

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

Кафедра ІІІ

Звіт

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

„ ДЕРЕВА ”

Виконав

ІІІ-15 Костін Вадим
(шифр, прізвище, ім'я, по батькові)

Перевірів(ла)

Вєчерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Київ 2022

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.

Header.h

```
1  #pragma once
2  #include <iostream>
3  #include <string>
4  #include <fstream>
5  #include <iomanip>
6
7  using namespace std;
8
9  struct Tree
10 {
11     string data;
12
13     int line;
14
15     Tree* left;
16     Tree* right;
17
18     Tree(string d, int n, Tree* l = NULL, Tree* r = NULL):data(d), line(n), left(l), right(r) {}
19 };
20
21 Tree* readFile();
22
23 void insert(Tree*&, string, int, int);
24
25 void displayTree(Tree*, int);
```

Func.cpp

```
1  #include "Header.h"
2
3  Tree* readFile()
4  {
5      Tree* root = NULL;
6      ifstream outFile("Program.txt");
7
8      if (!outFile) cout << "Cannot open the file!" << endl;
9
10     cout << "Program:" << endl;
11
12     int i = 1;
13
14     while (!outFile.eof())
15     {
16         string str;
17         string name;
18         bool write = false;
19         getline(outFile, str);
20         cout << setw(2) << i << " " << str << endl;
21         if (str.find("int ") < str.length() && str != "int main()" && str != "int main(){}")
22         {
23             if (str.find('=') < str.length()) name = str.substr(str.find("int ") + 4, str.find('=') - 5 - str.find("int "));
24             else name = str.substr(str.find("int ") + 4, str.length() - 5 - str.find("int "));
25             write = true;
26         }
27         if (str.find("float ") < str.length())
28         {
29             if (str.find('=') < str.length()) name = str.substr(str.find("float ") + 6, str.find('=') - 7 - str.find("float "));
```

```

27     if (str.find("float ") < str.length())
28     {
29         if (str.find('=') < str.length()) name = str.substr(str.find("float ") + 6, str.find('=') - 7 - str.find("float "));
30         else name = str.substr(str.find("float ") + 6, str.length() - 7 - str.find("float "));
31         write = true;
32     }
33     if (str.find("double ") < str.length())
34     {
35         if (str.find('=') < str.length()) name = str.substr(str.find("double ") + 7, str.find('=') - 8 - str.find("double "));
36         else name = str.substr(str.find("double ") + 7, str.length() - 8 - str.find("double "));
37         write = true;
38     }
39     if (str.find("string ") < str.length())
40     {
41         if (str.find('=') < str.length()) name = str.substr(str.find("string ") + 7, str.find('=') - 8 - str.find("string "));
42         else name = str.substr(str.find("string ") + 7, str.length() - 8 - str.find("string "));
43         write = true;
44     }
45     if (str.find("bool ") < str.length())
46     {
47         if (str.find('=') < str.length()) name = str.substr(str.find("bool ") + 5, str.find('=') - 6 - str.find("bool "));
48         else name = str.substr(str.find("bool ") + 5, str.length() - 6 - str.find("bool "));
49         write = true;
50     }
51
52     if (write)
53         if (str.find("const") < str.length()) insert(root, name, i, -1);
54         else insert(root, name, i, 1);
55
56     i++;
57 }
58
59 outFile.close();
60 return root;
61 }

```

```

63 void insert(Tree*& current, string name, int line, int branch)
64 {
65     if (current == NULL) current = new Tree(name, line);
66     else
67     {
68         if (branch < 0) insert(current->right, name, line, branch);
69         else insert(current->left, name, line, branch);
70     }
71 }
72
73 void displayTree(Tree* current, int count = 0)
74 {
75     if (current != NULL)
76     {
77         string space = " ";
78         for (int i = 0; i < count; i++)
79         {
80             space += " ";
81         }
82         displayTree(current->right, count + 1);
83         cout << space << current->line << " " << current->data << endl;
84         displayTree(current->left, count + 1);
85     }
86 }

```

main.cpp

```
1      #include "Header.h"
2
3      int main()
4      {
5          Tree* root = NULL;
6
7          root = readFile();
8
9          cout << "\nTree:" << endl;
10
11         displayTree(root,0);
12
13         cout << endl;
14
15         system("pause");
16     }
```

Результат роботи програми



```
Program:
1 #include <iostream>
2
3 using namespace std;
4
5 const int num = 12;
6
7 int main()
8 {
9     int a;
10    int b = 5;
11    bool c = true;
12    const int N = 1;
13    const double A = 13;
14    cout << a + b + c << endl << num + N + A << endl;
15    system("pause")
16 }

Tree:
    13) A
    12) N
    5) num
    9) a
    10) b
    11) c

Press any key to continue . . .
```