
 else:
 self.gnlvcode(arg1, 0)
 self.assembly_file.write(f' sw t0, -(12+4*self.param_counter)(fp)\n')
 self.param_counter += 1
 else:
 if arg1.startswith("T@"):
 self.assembly_file.write(f' addi t0, gp, -{int(arg1[2:])*4+40}\n')
 self.assembly_file.write(f' sw t0, -(12+4*self.param_counter)(fp)\n')
 self.param_counter += 1
 else:
 self.assembly_file.write(f' # Warning: Symbol {arg1} not found in symbol table (current scope: {self.current_scope})\n')
 self.param_counter += 1
 elif arg2 == "RET":
 # Return value setup
 if arg1.startswith("T@"):
 self.assembly_file.write(f' addi t0, gp, -{int(arg1[2:])*4+40}\n')
 self.assembly_file.write(f' sw t0, -8(fp)\n')
 else:
 symbol = self.symbol_table.lookup(arg1)
 if not symbol:
 symbol = self.symbol_table.lookup_with_nesting(arg1)

 if symbol:
 if symbol.scope == "global":
 self.assembly_file.write(f' addi t0, gp, -{symbol.offset}\n')
 else:
 self.assembly_file.write(f' addi t0, sp, -{symbol.offset}\n')
 self.assembly_file.write(f' sw t0, -8(fp)\n')
 else:
 self.assembly_file.write(f' # Warning: Symbol {arg1} not found in symbol table\n')

```

Η μέθοδος `create_parameter` αναζητά την επόμενη τετράδα "call" για να προσδιορίσει ποια συνάρτηση θα καλέσει και να προετοιμάσει το activation record. Όταν εντοπίζεται η πρώτη παράμετρος μιας νέας κλήσης, δημιουργείται το frame pointer για τη νέα στοίβα και υπολογίζονται οι σωστές θέσεις για κάθε παράμετρο. Οι παράμετροι που καλούνται με τιμή αντιγράφονται άμεσα στη νέα στοίβα, ενώ οι παράμετροι που καλούνται με αναφορά αντιγράφουν τη διεύθυνσή τους. Για τις συναρτήσεις, δημιουργείται επιπλέον μια ειδική εγγραφή για τη διεύθυνση του return value. Αυτή η διεύθυνση χρησιμοποιείται για να αποθηκεύσει το αποτέλεσμά της.

```

def create_call(self, func_name):
 func_symbol = self.symbol_table.lookup(func_name, self.current_scope)
 if not func_symbol:
 func_symbol = self.symbol_table.lookup(func_name) or self.symbol_table.lookup_with_nesting(func_name)
 if not func_symbol:
 self.assembly_file.write(f' # Warning: Function {func_name} not found in symbol table (current scope: {self.current_scope})\n')
 return

 # Εύρεση του label αυτής της κλήσης
 current_quad_label = None
 for i, quad in enumerate(self.quad_manager.quads):
 if quad[1] == "call" and quad[2] == func_name:
 current_quad_label = quad[0]
 break

 # Έλεγχος consistency με parameters
 expected_func = None
 if current_quad_label in self.call_setups:
 expected_func = self.call_setups[current_quad_label]

 if expected_func and expected_func != func_name:
 self.assembly_file.write(f' # Warning: Calling {func_name} but parameters were set up for {expected_func}\n')
 self.assembly_file.write(f' # Fixing frame pointer setup for {func_name}\n')
 self.assembly_file.write(f' addi fp, sp, {func_symbol.frame_length}\n')

 # Static link setup
 self.assembly_file.write(f' # Setting up static link for {func_name}\n')
 if self.symbol_table.current_nesting_level() == func_symbol.nesting_level:
 self.assembly_file.write(' lw t0, -4(sp)\n')
 self.assembly_file.write(' sw t0, -4(fp)\n')
 else:
 self.assembly_file.write(' sw sp, -4(fp)\n')

 # Function call
 func_type = "function" if func_symbol.symbol_type == SymbolType.FUNCTION else "procedure"
 self.assembly_file.write(f' # Calling {func_type} {func_name}\n')
 self.assembly_file.write(f' addi sp, sp, {func_symbol.frame_length}\n')
 self.assembly_file.write(f' jal L{func_symbol.entry_label}\n')

 # Return value handling (functions only)
 if func_symbol.symbol_type == SymbolType.FUNCTION:
 self.assembly_file.write(f' # Getting return value from function {func_name}\n')
 self.assembly_file.write(' lw t0, -8(sp)\n')
 self.assembly_file.write(' lw t1, 0(t0)\n')
 self.assembly_file.write(' sw t1, 0(t0)\n')

 # Stack cleanup
 self.assembly_file.write(f' addi sp, sp, -{func_symbol.frame_length}\n')

 # Reset parameter counter
 self.param_counter = -1
 if current_quad_label in self.call_setups:
 del self.call_setups[current_quad_label]

```

```

def create_begin_block(self, block_name):
 self.current_scope = block_name
 self.scope_stack.append(block_name) # Push new scope onto stack

 if block_name == self.quad_manager.program_name:

 frame_length = self.symbol_table.get_frame_length("global")
 self.assembly_file.write(f' addi sp, sp, {frame_length}\n') # Allocate space for main AR
 self.assembly_file.write(' mv gp, sp\n') # Set global pointer
 else:

 self.assembly_file.write(' sw ra, 0(sp)\n') # Save return address at offset 0

 func_symbol = self.symbol_table.lookup_in_all_scopes(block_name)
 if func_symbol and func_symbol.frame_length > 12:
 self.assembly_file.write(f' # Frame length: {func_symbol.frame_length}\n')
 self.assembly_file.write(f' # Initializing local variables space\n')

def create_end_block(self, block_name):
 if block_name != self.quad_manager.program_name:

 self.assembly_file.write(' lw ra, 0(sp)\n') # Restore return address
 self.assembly_file.write(' jr ra\n') # Return

 # Pop scope
 if len(self.scope_stack) > 1:
 self.scope_stack.pop()
 self.current_scope = self.scope_stack[-1] # Set to parent scope
 else:
 self.current_scope = "global"

```

Οι μέθοδοι `create_begin_block` και `create_end_block` διαχειρίζονται τα activation records. Για το κύριο πρόγραμμα, δημιουργείται χώρος για τις global μεταβλητές και θέτεται ο global pointer. Για τις συναρτήσεις και διαδικασίες, αποθηκεύεται η διεύθυνση επιστροφής

Κατά την έξοδο από ένα block, αν είναι συνάρτηση ή διαδικασία, επαναφέρεται η διεύθυνση επιστροφής και εκτελείται η εντολή `jr ra` για να επιστρέψουμε. Το scope stack ενημερώνεται κατάλληλα για να δείχνει την τρέχουσα εμβέλεια.

```

def load_value_reference(self, offset, register):
 # Load the address stored at offset(sp)
 self.assembly_file.write(f' lw t0, {offset}(sp)\n')
 # Load the value from that address into the target register
 self.assembly_file.write(f' lw t{register}, 0(t0)\n')

def store_value_reference(self, register, offset):
 # Load the address stored at offset(sp)
 self.assembly_file.write(f' lw t0, {offset}(sp)\n')
 # Store the value from register to that address
 self.assembly_file.write(f' sw t{register}, 0(t0)\n')

```

Οι μέθοδοι `load_value_reference` και `store_value_reference` υλοποιούν την πρόσβαση για τις παραμέτρους που είναι με αναφορά. Πρώτα φορτώνουν τη διεύθυνση από τη στοίβα, και στη συνέχεια παίρνουν την τιμή από αυτή τη διεύθυνση.

```

def create_halt(self):
 self.assembly_file.write(' li a0, 0\n') # Exit code 0
 self.assembly_file.write(' li a7, 93\n') # System call for exit
 self.assembly_file.write(' ecall\n')

```

Η μέθοδος `create_halt` τερματίζει το πρόγραμμα με την `system call` για έξοδο με κωδικό 0.

# TESTS KAI ΠΑΡΑΔΕΙΓΜΑΤΑ ΟΡΘΗΣ ΛΕΙΤΟΥΡΓΙΑΣ

## TEST1

```
(base) chrisk@chrisMBA phase3_final % cat ./Tests/test1.gr
πρόγραμμα τεστ

δήλωση α
δήλωση β,ι
δήλωση τ
δήλωση π

αρχή_προγράμματος

 α := θ ;
 β := 1 ;
 ι := 0;
 όσο ι<=10 επανάλαβε
 γράψε α;
 τ := α+β;
 α := β;
 β := τ;
 ι := ι+1
 όσο_τέλος;

 π := 1;
 ι := 1;
 επανάλαβε
 γράψε π;
 π := π*ι;
 ι := ι+1
 μέχρι ι>7

 τέλος_προγράμματος

(base) chrisk@chrisMBA phase3_final % python3 compiler.py ./Tests/test1.gr
-- Lex Analyzer --
Opening file './Tests/test1.gr'...
Beginning lexical analysis...

πρόγραμμα type:"KEYWORD", line: 1
τεστ type:"IDENTIFIER", line: 1
δήλωση type:"KEYWORD", line: 3
α type:"IDENTIFIER", line: 3
δήλωση type:"KEYWORD", line: 4
β type:"IDENTIFIER", line: 4
, type:"DELIMITER", line: 4
ι type:"IDENTIFIER", line: 4
δήλωση type:"KEYWORD", line: 5
τ type:"IDENTIFIER", line: 5
δήλωση type:"KEYWORD", line: 6
π type:"IDENTIFIER", line: 6
αρχή_προγράμματος type:"KEYWORD", line: 8
α type:"IDENTIFIER", line: 10
```

```

α type:"IDENTIFIER", line: 10
:= type:"ASSIGNMENT", line: 10
0 type:"NUMBER", line: 10
; type:"DELIMITER", line: 10
β type:"IDENTIFIER", line: 11
:= type:"ASSIGNMENT", line: 11
1 type:"NUMBER", line: 11
; type:"DELIMITER", line: 11
L type:"IDENTIFIER", line: 12
:= type:"ASSIGNMENT", line: 12
0 type:"NUMBER", line: 12
; type:"DELIMITER", line: 12
σσο type:"KEYWORD", line: 13
L type:"IDENTIFIER", line: 13
<= type:"RELATIONAL_OPERATOR", line: 13
10 type:"NUMBER", line: 13
επανάλαβε type:"KEYWORD", line: 13
γράψε type:"KEYWORD", line: 14
α type:"IDENTIFIER", line: 14
; type:"DELIMITER", line: 14
τ type:"IDENTIFIER", line: 15
:= type:"ASSIGNMENT", line: 15
α type:"IDENTIFIER", line: 15
+ type:"OPERATOR", line: 15
β type:"IDENTIFIER", line: 15
; type:"DELIMITER", line: 15
α type:"IDENTIFIER", line: 16
:= type:"ASSIGNMENT", line: 16
β type:"IDENTIFIER", line: 16
; type:"DELIMITER", line: 16
β type:"IDENTIFIER", line: 17
:= type:"ASSIGNMENT", line: 17
τ type:"IDENTIFIER", line: 17
; type:"DELIMITER", line: 17
L type:"IDENTIFIER", line: 18
:= type:"ASSIGNMENT", line: 18
L type:"IDENTIFIER", line: 18
+ type:"OPERATOR", line: 18
1 type:"NUMBER", line: 18
σσο_τέλος type:"KEYWORD", line: 19
; type:"DELIMITER", line: 19
π type:"IDENTIFIER", line: 22
:= type:"ASSIGNMENT", line: 22
1 type:"NUMBER", line: 22
; type:"DELIMITER", line: 22
L type:"IDENTIFIER", line: 23
:= type:"ASSIGNMENT", line: 23
1 type:"NUMBER", line: 23
; type:"DELIMITER", line: 23
επανάλαβε type:"KEYWORD", line: 24
γράψε type:"KEYWORD", line: 25
π type:"IDENTIFIER", line: 25

```

```

Ψράψε type:"KEYWORD", line: 25
π type:"IDENTIFIER", line: 25
; type:"DELIMITER", line: 25
π type:"IDENTIFIER", line: 26
:= type:"ASSIGNMENT", line: 26
π type:"IDENTIFIER", line: 26
* type:"OPERATOR", line: 26
λ type:"IDENTIFIER", line: 26
; type:"DELIMITER", line: 26
λ type:"IDENTIFIER", line: 27
:= type:"ASSIGNMENT", line: 27
λ type:"IDENTIFIER", line: 27
+ type:"OPERATOR", line: 27
1 type:"NUMBER", line: 27
μέχρι type:"KEYWORD", line: 28
λ type:"IDENTIFIER", line: 28
> type:"RELATIONAL_OPERATOR", line: 28
7 type:"NUMBER", line: 28
τέλος_προγράμματος type:"KEYWORD", line: 31
-- Reached EOF --

-- Lex Analyzer finished --

-- Syntax Analyzer --
Beginning syntactical analysis...

--No errors--

-- Intermediate Code --
Program: τεστ

1 : begin_block , τεστ , _ , _
2 : := , θ , _ , α
3 : := , 1 , _ , β
4 : := , θ , _ , λ
5 : <= , λ , 10 , 7
6 : jump , _ , _ , 15
7 : out , α , _ , _
8 : + , α , β , T@θ
9 : := , T@θ , _ , τ
10 : := , β , _ , α
11 : := , τ , _ , β
12 : + , λ , 1 , T@1
13 : := , T@1 , _ , λ
14 : jump , _ , _ , 5
15 : := , 1 , _ , π
16 : := , 1 , _ , λ
17 : out , π , _ , _
18 : * , π , λ , T@2
19 : := , T@2 , _ , π
20 : + , λ , 1 , T@3
21 : := , T@3 , _ , λ

```

```

21 : := , T@3 , _ , l
22 : > , l , 7 , 24
23 : jump , _ , _ , 17
24 : halt , _ , _ , _
25 : end_block , τεστ , _ , _

Intermediate code saved in: test1.int
Symbol table scope information saved in: test1.sym
RISC-V assembly saved in: test1.s

-- Syntax Analyzer finished --
(base) chrisk@chrisMBA phase3_final % cat test1.sym
Symbol Table
=====
Global Scope:

Frame Length: 48
Nesting Level: 0

Name: α
Type: VARIABLE
Offset: 12

Name: β
Type: VARIABLE
Offset: 16

Name: l
Type: VARIABLE
Offset: 20

Name: τ
Type: VARIABLE
Offset: 24

Name: π
Type: VARIABLE
Offset: 28

Name: T@0
Type: TEMPORARY
Offset: 32

Name: T@1
Type: TEMPORARY
Offset: 36

Name: T@2
Type: TEMPORARY
Offset: 40

```

```

Name: T@2
Type: TEMPORARY
Offset: 40

Name: T@3
Type: TEMPORARY
Offset: 44

(base) chris@chrisMBA phase3_final % cat test1.int
1 : begin_block , T@0 , _ , -
2 : := , 0 , _ , α
3 : := , 1 , _ , β
4 : := , 0 , _ , ℓ
5 : <= , ℓ , 10 , 7
6 : jump , _ , _ , 15
7 : out , α , _ , -
8 : + , α , β , T@0
9 : := , T@0 , _ , T
10 : := , β , _ , α
11 : := , T , _ , β
12 : + , ℓ , 1 , T@1
13 : := , T@1 , _ , ℓ
14 : jump , _ , _ , 5
15 : := , 1 , _ , π
16 : := , 1 , _ , ℓ
17 : out , π , _ , -
18 : * , π , ℓ , T@2
19 : := , T@2 , _ , π
20 : + , ℓ , 1 , T@3
21 : := , T@3 , _ , ℓ
22 : > , ℓ , 7 , 24
23 : jump , _ , _ , 17
24 : halt , _ , _ , -
25 : end_block , T@0 , _ , -
(base) chris@chrisMBA phase3_final % java -jar ./RISCV_simulator/rars/rars_46ab74d.jar test1.s
RARS 1.5 Copyright 2003-2019 Pete Sanderson and Kenneth Vollmar

0
1
1
2
3
5
8
13
21
34
55
1
1
2
6

```

```

Offset: 44
(base) chrisk@chrisMBA phase3_final % cat test1.int
1 : begin_block , TEST , - , -
2 : := , 0 , - , α
3 : := , 1 , - , β
4 : := , 0 , - , t
5 : <= , t , 10 , 7
6 : jump , - , - , 15
7 : out , α , - , -
8 : + , α , β , T@0
9 : := , T@0 , - , T
10 : := , β , - , α
11 : := , T , - , β
12 : + , t , 1 , T@1
13 : := , T@1 , - , t
14 : jump , - , - , 5
15 : := , 1 , - , π
16 : := , 1 , - , t
17 : out , π , - , -
18 : * , π , t , T@2
19 : := , T@2 , - , π
20 : + , t , 1 , T@3
21 : := , T@3 , - , t
22 : > , t , 7 , 24
23 : jump , - , - , 17
24 : halt , - , - , -
25 : end_block , TEST' , -
(base) chrisk@chrisMBA phase3_final % java -jar ./RISCV_simulator/rars/rars_46ab74d.jar test1.s
RARS 1.5 Copyright 2003-2019 Pete Sanderson and Kenneth Vollmar

0
1
1
2
3
5
8
13
21
34
55
1
1
2
6
24
120
720

Program terminated by calling exit
(base) chrisk@chrisMBA phase3_final %

```

TEST2

```

(base) chrisk@chrisMBA phase3_final % cat ./Tests/test2.gr
πρόγραμμα τεστ2
δήλωση α.β

αρχή_προγράμματος

α := 1;
όσο α<10 επανάλαβε
 α := α+1
όσο_τέλος;
γράψε α;

β := α;
επανάλαβε
 β := β+1;
 β := β-2
μέχρι β=0;
γράψε β;

εάν +β<=-1 τότε
 α := -1
αλλιώς
 α := -2
εάν_τέλος;
γράψε α;

εάν α<β τότε
 α := 1
αλλιώς
 α := 2
εάν_τέλος;
γράψε α

τέλος_προγράμματος
(base) chrisk@chrisMBA phase3_final % python3 compiler.py ./Tests/test2.gr
-- Lex Analyzer --
Opening file './Tests/test2.gr'...
Beginning lexical analysis...

πρόγραμμα type:"KEYWORD", line: 1
τεστ2 type:"IDENTIFIER", line: 1
δήλωση type:"KEYWORD", line: 2
α type:"IDENTIFIER", line: 2
, type:"DELIMITER", line: 2
β type:"IDENTIFIER", line: 2
αρχή_προγράμματος type:"KEYWORD", line: 4
α type:"IDENTIFIER", line: 6
:= type:"ASSIGNMENT", line: 6
1 type:"NUMBER", line: 6
; type:"DELIMITER", line: 6
όσο type:"KEYWORD", line: 7
α type:"IDENTIFIER", line: 7

```

```

< type:"RELATIONAL_OPERATOR", line: 7
10 type:"NUMBER", line: 7
επανάλαβε type:"KEYWORD", line: 7
α type:"IDENTIFIER", line: 8
:= type:"ASSIGNMENT", line: 8
α type:"IDENTIFIER", line: 8
+ type:"OPERATOR", line: 8
1 type:"NUMBER", line: 8
όσο_τέλος type:"KEYWORD", line: 9
: type:"DELIMITER", line: 9
γράψε type:"KEYWORD", line: 10
α type:"IDENTIFIER", line: 10
; type:"DELIMITER", line: 10
β type:"IDENTIFIER", line: 12
:= type:"ASSIGNMENT", line: 12
α type:"IDENTIFIER", line: 12
: type:"DELIMITER", line: 12
επανάλαβε type:"KEYWORD", line: 13
β type:"IDENTIFIER", line: 14
:= type:"ASSIGNMENT", line: 14
β type:"IDENTIFIER", line: 14
+ type:"OPERATOR", line: 14
1 type:"NUMBER", line: 14
: type:"DELIMITER", line: 14
β type:"IDENTIFIER", line: 15
:= type:"ASSIGNMENT", line: 15
β type:"IDENTIFIER", line: 15
- type:"OPERATOR", line: 15
2 type:"NUMBER", line: 15
μέχρι type:"KEYWORD", line: 16
β type:"IDENTIFIER", line: 16
= type:"RELATIONAL_OPERATOR", line: 16
0 type:"NUMBER", line: 16
; type:"DELIMITER", line: 16
γράψε type:"KEYWORD", line: 17
β type:"IDENTIFIER", line: 17
; type:"DELIMITER", line: 17
έὰν type:"KEYWORD", line: 19
+ type:"OPERATOR", line: 19
β type:"IDENTIFIER", line: 19
<= type:"RELATIONAL_OPERATOR", line: 19
- type:"OPERATOR", line: 19
1 type:"NUMBER", line: 19
τότε type:"KEYWORD", line: 19
α type:"IDENTIFIER", line: 20
:= type:"ASSIGNMENT", line: 20
- type:"OPERATOR", line: 20
1 type:"NUMBER", line: 20
αλλιώς type:"KEYWORD", line: 21
α type:"IDENTIFIER", line: 22
:= type:"ASSIGNMENT", line: 22
- type:"OPERATOR", line: 22

```

```

2 type:"NUMBER", line: 22
εάν_τέλος type:"KEYWORD", line: 23
; type:"DELIMITER", line: 23
γράψε type:"KEYWORD", line: 24
α type:"IDENTIFIER", line: 24
; type:"DELIMITER", line: 24
εάν type:"KEYWORD", line: 26
α type:"IDENTIFIER", line: 26
< type:"RELATIONAL_OPERATOR", line: 26
β type:"IDENTIFIER", line: 26
τότε type:"KEYWORD", line: 26
α type:"IDENTIFIER", line: 27
:= type:"ASSIGNMENT", line: 27
1 type:"NUMBER", line: 27
αλλιώς type:"KEYWORD", line: 28
α type:"IDENTIFIER", line: 29
:= type:"ASSIGNMENT", line: 29
2 type:"NUMBER", line: 29
εάν_τέλος type:"KEYWORD", line: 30
; type:"DELIMITER", line: 30
γράψε type:"KEYWORD", line: 31
α type:"IDENTIFIER", line: 31
τέλος_προγράμματος type:"KEYWORD", line: 33
-- Reached EOF --
-- Lex Analyzer --
Beginning syntactical analysis...
--No errors--
-- Intermediate Code --
Program: τεστ2

1 : begin_block , τεστ2 , _ , _
2 : := , 1 , _ , α
3 : < , α , 10 , 5
4 : jump , _ , _ , 8
5 : + , α , 1 , T@0
6 : := , T@0 , _ , α
7 : jump , _ , _ , 3
8 : out , α , _ , -
9 : := , α , _ , β
10 : + , β , 1 , T@1
11 : := , T@1 , _ , β
12 : - , β , 2 , T@2
13 : := , T@2 , _ , β
14 : = , β , 0 , 16
15 : jump , _ , _ , 10
16 : out , β , _ ,
17 : - , 0 , 1 , T@3

```

```

17 : - , 0 , 1 , T@3
18 : <= , β , T@3 , 20
19 : jump , - , - , 23
20 : - , 0 , 1 , T@4
21 : := , T@4 , - , α
22 : jump , - , - , 25
23 : - , 0 , 2 , T@5
24 : := , T@5 , - , α
25 : out , α , - , -
26 : < , α , β , 28
27 : jump , - , - , 30
28 : := , 1 , - , α
29 : jump , - , - , 31
30 : := , 2 , - , α
31 : out , α , - , -
32 : halt , - , - ,
33 : end_block , T@T2 , - , -

Intermediate code saved in: test2.int
Symbol table scope information saved in: test2.sym
RISC-V assembly saved in: test2.s

-- Syntax Analyzer finished --
(base) chrisk@chrisMBA phase3_final % cat test2.sym
Symbol Table
=====

Global Scope:

Frame Length: 44
Nesting Level: 0

Name: α
Type: VARIABLE
Offset: 12

Name: β
Type: VARIABLE
Offset: 16

Name: T@0
Type: TEMPORARY
Offset: 20

Name: T@1
Type: TEMPORARY
Offset: 24

Name: T@2
Type: TEMPORARY
Offset: 28

```

```

Name: T@3
Type: TEMPORARY
Offset: 32

Name: T@4
Type: TEMPORARY
Offset: 36

Name: T@5
Type: TEMPORARY
Offset: 40

(base) chrisk@chrisMBA phase3_final % cat test2.int
1 : begin_block , T@T2 , - , -
2 : := , 1 , _ , α
3 : < , α , 10 , 5
4 : jump , - , - , 8
5 : + , α , 1 , T@0
6 : := , T@0 , - , α
7 : jump , _ , _ , 3
8 : out , α , - , -
9 : := , α , _ , β
10 : + , β , 1 , T@1
11 : := , T@1 , - , β
12 : - , β , 2 , T@2
13 : := , T@2 , - , β
14 : = , β , 0 , 16
15 : jump , - , - , 10
16 : out , β , - , -
17 : - , 0 , 1 , T@3
18 : <= , β , T@3 , 20
19 : jump , - , - , 23
20 : - , 0 , 1 , T@4
21 : := , T@4 , - , α
22 : jump , - , - , 25
23 : - , 0 , 2 , T@5
24 : := , T@5 , - , α
25 : out , α , - , -
26 : < , α , β , 28-
27 : jump , - , - , 30
28 : := , 1 , - , α
29 : jump , - , - , 31
30 : := , 2 , - , α
31 : out , α , - , -
32 : halt , - , - , -
33 : end_block , T@T2 , - , -
(base) chrisk@chrisMBA phase3_final % java -jar ./RISCV_simulator/rars/rars_46ab74d.jar test2.s
RARS 1.5 Copyright 2003-2019 Pete Sanderson and Kenneth Vollmar

10
0

```

```

Name: T@4
Type: TEMPORARY
Offset: 36

Name: T@5
Type: TEMPORARY
Offset: 40

(base) chrisk@chrisMBA phase3_final % cat test2.int
1 : begin_block , T@T2 , _ , -
2 : := , 1 , _ , α
3 : < , α , 10 , 5
4 : jump , _ , _ , 8
5 : + , α , 1 , T@0
6 : := , T@0 , _ , α
7 : jump , _ , _ , 3
8 : out , α , - , -
9 : := , α , _ , β
10 : + , β , 1 , T@1
11 : := , T@1 , _ , β
12 : - , β , 2 , T@2
13 : := , T@2 , _ , β
14 : = , β , 0 , 16
15 : jump , _ , _ , 10
16 : out , β , - , -
17 : - , 0 , 1 , T@3
18 : <= , β , T@3 , 20
19 : jump , _ , _ , 23
20 : - , 0 , 1 , T@4
21 : := , T@4 , _ , α
22 : jump , _ , _ , 25
23 : - , 0 , 2 , T@5
24 : := , T@5 , _ , α
25 : out , α , - , -
26 : < , α , β , 28
27 : jump , _ , _ , 30
28 : := , 1 , _ , α
29 : jump , _ , _ , 31
30 : := , 2 , _ , α
31 : out , α , - , -
32 : halt , _ , _ , -
33 : end_block , T@T2 , _ , -
(base) chrisk@chrisMBA phase3_final % java -jar ./RISCV_simulator/rars/rars_46ab74d.jar test2.s
RARS 1.5 Copyright 2003-2019 Pete Sanderson and Kenneth Vollmar

10
0
-2
1

Program terminated by calling exit
(base) chrisk@chrisMBA phase3_final %

```

## TEST3

```
Last login: Sat May 24 16:57:37 on ttys016
(base) chris@chrisMBA phase3_final % cat Tests/test_all_features.gr
πρόγραμμα δοκιμή_λειτουργιών

δήλωση α, β, γ, δ, ε
δήλωση αποτέλεσμα, μετρητής

συνάρτηση πολλαπλασίασε(χ, ψ)
διαπροσωπεία
είσοδος χ, ψ

αρχή_συνάρτησης
πολλαπλασίασε := χ * ψ
τέλος_συνάρτησης

διαδικασία εμφάνισε_μήνυμα(μήνυμα)
διαπροσωπεία
είσοδος μήνυμα

αρχή_διαδικασίας
γράψε μήνυμα
τέλος_διαδικασίας

διαδικασία αύξησε_κατά_ένα(αριθμός)
διαπροσωπεία
έξοδος αριθμός

αρχή_διαδικασίας
αριθμός := αριθμός + 1
τέλος_διαδικασίας

αρχή_προγράμματος
{ 1. Δοκιμή αριθμητικών πράξεων }
α := 5;
β := 10;
γ := α + β;
δ := γ - α;
ε := α * β / 2;

γράψε γ; { Αναμενόμενο: 15 }
γράψε δ; { Αναμενόμενο: 10 }
γράψε ε; { Αναμενόμενο: 25 }

{ 2. Δοκιμή συνάρτησης }
αποτέλεσμα := πολλαπλασίασε(α, β);
γράψε αποτέλεσμα; { Αναμενόμενο: 50 }

{ 3. Δοκιμή διαδικασίας }
εκτέλεσε εμφάνισε_μήνυμα(αποτέλεσμα); { Αναμενόμενο: 50 }

{ 4. Δοκιμή διαδικασίας με παράμετρο αναφοράς }
```

```

{ 4. Δοκιμή διαδικασίας με παράμετρο αναφοράς }
μετρητής := 0;
εκτέλεσε αύξησε_κατά_ένα(%μετρητής);
γράψε μετρητής; { Αναμενόμενο: 1 }

{ 5. Δοκιμή δομής επανάληψης για }
μετρητής := 0;
για α := 1 έως 5 επανάλαβε
 μετρητής := μετρητής + α
για_τέλος;
γράψε μετρητής; { Αναμενόμενο: 15 (1+2+3+4+5) }

{ 6. Δοκιμή δομής επανάληψης για με βήμα }
μετρητής := 0;
για α := 0 έως 10 με_βήμα 2 επανάλαβε
 μετρητής := μετρητής + α
για_τέλος;
γράψε μετρητής; { Αναμενόμενο: 30 (0+2+4+6+8+10) }

{ 7. Δοκιμή δομής επανάληψης όσο }
α := 1;
μετρητής := 0;
όσο α <= 5 επανάλαβε
 μετρητής := μετρητής + α;
 α := α + 1
όσο_τέλος;
γράψε μετρητής; { Αναμενόμενο: 15 (1+2+3+4+5) }

{ 8. Δοκιμή δομής επανάληψης επανάλαβε-μέχρι }
α := 1;
μετρητής := 0;
επανάλαβε
 μετρητής := μετρητής + α;
 α := α + 1
μέχρι α > 5;
γράψε μετρητής; { Αναμενόμενο: 15 (1+2+3+4+5) }

{ 9. Δοκιμή δομής επιλογής εάν-τότε }
α := 5;
β := 10;
εάν α < β τότε
 γράψε 1
εάν_τέλος; { Αναμενόμενο: 1 }

{ 10. Δοκιμή δομής επιλογής εάν-τότε-αλλιώς }
εάν α > β τότε
 γράψε 1
αλλιώς
 γράψε 0
εάν_τέλος; { Αναμενόμενο: 0 }

{ 11. Δοκιμή σύγθετων λογικών εκφράσεων }

```

```

{ 11. Δοκιμή σύνθετων λογικών εκφράσεων }
εάν [α < β] και [α > 0] τότε
 γράψε 1
εάν_τέλος; { Αναμενόμενο: 1 }

εάν [α > β] ή [β > 0] τότε
 γράψε 1
εάν_τέλος; { Αναμενόμενο: 1 }

{ 12. Δοκιμή εισόδου από χρήστη }
γράψε 100; { Εμφάνιση μηνύματος πριν την εισόδο }
διάβασε α;
γράψε α { Εμφάνιση της τιμής που εισήγαγε ο χρήστης }

τέλος_προγράμματος
(base) chrisk@chrisMBA phase3_final % python3 compiler.py ./Tests/test_all_features.gr
-- Lex Analyzer --
Opening file './Tests/test_all_features.gr'...
Beginning lexical analysis...

πρόγραμμα type:"KEYWORD", line: 1
δοκιμή_λειτουργιών type:"IDENTIFIER", line: 1
δήλωση type:"KEYWORD", line: 3
α type:"IDENTIFIER", line: 3
, type:"DELIMITER", line: 3
β type:"IDENTIFIER", line: 3
, type:"DELIMITER", line: 3
γ type:"IDENTIFIER", line: 3
, type:"DELIMITER", line: 3
δ type:"IDENTIFIER", line: 3
, type:"DELIMITER", line: 3
ε type:"IDENTIFIER", line: 3
δήλωση type:"KEYWORD", line: 4
αποτέλεσμα type:"IDENTIFIER", line: 4
, type:"DELIMITER", line: 4
μετρητής type:"IDENTIFIER", line: 4
συνάρτηση type:"KEYWORD", line: 6
πολλαπλασίασε type:"IDENTIFIER", line: 6
(type:"GROUP_SYMBOL", line: 6
X type:"IDENTIFIER", line: 6
, type:"DELIMITER", line: 6
ψ type:"IDENTIFIER", line: 6
) type:"GROUP_SYMBOL", line: 6
διαμροσωμένα type:"KEYWORD", line: 7
εισόδος type:"KEYWORD", line: 8
X type:"IDENTIFIER", line: 8
, type:"DELIMITER", line: 8
ψ type:"IDENTIFIER", line: 8
αρχή_συνάρτησης type:"KEYWORD", line: 11
πολλαπλασίασε type:"IDENTIFIER", line: 12
:= type:"ASSIGNMENT", line: 12
X type:"IDENTIFIER", line: 12

```

```

X type:"IDENTIFIER", line: 12
* type:"OPERATOR", line: 12
Ψ type:"IDENTIFIER", line: 12
Τέλος_συνάρτησης type:"KEYWORD", line: 13
διαδικασία type:"KEYWORD", line: 15
εμφάνισε_μήνυμα type:"IDENTIFIER", line: 15
(type:"GROUP_SYMBOL", line: 15
μήνυμα type:"IDENTIFIER", line: 15
) type:"GROUP_SYMBOL", line: 15
διαπροσωπεία type:"KEYWORD", line: 16
εισόδος type:"KEYWORD", line: 17
μήνυμα type:"IDENTIFIER", line: 17
αρχή_διαδικασίας type:"KEYWORD", line: 19
γράψε type:"KEYWORD", line: 20
μήνυμα type:"IDENTIFIER", line: 20
τέλος_διαδικασίας type:"KEYWORD", line: 21
διαδικασία type:"KEYWORD", line: 23
αύξησε_κατά_ένα type:"IDENTIFIER", line: 23
(type:"GROUP_SYMBOL", line: 23
αριθμός type:"IDENTIFIER", line: 23
) type:"GROUP_SYMBOL", line: 23
διαπροσωπεία type:"KEYWORD", line: 24
έξοδος type:"KEYWORD", line: 25
αριθμός type:"IDENTIFIER", line: 25
αρχή_διαδικασίας type:"KEYWORD", line: 27
αριθμός type:"IDENTIFIER", line: 28
:= type:"ASSIGNMENT", line: 28
αριθμός type:"IDENTIFIER", line: 28
+ type:"OPERATOR", line: 28
1 type:"NUMBER", line: 28
τέλος_διαδικασίας type:"KEYWORD", line: 29
αρχή_προγράμματος type:"KEYWORD", line: 31
α type:"IDENTIFIER", line: 33
:= type:"ASSIGNMENT", line: 33
5 type:"NUMBER", line: 33
; type:"DELIMITER", line: 33
β type:"IDENTIFIER", line: 34
:= type:"ASSIGNMENT", line: 34
10 type:"NUMBER", line: 34
; type:"DELIMITER", line: 34
γ type:"IDENTIFIER", line: 35
:= type:"ASSIGNMENT", line: 35
α type:"IDENTIFIER", line: 35
+ type:"OPERATOR", line: 35
β type:"IDENTIFIER", line: 35
; type:"DELIMITER", line: 35
δ type:"IDENTIFIER", line: 36
:= type:"ASSIGNMENT", line: 36
γ type:"IDENTIFIER", line: 36
- type:"OPERATOR", line: 36
α type:"IDENTIFIER", line: 36
; type:"DELIMITER", line: 36

```

```

α type:"IDENTIFIER", line: 36
: type:"DELIMITER", line: 36
ε type:"IDENTIFIER", line: 37
:= type:"ASSIGNMENT", line: 37
α type:"IDENTIFIER", line: 37
* type:"OPERATOR", line: 37
β type:"IDENTIFIER", line: 37
/ type:"OPERATOR", line: 37
2 type:"NUMBER", line: 37
; type:"DELIMITER", line: 37
γράψε type:"KEYWORD", line: 39
γ type:"IDENTIFIER", line: 39
; type:"DELIMITER", line: 39
γράψε type:"KEYWORD", line: 40
δ type:"IDENTIFIER", line: 40
; type:"DELIMITER", line: 40
γράψε type:"KEYWORD", line: 41
ε type:"IDENTIFIER", line: 41
; type:"DELIMITER", line: 41
αποτέλεσμα type:"IDENTIFIER", line: 44
:= type:"ASSIGNMENT", line: 44
πολλαπλασιασε type:"IDENTIFIER", line: 44
(type:"GROUP_SYMBOL", line: 44
α type:"IDENTIFIER", line: 44
, type:"DELIMITER", line: 44
β type:"IDENTIFIER", line: 44
) type:"GROUP_SYMBOL", line: 44
; type:"DELIMITER", line: 44
γράψε type:"KEYWORD", line: 45
αποτέλεσμα type:"IDENTIFIER", line: 45
; type:"DELIMITER", line: 45
εκτέλεσε type:"KEYWORD", line: 48
εμφάνισε_μήνυμα type:"IDENTIFIER", line: 48
(type:"GROUP_SYMBOL", line: 48
αποτέλεσμα type:"IDENTIFIER", line: 48
) type:"GROUP_SYMBOL", line: 48
; type:"DELIMITER", line: 48
μετρητής type:"IDENTIFIER", line: 51
:= type:"ASSIGNMENT", line: 51
θ type:"NUMBER", line: 51
; type:"DELIMITER", line: 51
εκτέλεσε type:"KEYWORD", line: 52
αύξησε_κατά_ένα type:"IDENTIFIER", line: 52
(type:"GROUP_SYMBOL", line: 52
% type:"PASSBYREFERENCE", line: 52
μετρητής type:"IDENTIFIER", line: 52
) type:"GROUP_SYMBOL", line: 52
; type:"DELIMITER", line: 52
γράψε type:"KEYWORD", line: 53
μετρητής type:"IDENTIFIER", line: 53
; type:"DELIMITER", line: 53
μετρητής type:"IDENTIFIER", line: 56

```

```

:= type:"ASSIGNMENT", line: 56
0 type:"NUMBER", line: 56
; type:"DELIMITER", line: 56
για type:"KEYWORD", line: 57
α type:"IDENTIFIER", line: 57
:= type:"ASSIGNMENT", line: 57
1 type:"NUMBER", line: 57
έως type:"KEYWORD", line: 57
5 type:"NUMBER", line: 57
επανάλαβε type:"KEYWORD", line: 57
μετρητής type:"IDENTIFIER", line: 58
:= type:"ASSIGNMENT", line: 58
μετρητής type:"IDENTIFIER", line: 58
+ type:"OPERATOR", line: 58
α type:"IDENTIFIER", line: 58
για_τέλος type:"KEYWORD", line: 59
; type:"DELIMITER", line: 59
γράψε type:"KEYWORD", line: 60
μετρητής type:"IDENTIFIER", line: 60
; type:"DELIMITER", line: 60
μετρητής type:"IDENTIFIER", line: 63
:= type:"ASSIGNMENT", line: 63
0 type:"NUMBER", line: 63
; type:"DELIMITER", line: 63
για type:"KEYWORD", line: 64
α type:"IDENTIFIER", line: 64
:= type:"ASSIGNMENT", line: 64
0 type:"NUMBER", line: 64
έως type:"KEYWORD", line: 64
10 type:"NUMBER", line: 64
με_βήμα type:"KEYWORD", line: 64
2 type:"NUMBER", line: 64
επανάλαβε type:"KEYWORD", line: 64
μετρητής type:"IDENTIFIER", line: 65
:= type:"ASSIGNMENT", line: 65
μετρητής type:"IDENTIFIER", line: 65
+ type:"OPERATOR", line: 65
α type:"IDENTIFIER", line: 65
για_τέλος type:"KEYWORD", line: 66
; type:"DELIMITER", line: 66
γράψε type:"KEYWORD", line: 67
μετρητής type:"IDENTIFIER", line: 67
; type:"DELIMITER", line: 67
α type:"IDENTIFIER", line: 70
:= type:"ASSIGNMENT", line: 70
1 type:"NUMBER", line: 70
; type:"DELIMITER", line: 70
μετρητής type:"IDENTIFIER", line: 71
:= type:"ASSIGNMENT", line: 71
0 type:"NUMBER", line: 71
; type:"DELIMITER", line: 71
όσο type:"KEYWORD", line: 72

```

```

όσο type:"KEYWORD", line: 72
α type:"IDENTIFIER", line: 72
<= type:"RELATIONAL_OPERATOR", line: 72
5 type:"NUMBER", line: 72
επανάλαβε type:"KEYWORD", line: 72
μετρητής type:"IDENTIFIER", line: 73
:= type:"ASSIGNMENT", line: 73
μετρητής type:"IDENTIFIER", line: 73
+ type:"OPERATOR", line: 73
α type:"IDENTIFIER", line: 73
; type:"DELIMITER", line: 73
α type:"IDENTIFIER", line: 74
:= type:"ASSIGNMENT", line: 74
α type:"IDENTIFIER", line: 74
+ type:"OPERATOR", line: 74
1 type:"NUMBER", line: 74
όσο_τέλος type:"KEYWORD", line: 75
; type:"DELIMITER", line: 75
γράψε type:"KEYWORD", line: 76
μετρητής type:"IDENTIFIER", line: 76
; type:"DELIMITER", line: 76
α type:"IDENTIFIER", line: 79
:= type:"ASSIGNMENT", line: 79
1 type:"NUMBER", line: 79
; type:"DELIMITER", line: 79
μετρητής type:"IDENTIFIER", line: 80
:= type:"ASSIGNMENT", line: 80
0 type:"NUMBER", line: 80
; type:"DELIMITER", line: 80
επανάλαβε type:"KEYWORD", line: 81
μετρητής type:"IDENTIFIER", line: 82
:= type:"ASSIGNMENT", line: 82
μετρητής type:"IDENTIFIER", line: 82
+ type:"OPERATOR", line: 82
α type:"IDENTIFIER", line: 82
; type:"DELIMITER", line: 82
α type:"IDENTIFIER", line: 83
:= type:"ASSIGNMENT", line: 83
α type:"IDENTIFIER", line: 83
+ type:"OPERATOR", line: 83
1 type:"NUMBER", line: 83
μέχρι type:"KEYWORD", line: 84
α type:"IDENTIFIER", line: 84
> type:"RELATIONAL_OPERATOR", line: 84
5 type:"NUMBER", line: 84
; type:"DELIMITER", line: 84
γράψε type:"KEYWORD", line: 85
μετρητής type:"IDENTIFIER", line: 85
; type:"DELIMITER", line: 85
α type:"IDENTIFIER", line: 88
:= type:"ASSIGNMENT", line: 88
5 type:"NUMBER", line: 88

```

```
5 type:"NUMBER", line: 88
:
β type:"IDENTIFIER", line: 89
:= type:"ASSIGNMENT", line: 89
10 type:"NUMBER", line: 89
:
εάν type:"KEYWORD", line: 90
α type:"IDENTIFIER", line: 90
< type:"RELATIONAL_OPERATOR", line: 90
β type:"IDENTIFIER", line: 90
τότε type:"KEYWORD", line: 90
γράψε type:"KEYWORD", line: 91
1 type:"NUMBER", line: 91
εάν_Τέλος type:"KEYWORD", line: 92
:
εάν type:"KEYWORD", line: 92
α type:"IDENTIFIER", line: 95
> type:"RELATIONAL_OPERATOR", line: 95
β type:"IDENTIFIER", line: 95
τότε type:"KEYWORD", line: 95
γράψε type:"KEYWORD", line: 96
1 type:"NUMBER", line: 96
αλλιώς type:"KEYWORD", line: 97
γράψε type:"KEYWORD", line: 98
θ type:"NUMBER", line: 98
εάν_Τέλος type:"KEYWORD", line: 99
:
εάν type:"KEYWORD", line: 102
[type:"GROUP_SYMBOL", line: 102
α type:"IDENTIFIER", line: 102
< type:"RELATIONAL_OPERATOR", line: 102
β type:"IDENTIFIER", line: 102
] type:"GROUP_SYMBOL", line: 102
καλ type:"KEYWORD", line: 102
[type:"GROUP_SYMBOL", line: 102
α type:"IDENTIFIER", line: 102
> type:"RELATIONAL_OPERATOR", line: 102
θ type:"NUMBER", line: 102
] type:"GROUP_SYMBOL", line: 102
τότε type:"KEYWORD", line: 102
γράψε type:"KEYWORD", line: 103
1 type:"NUMBER", line: 103
εάν_Τέλος type:"KEYWORD", line: 104
:
εάν type:"KEYWORD", line: 106
[type:"GROUP_SYMBOL", line: 106
α type:"IDENTIFIER", line: 106
> type:"RELATIONAL_OPERATOR", line: 106
β type:"IDENTIFIER", line: 106
] type:"GROUP_SYMBOL", line: 106
η type:"KEYWORD", line: 106
[type:"GROUP_SYMBOL", line: 106
```

```

> type:"RELATIONAL_OPERATOR", line: 106
0 type:"NUMBER", line: 106
] type:"GROUP_SYMBOL", line: 106
τότε type:"KEYWORD", line: 106
γράψε type:"KEYWORD", line: 107
1 type:"NUMBER", line: 107
εάν_Τέλος type:"KEYWORD", line: 108
; type:"DELIMITER", line: 108
γράψε type:"KEYWORD", line: 111
100 type:"NUMBER", line: 111
; type:"DELIMITER", line: 111
διάβασε type:"KEYWORD", line: 112
α type:"IDENTIFIER", line: 112
; type:"DELIMITER", line: 112
γράψε type:"KEYWORD", line: 113
α type:"IDENTIFIER", line: 113
τέλος_προγράμματος type:"KEYWORD", line: 115
-- Reached EOF --

-- Lex Analyzer finished --

-- Syntax Analyzer --
Beginning syntactical analysis...

--No errors--

-- Intermediate Code --
Program: δοκιμή_λειτουργιών

1 : begin_block , πολλαπλασίασε , _ , _
2 : * , X , ψ , T@0
3 : := , T@0 , _ , πολλαπλασίασε
4 : end_block , πολλαπλασίασε , _ , _
5 : begin_block , εμφάνισε_μήνυμα , _ , _
6 : out , μήνυμα , _ , _
7 : end_block , εμφάνισε_μήνυμα , _ , _
8 : begin_block , αύξησε_κατά_ένα , _ , _
9 : + , αριθμός , 1 , T@1
10 : := , T@1 , _ , αριθμός
11 : end_block , αύξησε_κατά_ένα , _ , _
12 : begin_block , δοκιμή_λειτουργιών , _ , _
13 : := , 5 , _ , α
14 : := , 10 , _ , β
15 : + , α , β , T@2
16 : := , T@2 , _ , γ
17 : - , γ , α , T@3
18 : := , T@3 , _ , δ
19 : * , α , β , T@4
20 : / , T@4 , 2 , T@5
21 : := , T@5 , _ , ε
22 : out , γ , _ , -
23 : out , δ , _ , -

```

```

24 : out , ε , _ , -
25 : par , α , CV , -
26 : par , β , CV , -
27 : par , T@6 , RET , -
28 : call , πολλαπλασίασε , _ , -
29 : := , T@6 , _ , αποτέλεσμα
30 : out , αποτέλεσμα , _ , -
31 : par , αποτέλεσμα , CV , -
32 : call , εμφάνισε_μήνυμα , _ , -
33 : := , θ , _ , μετρητής
34 : par , μετρητής , REF , -
35 : call , αύξησε_κατά_ένα : _ , -
36 : out , μετρητής , _ , -
37 : := , θ , _ , μετρητής
38 : := , 1 , _ , α
39 : := , 5 , _ , T@7
40 : := , 1 , _ , T@8
41 : <= , α , T@7 , 43
42 : jump , _ , _ , 48
43 : + , μετρητής , α , T@9
44 : := , T@9 , _ , μετρητής
45 : + , α , T@8 , T@10
46 : := , T@10 , _ , α
47 : jump , _ , _ , 41
48 : out , μετρητής , _ , -
49 : := , θ , _ , μετρητής
50 : := , 0 , _ , α
51 : := , 10 , _ , T@11
52 : := , 2 , _ , T@12
53 : <= , α , T@11 , 55
54 : jump , _ , _ , 60
55 : + , μετρητής , α , T@13
56 : := , T@13 , _ , μετρητής
57 : + , α , T@12 , T@14
58 : := , T@14 , _ , α
59 : jump , _ , _ , 53
60 : out , μετρητής , _ , -
61 : := , 1 , _ , α
62 : := , 0 , _ , μετρητής
63 : <= , α , 5 , 65
64 : jump , _ , _ , 70
65 : + , μετρητής , α , T@15
66 : := , T@15 , _ , μετρητής
67 : + , α , 1 , T@16
68 : := , T@16 , _ , α
69 : jump , _ , _ , 63
70 : out , μετρητής , _ , -
71 : := , 1 , _ , α
72 : := , 0 , _ , μετρητής
73 : + , μετρητής , α , T@17
74 : := , T@17 , _ , μετρητής
75 : + , α , 1 , T@18

```

```

75 : + , α , 1 , T@18
76 : := , T@18 , _ , α
77 : > , α , 5 , _99
78 : jump , _ , _ , 73
79 : out , μετρητής , _ , _
80 : := , 5 , _ , α
81 : := , 10 , _ , β
82 : < , α , β , 84
83 : jump , _ , _ , 86
84 : out , 1 , _ , _
85 : jump , _ , _ , 86
86 : > , α , β , 88
87 : jump , _ , _ , 90
88 : out , 1 , _ , _
89 : jump , _ , _ , 91
90 : out , 0 , _ , _
91 : < , α , β , 93
92 : jump , _ , _ , 97
93 : > , α , 0 , 95
94 : jump , _ , _ , 97
95 : out , 1 , _ , _
96 : jump , _ , _ , 97
97 : > , α , β , 101
98 : jump , _ , _ , 99
99 : > , β , 0 , 101
100 : jump , _ , _ , 103
101 : out , 1 , _ , _
102 : jump , _ , _ , 103
103 : out , 100 , _ , _
104 : in , α , _ , _
105 : out , α , _ , _
106 : halt , _ , _ , _
107 : end_block , δοκιμή_λειτουργιών , _ , _

Intermediate code saved in: test_all_features.int
Symbol table scope information saved in: test_all_features.sym
RISC-V assembly saved in: test_all_features.s

-- Syntax Analyzer finished --
(base) chrisk@chrisMBA phase3_final % cat test_all_features.sym
Symbol Table
=====

Global Scope:

Frame Length: 120
Nesting Level: 0

Name: α
Type: VARIABLE
Offset: 12

```

```
Name: α
Type: VARIABLE
Offset: 12

Name: β
Type: VARIABLE
Offset: 16

Name: γ
Type: VARIABLE
Offset: 20

Name: δ
Type: VARIABLE
Offset: 24

Name: ε
Type: VARIABLE
Offset: 28

Name: αποτέλεσμα
Type: VARIABLE
Offset: 32

Name: μετρητής
Type: VARIABLE
Offset: 36

Name: πολλαπλασιασε
Type: FUNCTION
Offset: 40
Parameters: [('X', 'CV'), ('Ψ', 'CV')]
Frame Length: 24
Entry Label: 1

Name: εμφάνισε_μήνυμα
Type: PROCEDURE
Offset: 44
Parameters: [('μήνυμα', 'CV')]
Frame Length: 16
Entry Label: 5

Name: αύξησε_κατά_ένα
Type: PROCEDURE
Offset: 48
Parameters: [('αριθμός', 'CV')]
Frame Length: 20
Entry Label: 8

Name: T@2
Type: TEMPORARY
Offset: 52
```

```
Name: T@3
Type: TEMPORARY
Offset: 56

Name: T@4
Type: TEMPORARY
Offset: 60

Name: T@5
Type: TEMPORARY
Offset: 64

Name: T@6
Type: TEMPORARY
Offset: 68

Name: T@7
Type: TEMPORARY
Offset: 72

Name: T@8
Type: TEMPORARY
Offset: 76

Name: T@9
Type: TEMPORARY
Offset: 80

Name: T@10
Type: TEMPORARY
Offset: 84

Name: T@11
Type: TEMPORARY
Offset: 88

Name: T@12
Type: TEMPORARY
Offset: 92

Name: T@13
Type: TEMPORARY
Offset: 96

Name: T@14
Type: TEMPORARY
Offset: 100

Name: T@15
Type: TEMPORARY
Offset: 104
```

```
Name: T@16
Type: TEMPORARY
Offset: 108

Name: T@17
Type: TEMPORARY
Offset: 112

Name: T@18
Type: TEMPORARY
Offset: 116

Πολλαπλασιασε Scope:

Frame Length: 24
Nesting Level: 1
Parent Scope: <bound method SymbolTable.current_scope of <__main__.SymbolTable object at 0x104cd07d0>>

Name: X
Type: PARAMETER
Offset: 12
Parameter Mode: CV

Name: ψ
Type: PARAMETER
Offset: 16
Parameter Mode: CV

Name: T@0
Type: TEMPORARY
Offset: 20

Εμφανισε_Μήνυμα Scope:

Frame Length: 16
Nesting Level: 1
Parent Scope: <bound method SymbolTable.current_scope of <__main__.SymbolTable object at 0x104cd07d0>>

Name: μήνυμα
Type: PARAMETER
Offset: 12
Parameter Mode: CV

Αύξησε_Κατά_Ενα Scope:

Frame Length: 20
Nesting Level: 1
Parent Scope: <bound method SymbolTable.current_scope of <__main__.SymbolTable object at 0x104cd07d0>>
```

```

Name: αριθμός
Type: PARAMETER
Offset: 12
Parameter Mode: CV

Name: T@1
Type: TEMPORARY
Offset: 16

(base) chrisk@chrisMBA phase3_final % cat test_all_features.int
1 : begin_block , πολλαπλασιάσε , _ , -
2 : * . X . Ψ , T@0
3 : := , T@0 , _ , πολλαπλασιάσε
4 : end_block , πολλαπλασιάσε , _ , -
5 : begin_block , εμφάνισε_μήνυμα , _ , -
6 : out , μήνυμα , _ , -
7 : end_block , εμφάνισε_μήνυμα , _ , -
8 : begin_block , αύξησε_κατά_ένα , _ , -
9 : + , αριθμός , 1 , T@1
10 : := , T@1 , _ , αριθμός
11 : end_block , αύξησε_κατά_ένα , _ ,
12 : begin_block , δοκιμή_λειτουργιών , _ , -
13 : := , 5 , _ , α
14 : := , 10 , _ , β
15 : + , α , β , T@2
16 : := , T@2 , _ , γ
17 : - , γ , α , T@3
18 : := , T@3 , _ , δ
19 : * , α , β , T@4
20 : / , T@4 , 2 , T@5
21 : := , T@5 , _ , ε
22 : out , γ , _ , -
23 : out , δ , _ , -
24 : out , ε , _ , -
25 : par , α , CV , -
26 : par , β , CV , -
27 : par , T@6 . RET , -
28 : call , πολλαπλασιάσε , _ , -
29 : := , T@6 , _ , αποτέλεσμα
30 : out , αποτέλεσμα , _ , -
31 : par , αποτέλεσμα , CV , -
32 : call , εμφάνισε_μήνυμα , _ , -
33 : := , 0 , _ , μετρητής
34 : par , μετρητής , REF , -
35 : call , αύξησε_κατά_ένα , _ , -
36 : out , μετρητής , _ , -
37 : := , 0 , _ , μετρητής
38 : := , 1 , _ , α
39 : := , 5 , _ , T@7
40 : := , 1 , _ , T@8
41 : <= , α , T@7 , 43

```

```

42 : jump . _ , _ , 48
43 : + , μετρητής , α , T@9
44 : := , T@9 , _ , μετρητής
45 : + , α , T@8 , T@10
46 : := , T@10 , _ , α
47 : jump . _ , _ , 41
48 : out , μετρητής , _ , -
49 : := , θ , _ , μετρητής
50 : := , θ , _ , α
51 : := , 10 , _ , T@11
52 : := , 2 , _ , T@12
53 : <= , α , T@11 , 55
54 : jump . _ , _ , 60
55 : + , μετρητής , α , T@13
56 : := , T@13 , _ , μετρητής
57 : + , α , T@12 , T@14
58 : := , T@14 , _ , α
59 : jump . _ , _ , 53
60 : out , μετρητής , _ , -
61 : := , 1 , _ , α
62 : := , θ , _ , μετρητής
63 : <= , α , 5 , 65
64 : jump . _ , _ , 70
65 : + , μετρητής , α , T@15
66 : := , T@15 , _ , μετρητής
67 : + , α , 1 , T@16
68 : := , T@16 , _ , α
69 : jump . _ , _ , 63
70 : out , μετρητής , _ , -
71 : := , 1 , _ , α
72 : := , θ , _ , μετρητής
73 : + , μετρητής , α , T@17
74 : := , T@17 , _ , μετρητής
75 : + , α , 1 , T@18
76 : := , T@18 , _ , α
77 : > , α , 5 , 79
78 : jump . _ , _ , 73
79 : out , μετρητής , _ , -
80 : := , 5 , _ , α
81 : := , 10 , _ , β
82 : < , α , β , 84
83 : jump . _ , _ , 86
84 : out , 1 , _ , -
85 : jump . _ , _ , 86
86 : > , α , β , 88
87 : jump . _ , _ , 90
88 : out , 1 , _ , -
89 : jump . _ , _ , 91
90 : out , 0 , _ , -
91 : < , α , β , 93
92 : jump . _ , _ , 97
93 : > , α , 0 , 95

```

```

79 : out , μετρητής . _ _ -
80 : := , 5 , _ , α
81 : := , 10 , _ , β
82 : < , α , β , 84
83 : jump , _ , _ , 86
84 : out , 1 , _ , _ - 86
85 : jump , _ , _ , _ - 86
86 : > , α , β , 88
87 : jump , _ , _ , 90
88 : out , 1 , _ , _ - 90
89 : jump , _ , _ , 91
90 : out , 0 , _ , _ - 91
91 : < , α , β , 93
92 : jump , _ , _ , 97
93 : > , α , 0 , 95
94 : jump , _ , _ , 97
95 : out , 1 , _ , _ - 97
96 : jump , _ , _ , 97
97 : > , α , β , 101
98 : jump , _ , _ , 99
99 : > , β , 0 , 101
100 : jump , _ , _ , 103
101 : out , 1 , _ , _ - 103
102 : jump , _ , _ , 103
103 : out , 100 , _ , _ - 103
104 : in , α , _ , _ - 104
105 : out , α , _ , _ - 105
106 : halt , _ , _ , _ - 106
107 : end_block , δοκιμή λειτουργιών , _ , _ - 107
(base) chrisk@chrisMBA phase3_final % java -jar RISCV_simulator/rars/rars_46ab74d.jar test_all_features.s
RARS 1.5 Copyright 2003-2019 Pete Sanderson and Kenneth Vollmar

15
10
25
50
50
0
15
30
15
15
1
0
1
1
100
5
5

Program terminated by calling exit
(base) chrisk@chrisMBA phase3_final %

```

# TEST DEFAULT INTERMEDIATE CODE

Testing test default

Thursday, 22 May 2025 23:09

πρόγραμμα τεστ  
δήλωση α,β  
δήλωση γ

```

1 : begin_block , αύξηση , - , -
2 : + , α , 1 , T@0
3 : := , T@0 , - , β
4 : + , α , 1 , T@1
5 : = , T@1 , - , αύξηση
6 : end_block , αύξηση , - , -

```

συνάρτηση αύξηση(α,β)  
διαπροσωπεία  
είσοδος α  
έξοδος β  
αρχή\_συνάρτησης  
β:=α+1;  
αύξηση:=α+1 { δεν μπαίνει ερωτηματικό  
είναι τέλος block }  
τέλος\_συνάρτησης

```

7 : begin_block , τύπωσε_συν_1 , - , -
8 : + , X , 1 , T@2
9 : out , T@2 , - , -
10 : end_block , τύπωσε_συν_1 , - , -

```

διαδικασία τύπωσε\_συν\_1(χ)  
διαδικασία  
είσοδος χ  
αρχή\_διαδικασίας  
γράμματα;  
τέλος\_διαδικασίας

```

11 : begin_block , τεστ , - , -
12 : := , 1 , - , α
13 : * , α , α , T@3
14 : * , 2 , α , T@4
15 : * , 2 , α , T@5
16 : - , T@4 , T@5 , T@6
17 : / , T@3 , T@6 , T@7
18 : + , 2 , T@7 , T@8
19 : = , T@8 , - , β

```

αρχή\_προγράμματος  
α:=1;  
β=2 + α \* α / (2 \* α - (2 \* α));

```

20 : par , α , CV ,
21 : par , β , REF ,
22 : par , T@9 , RET , -
23 : call , ούνηση , - , -
24 : = , T@9 , - , γ

```

```

25 : := , 1 , - , α
26 : := , 8 , - , T@10
27 : := , 2 , - , T@11
28 : <= , α , T@10 , 30
29 : jump , - , - , 35
30 : par , α , CV ,
31 : call , τύπωσε_συν_1 , - , -
32 : + , α , T@11 , T@12
33 : := , T@12 , - , α
34 : jump , - , - , 28

```

για α:=1 έως 8 με βήμα 2 επανάλαβε  
ΕΚΤΕΛΕΣΕ τύπωσε\_συν\_1()  
για τέλος;

```

35 : := , 1 , - , β
36 : < , β , 10 , 38
37 : jump , - , - , 48
38 : > , β , 22 , 44
39 : jump , - , - , 40
40 : >= , β , 23 , 42
41 : jump , - , - , 47
42 : <= , β , 24 , 44
43 : jump , - , - , 47
44 : + , β , 1 , T@13
45 : := , T@13 , - , β
46 : jump , - , - , 47
47 : jump , - , - , 36

```

β:=1;  
όσο β<10 επανάλαβε  
εάν β>22 ή [β>23 και β<=24] τότε  
β:=β+1  
εάν τέλος { όχι ερωτηματικό, είναι τέλος block }  
όσο\_τέλος; { θέλει ερωτηματικό  
χωρίζει εκτελέσιμες, εντολές }  
Return to the for loop 46 : := , T@14 , - , β

```

48 : in , β , - ,
49 : + , β , 1 , T@14
50 : := , T@14 , - , β
51 : - , 0 , 100 , T@15
52 : < , β , T@15 , 54
53 : jump , - , - , 49
54 : halt , - , - , -
55 : end_block , τεστ , - , -

```

διάβασε β;  
επανάλαβε  
β:=β+1  
μέχρι β<=100  
τέλος\_προγράμματος  
β<-100  
Ends the while loop