Лабораторная работа №1

Создание списка продуктов

Демидова Екатерина Алексеевна

Содержание

# 1 Цель работы

Написать программу на C++ которая реализует работу со списком элементов(товаров).

# 2 Задание

Каждый элемент (товар) должен иметь следующую структуру:

* Название товара.
* Цена товара.
* Количество

Программа должна предлагать пользователю на выбор несколько действий

* Добавить новый товар во списка.
* добавить новый товар и конец списка.
* Добавить новый гобар перед существующим выбранным товаром в списке.
* Добавить новый товар после существующего товара в списке.
* Удалить определенный товар из списка.
* Вывести на экран информацию о выбранном товаре.
* Вывести на экран весь список целиком.
* Выйти из программы.

Программа должна работать до тех пор, пока пользователь не выберет опцию “Выйти из программы” Необходимо реализовать следующие функции

* Создание нового элемента (товара) с помощью ввода с клавиатуры цены и количества
* Добавление созданного элемента в список
  + В начало списка
  + В конец списка
  + После определенного элемента
  + Перед определенным элементом
* Удаление выбранного мента.
* Вывод информации об элементе.
* Вывод информации о всех элементах списка.

# 3 Выполнение проекта

Для начала была реализована структура узла, в которой name - переменная наименование товара, p -цена товара, amount - количество товаров, next - ссылка на следующий элемент.

struct Product{  
 string name;  
 float price;  
 int amount;  
 Product \*next;  
};

Также была реализована функция добавления элемента в начало списка и печати списка:

void add\_first (Product \*&p\_product, string new\_name, float p, int a){  
 Product \*new\_product = new Product;  
 new\_product ->name = new\_name;   
 new\_product ->price = p;  
 new\_product ->amount = a;  
 new\_product ->next = p\_product;  
 p\_product = new\_product;  
};  
  
  
void print\_list(Product \*p\_product){  
 if(p\_product == NULL){  
 cout << "The list is empty"; return;  
 };  
 Product \*tmp = p\_product;  
 while (tmp != NULL){  
 cout << "Product: "<< tmp->name <<"\nPrice:"<< tmp->price <<"\nAmount:"<<tmp->amount<<"\n\n";  
 tmp = tmp->next;  
 };  
 return;  
};

Приведём пример использования этих функций. (рис. 1)

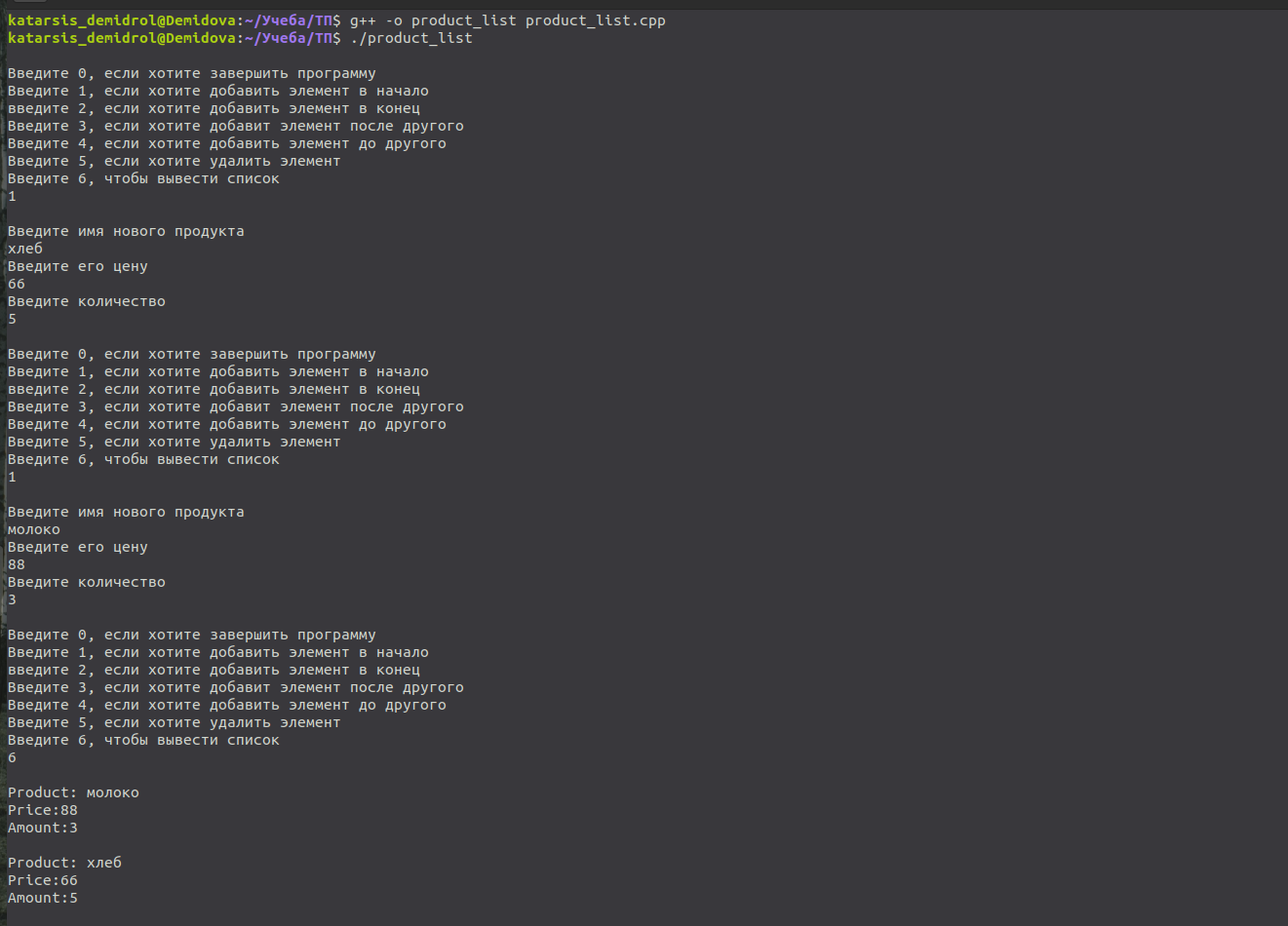


Рис. 1: Добавление элемента в начало

Кроме того, была написана функция для добавления в конец списка. Если в списке нет элементов, то вызывается функция добавления в начало.

Product \*tmp = p\_product;  
 if (p\_product == NULL){  
 add\_first(p\_product, new\_name, p, a);  
 return;  
 }  
 while (tmp ->next) tmp = tmp->next;  
 Product \*new\_product = new Product;  
 new\_product->name = new\_name;  
 new\_product->price = p;  
 new\_product->amount = a;  
 new\_product->next = NULL;  
 tmp->next = new\_product;  
 return;  
};

Также была создана функции для удаления элемента. Ей в аргументы поступает голова списка и имя элемента, который необходимо удалить.

int del\_product(Product \*&p\_product, string del\_name){  
 if (p\_product = NULL) return 1;  
 Product \*tmp = p\_product;  
 if (p\_product->name == del\_name){  
 p\_product = p\_product->next;   
 delete tmp;   
 return 0;  
 }  
 if (p\_product->next == NULL) return 1;  
 Product \*prev = p\_product; tmp = p\_product->next;  
 while (tmp->next != NULL && tmp->name != del\_name){  
 prev = prev->next;   
 tmp = tmp->next;  
 }  
 if (tmp->next == NULL && tmp->name!=del\_name) return 1;   
 prev->next = tmp->next;  
 delete tmp;  
 return 0;  
};

Были реализованы функции добавления до определённого элемента и после:

int add\_after (Product \*p\_product, string new\_name, float p, int a, string name\_after){  
 if (p\_product == NULL) return 1;  
 Product \*tmp = p\_product;  
 while (tmp != NULL && tmp->name != name\_after){  
 tmp = tmp->next;  
 }  
 if (tmp == NULL) return 1;  
 Product \*new\_product = new Product;  
 new\_product->name = new\_name;  
 new\_product->price = p;  
 new\_product->amount = a;  
 new\_product->next = tmp->next;  
 tmp->next = new\_product;  
 return 0;  
};  
  
int add\_before(Product \*&p\_product, string new\_name, float p, int a, string name\_before){  
  
 if (p\_product == NULL) return 1;  
 if (p\_product->name == name\_before){  
 add\_first(p\_product, new\_name, p, a);  
 return 0;  
 }  
 if (p\_product->next == NULL) return 1;  
 Product \*prev = p\_product, \*tmp = p\_product->next;  
 while (tmp != NULL && tmp->name != name\_before){  
 prev = prev->next;   
 tmp = tmp->next;  
 }  
 if (tmp->next == NULL && tmp -> name!=name\_before) return 1;  
 Product \*new\_product = new Product;  
 new\_product->name = new\_name;  
 new\_product->price = p;  
 new\_product->amount = a;  
 new\_product->next = tmp;  
 prev->next = new\_product;  
 return 0;  
};

К ним на вход подаются голова списка, имя нового товара, его цена и количество, а также имя товара до/после которого добавиться новый элемент. (рис. 2, 3)

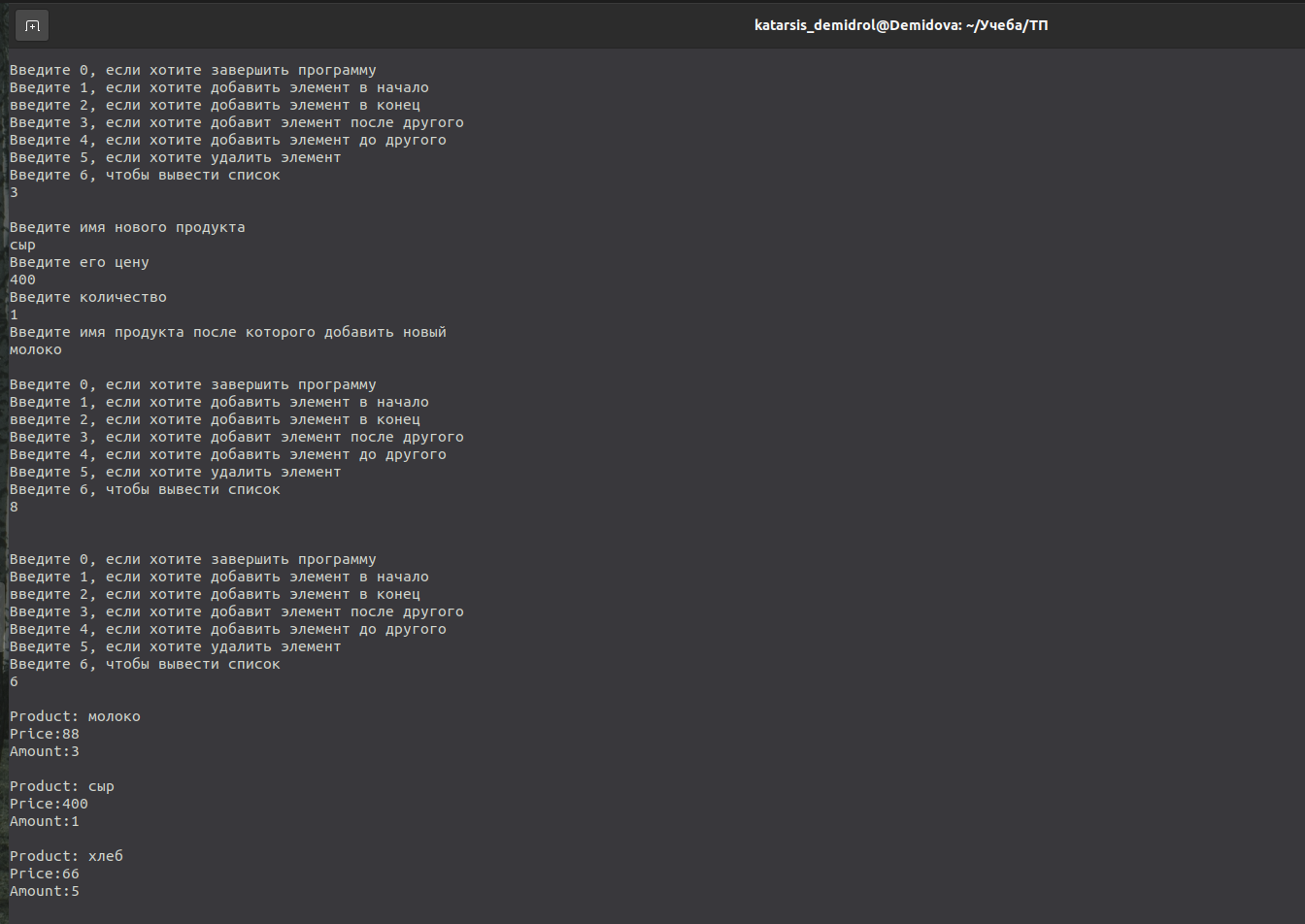


Рис. 2: Добавление элемента после другого

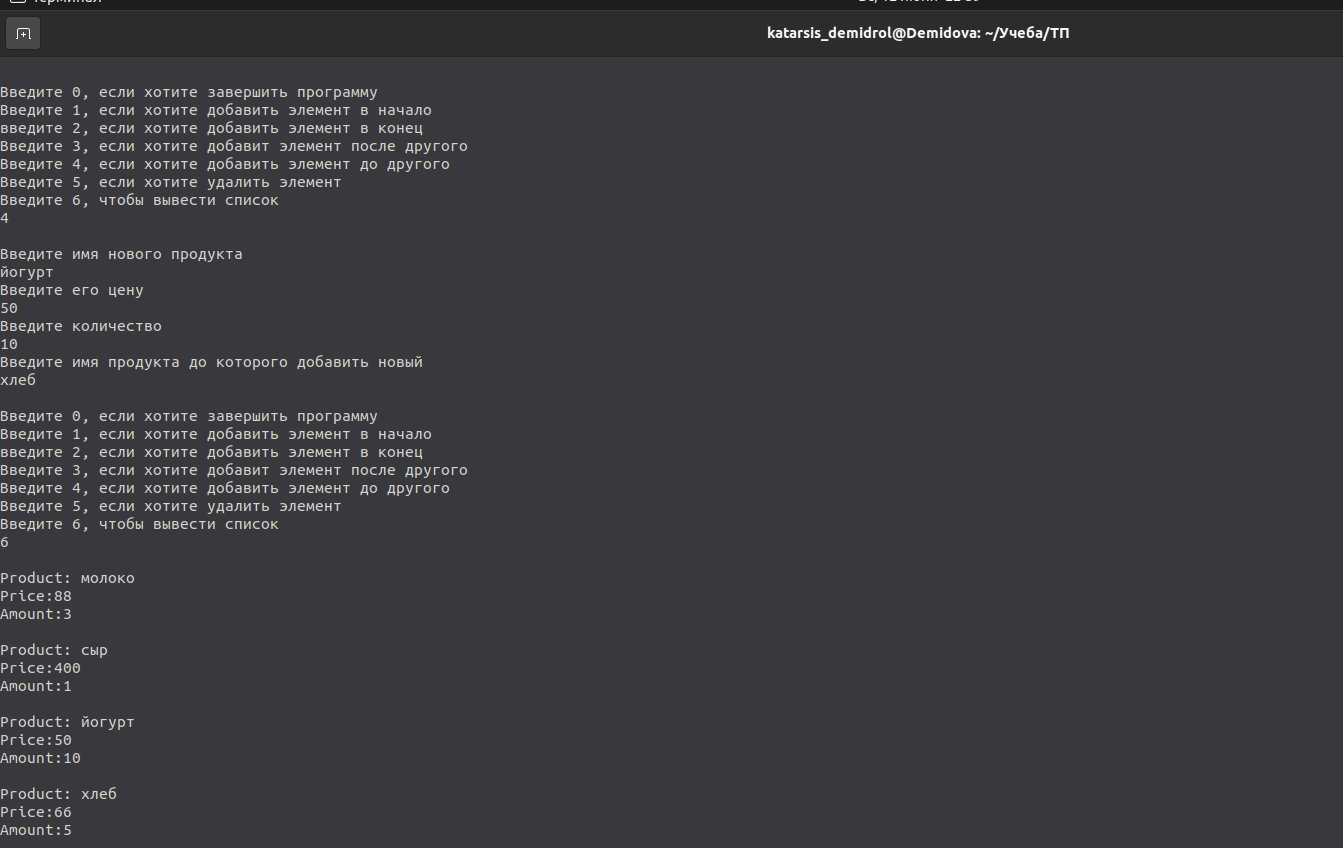


Рис. 3: Добавление элемента до другого

# 4 Выводы

В результате выполнения лабораторной работы были получены практические навыки работы со списками, была написана программа на языке C++, в которой реализован список товаров, с помощью неё можно добавлять товары(в конец, в начало, до и после какого-то товара).