

Лабораторная работа №2. Классы

Вариант №4

Проверил:
ассистент кафедры ИТАС
Прищепчик М. В.

Выполнили: ст. группы 120602
Анашкевич П. С.
Будный Р. И.

1 Цель работы

1. Изучение возможностей языка C++ в определении пользовательских типов данных.
2. Изучение правил определения и переопределения функций доступа к объектам класса.
3. Использование статических элементов классов и дружественных функций.

2 Задание

Определите узел бинарного дерева следующим образом:

```
1 class Node
2 {
3     char name[10];
4     Node * left;
5     Node * right;
6     ...
7 };
```

Определите в классе следующие функции:

Init() – инициализация узла. Функция должна установить указатели на левый и правый узел в ноль;

AddNode() – добавление узла в левую или правую ветви. Если слева нет узла, то добавить слева, если справа нет узла, то добавить справа, иначе ничего не добавлять;

DelTree() – удаление поддеревьев;

Print() – рекурсивная функция вывода дерева на экран.

3 Листинг программы

Listing 1: main.cpp

```
1 #include "tree.h"
2
3 int main() {
4     node * myTree = new node();
5
6     myTree->init("a")->addNode("t")->addNode("e")->addNode("r")->addNode("m")
7         ->addNode("m")->addNode("32")->addNode("46");
8
9     myTree->print();
10    myTree->delTree();
11
12    return 0;
13 }
```

Listing 2: tree.h

```
1 #include "tree.cpp"
```

Listing 3: tree.cpp

```
1 #include <string.h>
2 #include <iostream>
3 #define NAME_LENGTH 10
4
5 using namespace std;
6
7 class node {
8     char name[NAME_LENGTH];
9     node * left;
10    node * right;
11 public:
12    node * init(const char * name);
13    node * addNode(const char * name);
14    void print();
15    void delTree();
16 };
17
18
19 node * node::init(const char * name) {
20     this->left = 0;
21     this->right = 0;
22     strncpy(this->name, name, NAME_LENGTH);
23     return this;
24 }
25
26 node * node::addNode(const char * name) {
27     node * cur = new node;
28     cur->init(name);
29
30     node * parent = this;
31     node * leaf = (strcmp(cur->name, parent->name) > 0 ? parent->right :
32                     parent->left);
33
34     while (leaf != 0) {
35         parent = leaf;
36         if (strcmp(cur->name, parent->name) > 0)
37             leaf = leaf->right;
38         else
39             leaf = leaf->left;
40     }
41     if (strcmp(cur->name, parent->name) > 0)
42         parent->right = cur;
43     else
44         parent->left = cur;
45     return this;
46 }
47
48 void node::print() {
49     if (this != 0) {
```

```
49     if (this->left != 0) this->left->print();
50     cout << this->name << endl;
51     if (this->right != 0) this->right->print();
52 }
53 }
54
55 void node::delTree() {
56     if (this->left != 0)
57         this->left->delTree();
58     if (this->right != 0)
59         this->right->delTree();
60     if (this != 0)
61         delete this;
62 };
```

4 Выводы

В ходе проведения лабораторной работы были изучены возможности языка C++ в определении пользовательских типов данных.