

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«МИРЭА – Российский технологический университет»**

**РТУ МИРЭА**

|  |
| --- |
|  |

Институт искусственного интеллекта

Кафедра общей информатики

**ОТЧЕТ**

**ПО ПРАКТИЧЕСКОЙ РАБОТЕ №4**

**по дисциплине**

«Структуры и алгоритмы обработки данных»

Тема. Структуризация многоэлементных структур средствами struct

Выполнил студент группы ИКБО-13-21 Смольников А.Б.

Принял

старший преподаватель Скворцова Л.А.

Москва 2022

Вариант №20

Касса магазина. Структура записи операции по кассе: номер кассы, код товара, количество товара, цена товара, процентная скидка на товар, сумма за товар с учетом скидки.

Операции:

Заполнение записи по одной операции с клавиатуры.

Вставить записи по отдельной операции, проведенной кассой, в таблицу, располагая ее в начале списка других операций, проведенных этой кассой.

Вывести список операций, проведенных заданной кассой, с указанием стоимости покупки по каждой операции.

Удалить записи по операциям, проведенным определенной кассой.

**Задание 1.** Разработать набор операций для управления таблицей, созданной на основе статического массива. Структура записи элемента таблицы определена вариантом индивидуального задания.

**Задание 2**. Разработать набор операций для управления таблицей, созданной на основе динамического массива. Структура записи элемента таблицы определена вариантом индивидуального задания.

Требования по выполнению задания

**Задание 3**. Реализовать таблицу по заданию варианта, используя контейнер <vector>, в отчет включить только код и результаты тестирования на векторе.

1. **Разработка программы задачи 1**
   1. **Постановка задачи**

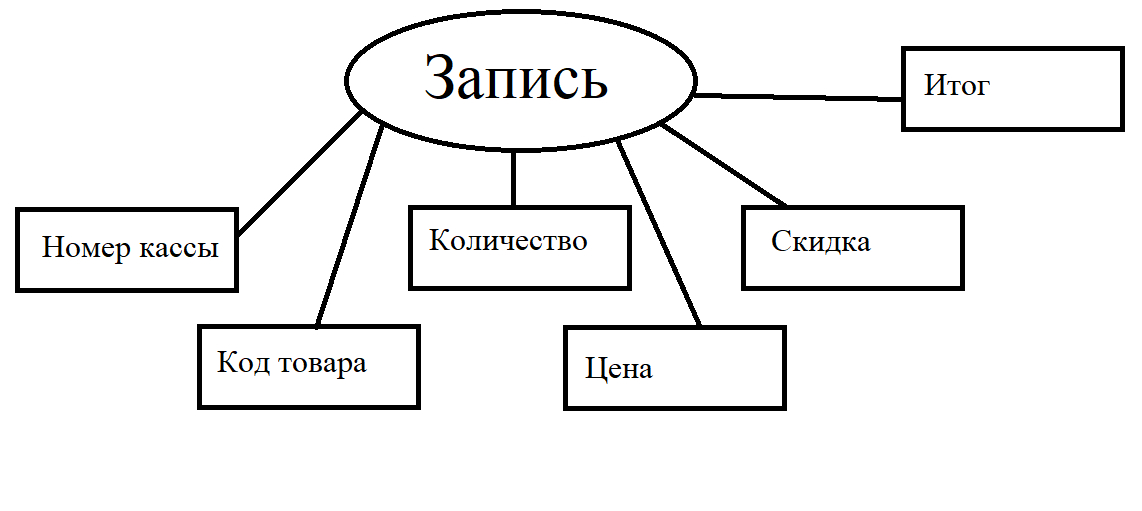
Дано. Статический массив структур размером 100

Результат. Заполненная по желанию пользователя структура

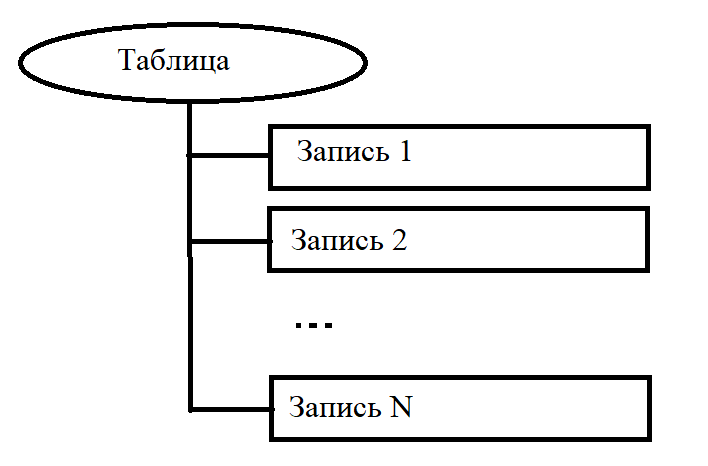
Ограничения. Ввод данных согласно заданному типу данных

* 1. **Определение модели решения**

Определим структуру записи данных:



Определим структуру всей таблицы:



Структура записи и структура таблицы средствами C++

Запись:

struct regRow {

int id;

int barcode;

int quantity;

double price;

double discount;

double total;

};

Таблица:

regRow table[N];

Подзадачи первой задачи:

* Вывод таблицы
* Вывод таблицы заданной кассы
* Заполнение записи
* Добавление записи в конец
* Вставка записи определенной кассы перед первой операцией этой кассы
* Удаление записей заданной кассы

Прототипы функций выделенных подзадач

Вывод таблицы

Предусловие: arr – массив записей, n – текущее количество записей

Постусловие: выведенная на экран таблица

void printTable(regRow\* arr,int n);

Вывод таблицы по id кассы

Предусловие: arr – массив записей, n – текущее количество записей

Постусловие: выведенная на экран таблица заданной кассы

void printTableID(regRow\* arr,int n);

Заполнение записи

Предусловие: row – пустая запись

Постусловие: Заполненная пользователем запись

regRow fillRow(regRow row);

Добавление записи в конец таблицы

Предусловие: arr – массив записей, n – текущее количество записей

Постусловие: добавленная в конец таблицы запись

void insert(regRow\* arr,int n);

Вставка записи определенной кассы

Предусловие: arr – массив записей, n – текущее количество записей

Постусловие: вставленная в таблицу запись

void insert(regRow\* arr, int& n);

Удаление всех записей определенной кассы

Предусловие: arr – массив записей, n – текущее количество записей

Постусловие: Таблица с удаленными записями определенной кассы

void deleteID(regRow\* arr, int& n);

Код программы для статической таблицы:

#include "Header.h"  
#include <iostream>  
#include <vector>  
using namespace std;  
const int N = 100;  
  
  
void printTable(regRow\* arr, int n) {  
 cout << "ID | Barcode | Quantity | Price | Discount | Total\n";  
 for (auto i = 0; i < n; i++) {  
 cout << arr[i].id << " " << arr[i].barcode << " " << arr[i].quantity << " " << arr[i].price << " " << arr[i].discount << " " << arr[i].total << endl;  
 }  
}  
  
void printTableID(regRow\* arr, int n) {  
 cout << "Введите ID кассы для вывода: \n";  
 int id;  
 cin >> id;  
  
 cout << "ID | Barcode | Quantity | Price | Discount | Total\n";  
 for (auto i = 0; i < n; i++) {  
 if(arr[i].id = id)  
 cout << arr[i].id << " " << arr[i].barcode << " " << arr[i].quantity << " " << arr[i].price << " " << arr[i].discount << " " << arr[i].total << endl;  
 }  
}  
  
  
regRow fillRow(regRow row) {  
 cout << "Введите id кассы, код товара, количество, цену и скидку" << endl;  
 cin >> row.id;  
 cin >> row.barcode;  
 cin >> row.quantity;  
 cin >> row.price;  
 cin >> row.discount;  
 row.total = row.price \* row.quantity \* (1 - (row.discount / 100));  
 return row;  
}  
  
//==================================STATIC=======================================  
  
void addRow(regRow\* arr, int& n) {  
 n++;  
 if (n == N) {  
 return;  
 }  
 arr[n-1] = fillRow(arr[n-1]);  
}  
  
void insert(regRow\* arr, int& n) {  
 int id;  
 cout << "Введите ID кассы для вставки: \n";  
 cin >> id;  
 regRow temp{};  
 temp = fillRow(temp);  
  
 for (auto i = 0; i < n; i++) {  
 if (arr[i].id == id) {  
 n++;  
 for (auto j = i; j < n; j++) {  
 swap(arr[j], temp);  
 }  
 return;  
 }  
 }  
}  
  
void deleteID(regRow\* arr, int& n) {  
 cout<<"Введите ID кассы для удаления: \n";  
 int id;  
 cin >> id;  
 for (auto i = 0; i < n; i++) {  
 if (arr[i].id == id) {  
 n--;  
 for (int j = i; j < n + 1; j++) {  
 swap(arr[j], arr[j + 1]);  
 }  
 i--;  
 }  
 }  
}

ЗАДАНИЕ 2

Разработать набор операций для управления таблицей, созданной на основе динамического массива. Структура записи элемента таблицы определена вариантом индивидуального задания.

Требования по выполнению задания

1. Создать новый проект в составе решения задания 1.
2. Выполнить все требования задания 1, только для динамической таблицы.
3. Скопировать функции, которые можно использовать без изменения для динамической таблицы.
4. Изменить функции вставки и добавления записей в динамическую таблицу.
5. Алгоритм формирования списка (указано в некоторых вариантах), выполнить на динамическом массиве.
6. Разработать алгоритм выполнения программы, демонстрирующей выполнение операций над таблицей по разработанным тестам. В программе определить таблицу через статический массив. Размер массива определить через константу (пусть размер будет N=100).
7. Разработать набор тестовых данных по наполнению таблицы. Определить тесты для текущего размера таблицы n<N.

Структура записи и структура таблицы средствами C++

Запись:

struct regRow {

int id;

int barcode;

int quantity;

double price;

double discount;

double total;

};

Таблица:

regRow\* table = new regRow;

Код задачи:

void printTable(regRow\* arr, int n) {  
 cout << "ID | Barcode | Quantity | Price | Discount | Total\n";  
 for (auto i = 0; i < n; i++) {  
 cout << arr[i].id << " " << arr[i].barcode << " " << arr[i].quantity << " " << arr[i].price << " " << arr[i].discount << " " << arr[i].total << endl;  
 }  
}  
  
void printTableID(regRow\* arr, int n) {  
 cout << "Введите ID кассы для вывода: \n";  
 int id;  
 cin >> id;  
  
 cout << "ID | Barcode | Quantity | Price | Discount | Total\n";  
 for (auto i = 0; i < n; i++) {  
 if(arr[i].id = id)  
 cout << arr[i].id << " " << arr[i].barcode << " " << arr[i].quantity << " " << arr[i].price << " " << arr[i].discount << " " << arr[i].total << endl;  
 }  
}  
  
  
regRow fillRow(regRow row) {  
 cout << "Введите id кассы, код товара, количество, цену и скидку" << endl;  
 cin >> row.id;  
 cin >> row.barcode;  
 cin >> row.quantity;  
 cin >> row.price;  
 cin >> row.discount;  
 row.total = row.price \* row.quantity \* (1 - (row.discount / 100));  
 return row;  
}

//==================================DYNAMIC=======================================  
  
void addRowD(regRow\* &arr,int &n) {  
 n++;  
 regRow\* arr1 = new regRow[n];  
 for (auto i = 0; i < n-1; i++) {  
 arr1[i] = arr[i];  
 }  
 arr1[n-1] = fillRow(arr1[n-1]);  
 arr = arr1;  
}  
void insertD(regRow\* &arr, int& n) {  
 int id;  
 cout << "Введите ID кассы для вставки: \n";  
 cin >> id;  
 regRow temp{};  
 temp = fillRow(temp);  
 n++;  
 regRow\* arr1 = new regRow[n];  
 arr1[n - 1] = {};  
  
 for (auto i = 0; i < n-1; i++) {  
 arr1[i] = arr[i];  
 }  
  
 for (auto i = 0; i < n; i++) {  
 if (arr1[i].id == id) {  
 for (auto j = i; j < n; j++)  
 swap(temp, arr1[j]);  
 break;  
 }  
 }  
 arr = arr1;  
}  
  
void deleteIDD(regRow\* &arr, int& n) {  
 cout << "Введите ID кассы для удаления: \n";  
 int id;  
 cin >> id;  
 for (auto i = 0; i < n; i++) {  
 if (arr[i].id == id) {  
 n--;  
 for (auto j = i; j < n; j++) {  
 swap(arr[j], arr[j + 1]);  
 }  
 i--;  
 }  
 }  
 regRow\* arr1 = new regRow[n];  
 for (auto i = 0; i < n; i++) {  
 arr1[i] = arr[i];  
 }  
 arr = arr1;  
}

**Задание 3**

Аналогично заданиям 1 и 2, но используется вектор.

Код задачи:

Также весь исходный код можно найти в репозитории [на GitHub](https://github.com/Joporezka/Siaod4_clion)

//==================================VECTOR=======================================  
  
  
int printTableV(vector<regRow> &arr, int n) {  
 cout << "ID | Barcode | Quantity | Price | Discount | Total\n";  
 for (int i = 0; i < n; i++) {  
 cout << arr[i].id << " " << arr[i].barcode << " " << arr[i].quantity << " " << arr[i].price << " " << arr[i].discount << " " << arr[i].total << endl;  
 }  
 return 0;  
}  
  
void printTableIDV(vector<regRow> &arr, int n) {  
 cout << "Введите ID кассы для вывода: \n";  
 int id;  
 cin >> id;  
  
 cout << "ID | Barcode | Quantity | Price | Discount | Total\n";  
 for (int i = 0; i < n; i++) {  
 if (arr[i].id == id)  
 cout << arr[i].id << " " << arr[i].barcode << " " << arr[i].quantity << " " << arr[i].price << " " << arr[i].discount << " " << arr[i].total << endl;  
 }  
}  
  
void addRowV(vector<regRow> &arr, int &n) {  
 n++;  
 regRow temp{};  
 arr.push\_back(fillRow(temp));  
}  
  
void insertV(vector<regRow> &arr, int &n) {  
 int id;  
 cout << "Введите ID кассы для вставки: \n";  
 cin >> id;  
 regRow temp{};  
 for (int i = 0; i < n; i++) {  
 if (arr[i].id == id) {  
 n++;  
 arr.insert(arr.begin() + i, fillRow(temp));  
 return;  
 }  
 }  
}  
  
void deleteIDV(vector<regRow> &arr, int &n) {  
 int id;  
 cout << "Введите ID кассы для удаления: \n";  
 cin >> id;  
 regRow temp{};  
 for (int i = 0; i < n; i++) {  
 if (arr[i].id == id) {  
 n--;  
 arr.erase(arr.begin() + i);  
 i--;  
 }  
 }  
}

Файл Header.h:

#pragma once  
#ifndef \_HEADER  
#define \_HEADER  
#include <iostream>  
#include <vector>  
using namespace std;  
  
struct regRow {  
 int id;  
 int barcode;  
 int quantity;  
 double price;  
 double discount;  
 double total;  
};  
  
//общие функции для статик динамик  
void printTable(regRow\* arr,int n);  
void printTableID(regRow\* arr, int n);  
regRow fillRow(regRow row);  
  
//статик  
void addRow(regRow\* arr, int& n);  
void insert(regRow\* arr, int& n);  
void deleteID(regRow\* arr, int& n);  
  
//функции динамик  
void addRowD(regRow\* &arr,int &n);  
void insertD(regRow\* &arr, int& n);  
void deleteIDD(regRow\* &arr, int& n);  
  
  
//функции вектор  
int printTableV(vector<regRow>& arr, int n);  
void printTableIDV(vector<regRow> &arr, int n);  
void addRowV(vector<regRow> &arr, int &n);  
void insertV(vector<regRow> &arr, int &n);  
void deleteIDV(vector<regRow> &arr, int &n);  
  
#endif // !\_HEADER

Файл main.cpp:

/\*  
Касса магазина. Структура записи операции по кассе: номер кассы, код товара, количество товара, цена товара, процентная скидка на товар, сумма за товар с учетом скидки.  
Операции  
!Заполнение записи по одной операции с клавиатуры.  
!Вставить записи по отдельной операции, проведенной кассой, в таблицу, располагая ее в начале списка других операций, проведенных этой кассой.  
!Вывести список операций, проведенных заданной кассой, с указанием стоимости покупки по каждой операции.  
Удалить записи по операциям, проведенным определенной кассой.  
  
\*/  
#include <iostream>  
#include "Header.h"  
#include <vector>  
  
  
using namespace std;  
const int N = 100;  
  
  
int main(){  
 int work = -1;  
 int choose = 100;  
 while (work != 0) {  
 cout << "Лабораторная работа №4 ИКБО-13-21 Смольников А.Б. Вариант 20" << endl << endl;  
 cout << "Касса магазина. Структура записи операции по кассе: номер кассы, код товара, количество товара, цена товара, процентная скидка на товар, сумма за товар с учетом скидки.\n"  
 "Операции:\n"  
 "Заполнение записи по одной операции с клавиатуры.\n"  
 "Вставить записи по отдельной операции, проведенной кассой, в таблицу, располагая ее в начале списка других операций, проведенных этой кассой. \n"  
 "Вывести список операций, проведенных заданной кассой, с указанием стоимости покупки по каждой операции.\n"  
 "Удалить записи по операциям, проведенным определенной кассой."<<endl;  
 cout << "Меню\n";  
 cout << "Задание 1 (Статический массив)" << endl;  
 cout << "Задание 2 (Динамический массив)" << endl;  
 cout << "Задание 3 (Вектор)" << endl;  
 cout << "0) Выход\n";  
 cout << "Ваш выбор: ";  
 cin >> work;  
 cout << "Лабораторная работа №3 ИКБО-13-21 Смольников А.Б. Вариант 20" << endl << endl;  
  
  
  
  
  
 switch (work) {  
 case 1: {  
 regRow reg[N];  
 int n = 0;  
 cout << "Касса магазина. Структура записи операции по кассе: номер кассы, код товара, количество товара, цена товара, процентная скидка на товар, сумма за товар с учетом скидки.\n";  
 cout << "1 Заполнение записи по одной операции с клавиатуры.\n"<<endl;  
 cout << "2 Вставить записи по отдельной операции, проведенной кассой, в таблицу, располагая ее в начале списка других операций, проведенных этой кассой. \n"<<endl;  
 cout << "3 Вывести список операций, проведенных заданной кассой, с указанием стоимости покупки по каждой операции.\n"<<endl;  
 cout<<"4 Удалить записи по операциям, проведенным определенной кассой."<<endl;  
 while(choose!=0){  
  
 cout<<"Введите номер операции\n";  
 cin>>choose;  
 switch (choose) {  
 case 1: {  
 addRow(reg,n);  
 break;  
 }  
 case 2: {  
 insert(reg,n);  
 break;  
 }  
 case 3: {  
 printTableID(reg, n);  
 break;  
 }  
 case 4: {  
 deleteID(reg, n);  
 break;  
 }  
 default:{  
 break;  
 }  
 }  
 }  
  
 break;  
 }  
 case 2: {  
 int n=0;  
 regRow\* regD = new regRow[n];  
  
 cout << "Задание 2 (динамический массив)" << endl;  
 cout << "Касса магазина. Структура записи операции по кассе: номер кассы, код товара, количество товара, цена товара, процентная скидка на товар, сумма за товар с учетом скидки.\n";  
 cout << "1 Заполнение записи по одной операции с клавиатуры.\n"<<endl;  
 cout << "2 Вставить записи по отдельной операции, проведенной кассой, в таблицу, располагая ее в начале списка других операций, проведенных этой кассой. \n"<<endl;  
 cout << "3 Вывести список операций, проведенных заданной кассой, с указанием стоимости покупки по каждой операции.\n"<<endl;  
 cout<<"4 Удалить записи по операциям, проведенным определенной кассой."<<endl;  
 while (choose>0){  
 cout<<"Введите номер операции\n";  
 cin>>choose;  
 switch (choose) {  
 case 1: {  
 addRowD(regD,n);  
 break;  
 }  
 case 2: {  
 insertD(regD,n);  
 break;  
 }  
 case 3: {  
 printTableID(regD, n);  
 break;  
 }  
 case 4: {  
 deleteIDD(regD, n);  
 break;  
 }  
 default:{  
 break;  
 }  
 }  
 }  
 break;  
 }  
 case 3: {  
 int n=0;  
 vector<regRow> regV;  
 cout << "Задание 3 (вектор)" << endl;  
 choose = -1;  
 cout << "Касса магазина. Структура записи операции по кассе: номер кассы, код товара, количество товара, цена товара, процентная скидка на товар, сумма за товар с учетом скидки.\n";  
 cout << "1 Заполнение записи по одной операции с клавиатуры.\n"<<endl;  
 cout << "2 Вставить записи по отдельной операции, проведенной кассой, в таблицу, располагая ее в начале списка других операций, проведенных этой кассой. \n"<<endl;  
 cout << "3 Вывести список операций, проведенных заданной кассой, с указанием стоимости покупки по каждой операции.\n"<<endl;  
 cout<<"4 Удалить записи по операциям, проведенным определенной кассой."<<endl;  
 while (choose!=0){  
 cout<<"Введите номер операции\n";  
 cin>>choose;  
 switch (choose) {  
 case 1: {  
 addRowV(regV,n);  
 break;  
 }  
 case 2: {  
 insertV(regV,n);  
 break;  
 }  
 case 3: {  
 printTableIDV(regV, n);  
 break;  
 }  
 case 4: {  
 deleteIDV(regV, n);  
 break;  
 }  
 default:{  
 break;  
 }  
 }  
 }  
  
 break;  
 }  
 default: {  
 break;  
 }  
 }  
 return 0;  
 }  
}

Тесты:

Для всех трех задач можно выделить оптимальный тест, который проверит работоспособность всех функций:

1. Добавить строку 1 201 10 10 10 (добавить в кассу 1 10 товаров (код 201) по 10 рублей со скидкой 10 процентов)
2. Добавить строку 2 201 2 2 50
3. Вывести на экран таблицу
4. Вставить запись кассы 1 (1 202 20 20 20)
5. Удалить записи кассы 2
6. Вывести на экран таблицу

Итог первого вывода:

ID | Barcode | Quantity | Price | Discount | Total

1 201 10 10 10 90

2 201 2 2 50 2

Итог второго вывода:

ID | Barcode | Quantity | Price | Discount | Total

1 202 20 20 20 320

1 201 10 10 10 90

Как можно увидеть, программа работает корректно, тест обрабатывается на всех трех реализациях (статик, динамик и вектор)

Также весь исходный код можно найти в репозитории [на GitHub](https://github.com/Joporezka/Siaod4_clion)