

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЯ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
Кафедра интеллектуальных информационных технологий

Отчёт по лабораторной работе №2
по дисциплине Объектно-Оrientированные Технологии Программирования и Стандарты
Проектирования за I семестр
Тема: "Потоковые классы"

Выполнил:
студент 2 курса
IV семестра
факультета ЭИС
группы ПО-4(1)
Галанин П. И.
«__» _____ 2021 г.

Проверил:
магистрант
кафедры ИИТ
Миндер А. В.
«__» _____ 2021 г.

Отчёт по лабораторной работе №2

Тема: «Потоковые классы»

Цель работы: Научиться программировать ввод и вывод в C++, используя объекты потоковых классов стандартной библиотеки C++.

Ход работы:

Вариант V

Задание 1

Напишите программу, которая печатает слова из файла, расположенные в порядке убывания частоты их появления. Перед каждым словом напечатайте число его появлений.

Программа решения задания

Листинг: main.cpp

```
1 #include <iostream>
2 #include <fstream>
3
4 using namespace std;
5
6 struct Node
7 {
8     static Node* head;
9     Node* next;
10    string word;
11    int counter;
12
13    Node();
14    void add_node(string word);
15    void show();
16    void print();
17    void print_table();
18    void del();
19    ~Node();
20 };
21
22 void option5();
23
24 int main()
25 {
26     option5();
27     return 0;
28 }
29
30 Node::Node()
31 {
```

					ЛР.190333.ПО4.02 81 00					
Изм	Лист	№ докум.	Подп.	Дата	Отчёт по лабораторной работе №2 Потоковые классы			Лит.	Лист	Листов
Разраб.	Галанин							Л	2	5
Пров.	Миндер									
Н. контр.										
Утв.					БрГТУ					

```

32     //cout << this << " construct" << endl;
33     this->word = "-";
34     this->counter = 0;
35     this->next = NULL;
36 }
37
38 void Node::add_node(string word)
39 {
40     for (struct Node* temp = head; temp != NULL; temp = temp->next)
41     {
42         if (word == temp->word)
43         {
44             (temp->counter)++;
45             return;
46         }
47     }
48     struct Node* new_node = new struct Node;
49     new_node->word = word;
50     new_node->counter = 1;
51     new_node->next = head;
52     head = new_node;
53 }
54
55 void Node::show()
56 {
57     cout << "{" << endl
58         << "\t" << "\"this\":" << "\"" << this << "\"" << endl
59         << "\t" << "\"next\":" << "\"" << this->next << "\", " << endl
60         << "\t" << "\"word\":" << "\"" << this->word << "\", " << endl
61         << "\t" << "\"counter\":" << "\"" << this->counter << "\", " << endl
62     << "}" << endl;
63 }
64
65 void Node::print()
66 {
67     for (struct Node* temp = head; temp != NULL; temp = temp->next)
68     {
69         temp->show();
70     }
71 }
72
73 void Node::print_table()
74 {
75     //what length?
76     int length = 0;
77     for (struct Node* temp = head; temp != NULL; temp = temp->next)
78     {
79         length++;
80     }
81     //create array
82     Node* arr = new Node [length];
83     //copy list ro array
84     int i = 0;
85     for (struct Node* temp = head; temp != NULL; temp = temp->next)
86     {
87         arr[i].word = temp->word;
88         arr[i].counter = temp->counter;
89         arr[i].next = NULL;
90         i++;
91     }

```

```

92     //sort array
93     for (int i = 0; i < length; i++)
94     {
95         for (int j = 0; j < length; j++)
96         {
97             if (arr[i].counter > arr[j].counter)
98             {
99                 Node temp = arr[i];
100                 arr[i] = arr[j];
101                 arr[j] = temp;
102             }
103         }
104     }
105     //print sorted table
106     for (int i = 0; i < length; i++)
107     {
108         cout << arr[i].counter << "\t" << " " << arr[i].word << " " << endl;
109     }
110     //delete array
111     delete [] arr;
112 }
113
114 void Node::del()
115 {
116     for (struct Node* temp = head; temp != NULL;)
117     {
118         struct Node* x = temp;
119         temp = temp->next;
120         delete x;
121     }
122 }
123
124 Node::~~Node()
125 {
126     //cout << this << " destruct" << endl;
127 }
128
129 Node* Node::head = NULL;
130
131 void option5()
132 {
133     struct Node myList; //list for word with counter
134     ifstream fin; //for read file
135     fin.open("../files/text.txt"); //open file
136     if(!fin.is_open()) //if file not open then do
137     {
138         cout << "err open" << endl;
139     }
140     else //if file open then do
141     {
142         string word;
143         char ch;
144         while(fin.get(ch))
145         {
146             if (ch == '\u' || ch == '\n' || ch == '\t')
147             {
148                 //cout << word << endl;
149                 if (word != "")
150                 {
151                     myList.add_node(word);

```

```

152         }
153         word = "";
154     }
155     else
156     {
157         word += ch;
158     }
159     //cout << ch;
160 }
161 }
162 //myList.print();
163 myList.print_table();
164 fin.close();
165 myList.del();
166 }

```

Результат работы программы

Листинг: files/text.txt

```

1 apple
2 pear
3 watermelon
4
5     pomelo
6
7     apple
8
9 avocado banana date pear     kiwi
10
11 banana             banana     banana
12
13 apple mango plum avocado

```

Листинг: Out

```

1 4      "banana"
2 3      "apple"
3 2      "pear"
4 2      "avocado"
5 1      "plum"
6 1      "mango"
7 1      "pomelo"
8 1      "watermelon"
9 1      "kiwi"
10 1      "date"

```

Изм	Лист	№ докум.	Подп.	Дата

ЛР.190333.ПО4.02 81 00

Лист

5