Министерство образования и науки РФ

Федеральное государственное автономное образовательное учебное учреждение высшего образования “Санкт-Петербургский национальный исследовательский университет ИТМО”

Факультет программной инженерии и компьютерной техники

Лабораторная работа №1

По дисциплине: Программирование на C++

Выполнил:

Ситдиков Рафаэль Ильдусович

Группа: P3131

Проверил:

Г. Санкт-Петербург, 2025г.

Задание и критерия:

**Цель работы:**

* Научиться использовать функции.
* Освоить управление динамической памятью.
* Изучить базовые шаблоны функций.
* Закрепить работу с десятичными числами (float / double) и строками.

**Условие задачи:**

Создайте программу, которая рассчитывает итоговую стоимость заказа в кафе с учётом количества, скидки и НДС. Пользователь вводит блюда, их цены и количество. Программа рассчитывает итоговую сумму, применяет скидку, добавляет налог и выводит чек.

**Обязательные элементы:**

* Использовать функции для:
  + ввода данных
  + расчёта стоимости
  + вывода чека
* Использовать динамическую память (new / delete) для хранения массива заказов.
* Применить шаблонную функцию для универсального вывода элементов (например, информации о товаре).
* Использовать строки (std::string) для хранения названий блюд.
* Использовать десятичные числа (float или double) для расчётов.

**Пример интерфейса программы:**

Введите количество блюд: 2

Блюдо 1:

Название: Кофе

Цена: 120.5

Количество: 2

Блюдо 2:

Название: Пирог

Цена: 250.0

Количество: 1

Введите скидку в процентах: 10

=== Чек ===

Кофе (2 шт): 241.00

Пирог (1 шт): 250.00

Скидка: -49.10

НДС (20%): 88.38

Итог: 530.28

**Подсказки по реализации:**

**Структура для блюда:**

struct Dish {

std::string name;

double price;

int quantity;

};

**Шаблон для вывода:**

template<typename T>

void printInfo(const T& item) {

std::cout << item << std::endl;

}

**Пример динамического выделения:**

Dish\* menu = new Dish[amount];

// ...

delete[] menu;

**Критерии оценки:**

* Корректность расчётов
* Использование функций
* Использование динамической памяти
* Шаблон хотя бы одной функции
* Работа со строками (std::string)
* Аккуратный вывод чека

Код программы:

#include <iostream>  
#include <vector>  
#include <windows.h>  
#include <fstream>  
  
using namespace std;  
  
// Структура для хранения данных о блюде  
struct Dish {  
 string name;  
 float sale;  
 int quantity;  
};  
  
struct Menu {  
 string name;  
 float sale;  
 int discount;  
};  
  
void save\_receipt(const string& one) {  
 // Чтение k из файла  
 ifstream num("receipt//count.txt");  
 int k = 0;  
  
 if (num.is\_open()) {  
 string k\_str;  
 getline(num, k\_str);  
 if (!k\_str.empty()) {  
 k = stoi(k\_str);  
 }  
 num.close();  
 }  
  
 // Создание имени файла  
 string filename = "receipt//receipt" + to\_string(k) + ".txt";  
  
 // Запись чека в файл  
 ofstream out(filename);  
 if (out.is\_open()) {  
 out << one << endl;  
 out.close();  
 cout << "\n Чек записан в файл: " << filename << endl;  
 } else {  
 cout << "\n Ошибка при записи файла: " << filename << endl;  
 }  
  
 // Увеличиваем k и сохраняем обратно  
 ofstream out\_k("receipt//count.txt");  
 if (out\_k.is\_open()) {  
 out\_k << (k + 1);  
 out\_k.close();  
 }  
}  
  
void load\_XML\_file() {  
 ifstream xml("receipt//users.xml");  
 int k = 0;  
  
  
}  
  
  
//Создаём перечень меню  
void menu\_dynamic() {  
 int k;  
 //vector<Menu> main\_menu;  
 float main\_food = 0;  
 int sale;  
 string food;  
 vector<pair<string, float>> main\_menu = {  
 {"Стейк", 2200},  
 {"Луковый суп", 900},  
 {"Капучино 200мл", 160},  
 {"Омлет", 400},  
 {"Тар-Тар", 900},  
 {"Торт Наполеон клубничный", 950},  
 {"Шампанское De Cruello", 6000}  
 };  
  
  
  
 vector<pair<string, float>> ordered;  
  
 cout << "----- Меню -----" << endl;  
 for (int i = 0; i < main\_menu.size(); i++) {  
 cout << (i + 1) << ". " << main\_menu[i].first << " - " << main\_menu[i].second << " руб." << endl;  
 }  
  
 while (true) {  
 cout << "Введите номер блюда для заказа (0 для завершения): ";  
 cin >> k;  
  
 if (k == 0) {  
 cout << "Заказ завершён \n";  
 break;  
 }  
  
 if (k >= 1 && k <= main\_menu.size()) {  
 main\_food += main\_menu[k - 1].second;  
 ordered.push\_back(main\_menu[k - 1]);  
 cout << "Добавлено: " << main\_menu[k - 1].first << " за " << main\_menu[k - 1].second << " руб." << endl;  
 } else {  
 cout << "Некорректный номер блюда, попробуйте снова.\n";  
 }  
 }  
  
 float main\_food1 = main\_food;  
 main\_food += main\_food / 5;  
  
 int dis;  
 {  
 int discount = 10;  
 cout << "Ваша скидка - " << discount << "%\n";  
 cout << "У вас есть скидка? ";  
 string ask;  
 cin >> ask;  
  
 if (ask == "Да" || ask == "Yes" || ask == "yes" || ask == "да") {  
 int discount\_ask;  
 cout << "Введите свою скидку в рублях: ";  
 cin >> discount\_ask;  
 if (main\_food - (main\_food / 100) \* discount < main\_food - discount\_ask) {  
 main\_food = main\_food - (main\_food / 100) \* discount;  
 dis = discount;  
 } else {  
 main\_food = main\_food - discount\_ask;  
 if (main\_food < 0) {  
 main\_food = 0;  
 dis = discount\_ask;  
 }  
 dis = discount\_ask;  
 }  
 } else {  
 dis = discount;  
 main\_food = main\_food - (main\_food / 100) \* discount;  
 }  
 }  
  
 if (main\_food >= 4000 && main\_food <= 10000) {  
 main\_food -= main\_food \* 0.10;  
 } else if (main\_food > 10000 && main\_food <= 20000) {  
 main\_food -= main\_food \* 0.13;  
 } else if (main\_food > 20000) {  
 main\_food -= main\_food \* 0.18;  
 }  
  
 cout << "Сумма заказа с учётом скидки: " << main\_food << " руб.\n";  
  
 cout << "Сохранить чек? - ";  
 string ask\_receipt;  
 cin >> ask\_receipt;  
  
 if (ask\_receipt == "Yes" || ask\_receipt == "yes" || ask\_receipt == "Да" || ask\_receipt == "да") {  
 string one;  
 one += "\n-------------Чек:--------------- \n";  
 one += "|--Блюдо--|--Цена--|\n";  
  
 for (const auto& dish : ordered) {  
 one += dish.first + " -- " + to\_string(dish.second) + "\n";  
 }  
  
 one += "Скидка - " + to\_string(dis) + "\n";  
 one += "НДС (20%) - " + to\_string(main\_food1) + " --> " + to\_string(main\_food) + "\n";  
 one += "Итоговая сумма - " + to\_string(main\_food) + "\n";  
 one += "--------------------------------\n";  
  
 save\_receipt(one);  
 }  
}  
  
  
// Основная функция ввода и вывода блюд  
void sum\_cafe() {  
 int k;  
 float total = 0;  
  
 cout << u8"Введите количество блюд: ";  
 cin >> k;  
 cin.ignore(); // Удаляем '\n' после cin >> k  
  
 vector<Dish> menu;  
  
 for (int i = 1; i <= k; i++) {  
 Dish dish;  
  
 cout << "\n" << u8"Блюдо " << i << ":\n";  
  
 // Сначала читаем название (с пробелами)  
 cout << u8"Введите название блюда: ";  
 getline(cin, dish.name);  
  
 // Затем цену  
 cout << u8"Введите цену: ";  
 cin >> dish.sale;  
  
 // Затем количество  
 cout << u8"Введите количество: ";  
 cin >> dish.quantity;  
 cin.ignore(); // удаляем '\n' перед следующим getline  
  
 // Добавляем блюдо в список  
 menu.push\_back(dish);  
  
 // Увеличиваем общую сумму  
 total += dish.sale \* dish.quantity;  
 }  
  
 //НДС 20%  
 int total\_Tax = total + (total / 100) \* 20;  
 int total\_Tax1 = total\_Tax;  
  
 int dis;  
 {  
 //Скидка магазина  
 int discount;  
 discount = 10;  
 cout << u8"Ваша скидка - " << discount << "%" << "\n";  
  
 //Скидка клиента  
 cout << u8"У вас есть скидка? ";  
 string ask;  
 cin >> ask;  
  
 //Вопрос по скидке. Если да, то сравниваем, если нет, то пропускаем.  
 if (ask == u8"Да" or ask == u8"Yes" or ask == u8"yes" or ask == u8"да") {  
 int discount\_ask;  
 cout << u8"Введите свою скидку в рублях: ";  
 cin >> discount\_ask;  
 cout << u8" В случае, если ваша скидка будет выгоднее нашей, \n "  
 "ваш заказ будет обработан только по вашей скидке. \n "  
 "В ином случае будет использована только наша скидка \n";  
  
 //Сравнение скидок  
 if (total\_Tax - (total\_Tax / 100) \* discount < total\_Tax - discount\_ask) {  
 total\_Tax = total\_Tax - (total\_Tax / 100) \* discount;  
 dis = discount;  
 }  
 else {  
 total\_Tax = total\_Tax - discount\_ask;  
 dis = discount\_ask;  
 }  
 }  
 else {  
 dis = discount;  
 }  
 }  
  
 //Дополнительная скидка при высоком чеке 10%  
 if (total\_Tax >= 4000 && total\_Tax <= 10000) {  
 total\_Tax -= total\_Tax \* 0.10; // скидка 10%  
 }  
 else if (total\_Tax > 10000 && total\_Tax <= 20000){  
 total\_Tax -= total\_Tax \* 0.13; // скидка 13%  
 }  
 else if (total\_Tax > 20000){  
 total\_Tax -= total\_Tax \* 0.18; // скидка 18%  
 }  
  
 // Вывод всех блюд  
 cout << u8"\n-------------Чек:--------------- \n";  
 cout << u8"|--Блюдо--|--Цена--|--Кол-во--|\n";  
  
 for (int i = 0; i < menu.size(); i++) {  
 cout << menu[i].name << " -- " << menu[i].sale << " -- " << menu[i].quantity << endl;  
 }  
 cout << u8"Скидка - " << dis << "\n";  
 cout << u8"НДС (20%) - " << total << " --> " << total\_Tax1 << "\n";  
 cout << u8"Итоговая сумма - " << total\_Tax;  
 cout << "\n" << u8"--------------------------------";  
  
 //Работа с файлом чека  
 cout << "Сохранить чек? - ";  
 string ask\_receipt;  
 cin >> ask\_receipt;  
  
 if (ask\_receipt == "Yes" or ask\_receipt == "yes" or ask\_receipt == "Да" or ask\_receipt == "да") {  
 // Запись итоговых данных  
 string one;  
 one += "\n-------------Чек:--------------- \n";  
 one += "|--Блюдо--|--Цена--|--Кол-во--|\n";  
  
 for (int i = 0; i < menu.size(); i++) {  
 one += menu[i].name + " -- " + to\_string(menu[i].sale) + " -- " + to\_string(menu[i].quantity) + "\n";  
 }  
  
 one += "Скидка - " + to\_string(dis) + "\n";  
 one += "НДС (20%) - " + to\_string(total) + " --> " + to\_string(total\_Tax1) + "\n";  
 one += "Итоговая сумма - " + to\_string(total\_Tax) + "\n";  
 one += "--------------------------------\n";  
  
 save\_receipt(one);  
 }  
}  
  
int login() {  
 //Авторизация и разделение прав доступа  
 int password = 0;  
 cout << "Авторизуйтесь, если вы работник - ";  
 cin >> password;  
 vector<int> passwords = {1111, 1342, 2222, 1393, 9979, 6261, 8433};  
 if (password != 0) {  
 for (int p : passwords) {  
 if (password == p) {  
  
 int ask\_stop;  
 cout << u8"Если хотите выйти, введите ноль -";  
 cin >> ask\_stop;  
  
 if (ask\_stop == 0) {  
 return 0;  
 }  
  
 cout << "Хотите что-то заказать?";  
 string new\_ask;  
 cin >> new\_ask;  
 if (new\_ask == u8"Да" or new\_ask == u8"Yes" or new\_ask == u8"yes" or new\_ask == u8"да") {  
 sum\_cafe();  
 }  
 else if (new\_ask == u8"Нет" or new\_ask == u8"No" or new\_ask == u8"no" or new\_ask == u8"нет"){  
 cout << "Преступайте к редактированию \n";  
 menu\_dynamic();  
  
  
 }  
 else {  
 cout << "Неверный пароль";  
 }  
 return 0;  
 }  
 }  
 }  
 else {  
 sum\_cafe();  
 }  
}  
  
// Точка входа  
int main() {  
 SetConsoleOutputCP(CP\_UTF8); // Вывод в UTF-8  
 SetConsoleCP(CP\_UTF8); // Ввод в UTF-8  
  
 int k;  
 cin >> k;  
 if (k == 0) {  
 menu\_dynamic();  
 }  
 else {  
 sum\_cafe();  
 }  
  
 return 0;  
}

Также доступен github: <https://github.com/raf-me/-C-/tree/main/lab1/ITMO>

Файлы сохранения

count.txt:

|  |
| --- |
| 1 |

receipt.txt:

|  |
| --- |
| -------------Чек:---------------  |--Блюдо--|--Цена--| Капучино 200мл -- 160.000000 Тар-Тар -- 900.000000 Капучино 200мл -- 160.000000 Стейк -- 2200.000000 Скидка - 1000 НДС (20%) - 3420.000000 --> 3104.000000 Итоговая сумма - 3104.000000 -------------------------------- |

users.xml:

<?xml version="1.0" encoding="UTF-8" ?>  
<Menu>  
 <main\_menu>  
 <name>"Стейк"</name>  
 <sale>2200.0</sale>  
 <discount>0</discount>  
 </main\_menu>  
  
 <main\_menu>  
 <name>"Луковый суп"</name>  
 <sale>900.0</sale>  
 <discount>7</discount>  
 </main\_menu>  
  
 <main\_menu>  
 <name>"Капучино 200м"</name>  
 <sale>160.0</sale>  
 <discount>0</discount>  
 </main\_menu>  
  
 <main\_menu>  
 <name>"Омлет"</name>  
 <sale>400.0</sale>  
 <discount>15</discount>  
 </main\_menu>  
  
 <main\_menu>  
 <name>"Торт Наполеон клубничный"</name>  
 <sale>950.0</sale>  
 <discount>0</discount>  
 </main\_menu>  
</Menu>

Заключение

В данной лабораторной работе я научился создавать и использовать функции, работать с памятью и её блоками, запись в файл и чтение файла, использовать библиотеки, работать с u8-строками через windows.h.