

Лабораторная работа 1

Дисциплина: Технологии программирования

Комаров Владимир Артемович

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	10
	Список литературы	11

Список иллюстраций

Список таблиц

1 Цель работы

Изучить структуры данных, научиться писать функции для работы со структурами данных, выработать базовое понимание работы структуры данных

2 Задание

Написать компьютерную программу, содержащую

- Описание структуры, содержащей поля типа `string`, `int`, `double`;
- Набор функций для работы со списком на базе этой структуры:
 - Добавление элемента в начало списка;
 - Добавление элемента в конец списка;
 - Добавление элемента в список после заданного элемента;
 - Добавление элемента в список перед заданным элементом;
 - Удаление из списка элемента с заданным именем;
 - Вывод содержания списка на экран;
- Функцию `main`, содержащую сценарий работы со списком, использующий разработанный инструментарий.

Представить результаты в виде двух файлов:

- Компьютерная программа на C++;
- Отчет о выполнении лабораторной работы с описанием алгоритма и структуры программы.

3 Выполнение лабораторной работы

Импортирую необходимые библиотеки, описываю структуру содержащую необходимые поля, инициализирую все функции, что будут использованы в дальнейшем и создаю функцию, содержащую сценарий работы со списком.

```
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  struct tovar {
7      string name = "";
8      double price = 0;
9      int amount = 0;
10     tovar* next = NULL;
11 };
12
13
14 void addfirst(tovar*& head, string n, double p, int a);
15 void printlist(tovar* head);
16 void addlast(tovar* head, string n, double p, int a);
17 void add_to_num(tovar*& head, int index, string n, double p, int a);
18 void add_before_num(tovar*& head, int index, string n, double p, int a);
19 void add_after_num(tovar*& head, int index, string n, double p, int a);
20 int deletelist(tovar*& head, string nm);
21
22 int main() {
23     setlocale(LC_ALL, "ru");
24
25     tovar* sklad = NULL;
26     addfirst(sklad, "cake", 57.25, 30);
27     addfirst(sklad, "tea", 27.5, 200);
28     addfirst(sklad, "coffe", 90, 70);
29     addfirst(sklad, "limonade", 60, 100);
30
31     addlast(sklad, "woter", 30, 40);
32     add_before_num(sklad, 3, "pivo", 100, 57);
33     add_after_num(sklad, 4, "vodka", 500, 20);
34
35     deletelist(sklad, "coffe");
36
37     printlist(sklad);
38
39     return 0;
40 }
```

Создаю функцию добавления элемента в начало списка и вывода всего списка товаров.

```

42 void addfirst(tovar* & head, string n, double p, int a) {
43     tovar* tmp = new tovar;
44     tmp->name = n;
45     tmp->price = p;
46     tmp->amount = a;
47     tmp->next = head;
48     head = tmp;
49     return;
50 }
51
52 void printlist(tovar* head) {
53     if (head == NULL) {
54         cout << "Пустой список \n";
55         return;
56     }
57     else {
58         tovar* tmp = head;
59
60         do {
61             cout << "Товар: " << tmp->name << "\n";
62             cout << "Цена: " << tmp->price << "\n";
63             cout << "Количество: " << tmp->amount << "\n" << "\n";
64             tmp = tmp->next;
65         } while (tmp != NULL);
66         return;
67     }
68 }
69 }

```

Создаю функцию удаления из списка по названию.

```

int deletelist(tovar* & head, string nm) {
    if (head == NULL) return 1;
    tovar* prev = head;

    if (prev->name == nm) {
        head = head->next;
        delete prev;
        return 0;
    }

    tovar* tmp = head->next;

    while (tmp != NULL and tmp->name != nm) {
        tmp = tmp->next;
        prev = prev->next;
    }

    if (tmp == NULL) return 1;

    prev->next = tmp->next;
    delete tmp;
    return 0;
}

```

Создаю функцию добавления элемента в конец списка и функцию, которую буду использовать в дальнейшем по добавлению элемента на определённое место


```

void addlast(tovar* head, string n, double p, int a) {
    tovar* last = head;
    tovar* tmp = new tovar;
    tmp->name = n;
    tmp->price = p;
    tmp->amount = a;

    while (last->next != NULL) {
        last = last->next;
    }

    last->next = tmp;
}

void add_to_num(tovar*& head, int index, string n, double p, int a) {
    if (index == 1) {
        addfirst(head, n, p, a);
        return;
    }

    tovar* prev = head;
    tovar* tmp = new tovar;
    tmp->name = n;
    tmp->price = p;
    tmp->amount = a;

    int counter = 1;

    while (index != counter and prev->next != NULL) {
        prev = prev->next;
        counter += 1;
    }

    tmp->next = prev->next;
    prev->next = tmp;
    return;
}

```

Создаю функцию добавления до заданного элемента и функцию добавления после заданного элемента с помощью ранее созданной функции добавления на определённое место.

```

void add_before_num(tovar*& head, int index, string n, double p, int a) {
    add_to_num(head, index - 1, n, p, a);

    return;
}

void add_after_num(tovar*& head, int index, string n, double p, int a) {
    add_to_num(head, index + 1, n, p, a);

    return;
}

```

4 Выводы

Я изучил структуры данных, научился писать функции для работы с ними, а так же выработал понимание работы со структурами данных.

Список литературы