

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 5 з дисципліни
«Основи програмування-2»

«Дерева»

Варіант 15

Виконав студент ІП-11 Книш Дмитро Олегович
(шифр, прізвище, ім'я, по батькові)

Перевірив _____
(прізвище, ім'я, по батькові)

Київ 2022

Лабораторна робота 5

Дерева

Мета – вивчити особливості організації і обробки дерев.

Індивідуальне завдання:

15. Текстовий файл містить програму мовою C/C++. Для збереження ідентифікаторів програми використати структуру типу дерева, елементами якого є ідентифікатори. Номер рядка, в якому оголошений ідентифікатор, визначає рівень дерева. Ліва гілка дерева визначає змінні, права гілка - константи.

Математична модель:

Таблиця імен змінних:

Змінна	Тип	Опис
file_path	string	Шлях до файлу
root_lexer	Lexer	Екземпляр лексера
line	string	Поточний рядок в файлі
line_number = 0	string	Поточна кількість опрацьованих рядків файла
tokens	vector<string>	Масив токенів поточного рядка
extracted_const_identifiers	vector<string>	Масив змінних поточного рядка
max_value_fn	vector<string>	Масив константних змінних рядка
Клас Lexer		
Структура IdentifierNode		
identifier_name	string	Ім'я поточного ідентифікатора
left	IdentifierNode*	Лівий нащадок
right	IdentifierNode*	Правий нащадок
m_root = nullptr	IdentifierNode*	Корінь дерева
Метод	Опис	
empty	Видаляє всі вузли дерева, залишаючи тільки кореневий вузол	
insert	Вставляє вузол від зазначеного предка	
printBT	Виводить дерево в консоль	
split	Розбиває рядок на токени за зазначеним роздільником	
is_data_type	Перевіряє, чи є рядок типом даних: <i>int</i> , <i>float</i> і т.д.	
erase_identifier_prefix_postfix	Видаляє з рядка-ідентифікатора префіксні (поінтер) та постфіксні (закр. дужка) символи	

Постановка задачі:

Щоб отримати ідентифікатори кожного рядка, читаємо кожен рядок вхідного файлу і розбиваємо його на токени, серед яких і шукаємо ідентифікатори. Щоб знайти звичайну неконстантну змінну серед токенів, необхідно перевірити такі умови:

1. Вона не має містити символ '(', оскільки це б означало, що вона є частиною назви функції;
2. Попереднім токеном має бути її **тип**, а попереднім токеном типу **не має** бути ключове слово *const*.

Щоб знайти серед токунів константну змінну, необхідно перевірити таку умову: токен має бути ключовим словом **const**, або містити його в собі (для випадків, коли константа є першою в сигнатурі функції), тоді назва змінної – або через один (*const a* = ...), або через два (*const int b* = ...), або ж через три (*const char *const c* = ...) токени від поточного.

На кожному читанні рядка з файла, до дерева додаються нові вузли з отриманими ідентифікаторами: отриману неконстантну змінну в самий лівий вузол, отриману константу – в самий правий вузол.

Розв'язання:

Код програми:

Код на C++

Результат виконання:

```
→ build git:(main) x ./lab5.exe  
Enter the file path: ../sample.cpp  
'-[ROOT]  
'-1: [NONE]  
'-2: [NONE]  
'-3: [NONE]  
'-4: [NONE]  
'-5: [NONE]  
'-6: { iter }  
'-7: [NONE]  
'-8: [NONE]  
'-9: [NONE]  
'-10: { _plain }  
'-11: [NONE]  
'-12: { _loc_string }  
'-13: [NONE]  
'-14: [NONE]  
'-15: [NONE]  
'-16: { character } { some_integer } { small_int }  
'-17: [NONE]  
'-18: { _loc_string }  
'-19: [NONE]  
'-20: [NONE]  
'-21: [NONE]  
'-22: [NONE]  
'-23: [NONE]  
'-24: { var_inside_main }  
'-25: [NONE]
```

```
✓ root_lexer: {...}
  ✓ m_root: 0x1695d0
    > identifier_name: "[ROOT]"
    ✓ left: 0x169740
      > identifier_name: "1: [NONE]"
      ✓ left: 0x169850
        > identifier_name: "2: [NONE]"
        ✓ left: 0x1698b0
          > identifier_name: "3: [NONE]"
          > left: 0x169ae0
          > right: 0x0
          > right: 0x0
          > right: 0x0
        ✓ right: 0x1697a0
          > identifier_name: "1: [NONE]"
          > left: 0x0
          ✓ right: 0x169b40
            > identifier_name: "2: { _Global_scoped }"
            > left: 0x0
            ✓ right: 0x169910
              > identifier_name: "3: [NONE]"
              > left: 0x0
              > right: 0x169dd0
```

Висновок:

Під час лабораторної роботи я набув практичних навичок використання структури дерева. У результаті виконання лабораторної роботи я створив клас, що представляє собою обгортку для виконання операцій над структурою дерева та роботи з ідентифікаторами. Визначив методи створення та вставлення цього вузла в дерево.