ЦЕЛЬ РАБОТЫ

Продемонстрировать работу с последовательными и ассоциативными контейнерами STL

ПРАКТИЧЕСКОЕ ЗАДАНИЕ

В каждом задании требуется обработать несколько (от двух до четырех) последовательностей из следующего набора:

A: сведения о потребителях, содержащие поля «Код потребителя», «Год рождения», «Улица проживания»;

B: сведения о товарах, содержащие поля «Артикул товара», «Категория», «Страна-производитель»;

C: скидки для потребителей в различных магазинах, содержащие поля «Код потребителя», «Название магазина», «Скидка (в процентах)»;

D: цены товаров в различных магазинах, содержащие поля «Артикул товара», «Название магазина», «Цена (в рублях)»;

E: сведения о покупках потребителей в различных магазинах, содержащие поля «Код потребителя», «Артикул товара», «Название магазина».

Порядок следования полей для элементов каждой последовательности определяется в формулировке задания.

В последовательности A все элементы имеют различные значения поля «Код потребителя».   
В последовательности B все элементы имеют различные значения поля «Артикул товара».   
В последовательности C все элементы имеют различные комбинации полей «Код потребителя» и «Название магазина».   
В последовательности D все элементы имеют различные комбинации полей «Артикул товара» и «Название магазина».   
Последовательность E может содержать одинаковые элементы (это соответствует ситуации, при которой один и тот же потребитель приобрел в одном и том же магазине несколько одинаковых товаров).

Все значения кодов потребителей и артикулов товаров, присутствующие в последовательностях C, D и E, обязательно содержатся в последовательностях A и B. Некоторые значения кодов потребителей и артикулов товаров, присутствующие в последовательностях A и B, могут отсутствовать в остальных последовательностях. Любая комбинация «магазин–товар», присутствующая в последовательности E, обязательно присутствует и в последовательности D. Комбинация «потребитель–магазин», присутствующая в последовательности E, может отсутствовать в последовательности C; это означает, что при покупке указанного товара в данном магазине потребитель не имел скидки (т. е. скидка была равна 0).

**Коды потребителей, годы рождения, скидки и цены задаются целыми числами; значения скидок лежат в диапазоне от 5 до 50**. Прочие поля являются строковыми и не содержат пробелов. Артикулы товаров имеют формат «AAddd-dddd», где на позициях, помеченных символом «A», располагается какая-либо заглавная латинская буква, а на позициях, помеченных символом «d», — какая-либо цифра.

Если потребитель приобрел товар, имеющий цену p, со скидкой d процентов, то размер скидки на данный товар должен вычисляться по формуле p · d / 100, где символ «/» обозначает операцию целочисленного деления (иными словами, при вычислении размера скидки копейки отбрасываются).

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Даны последовательности D и E, включающие следующие поля:

D: <Цена (в рублях)> <Название магазина> <Артикул товара>

E: <Артикул товара> <Название магазина> <Код потребителя>

Свойства последовательностей описаны в преамбуле к данной подгруппе заданий. Для каждой пары «магазин–товар», указанной в D, определить суммарную стоимость продаж этого товара в данном магазине (вначале выводится название магазина, затем артикул товара, затем суммарная стоимость его продаж). Если товар ни разу не был продан в некотором магазине, то для соответствующей пары «магазин–товар» в качестве суммарной стоимости выводится 0. Сведения о каждой паре «магазин–товар» выводить на новой строке и упорядочивать по названиям магазинов в алфавитном порядке, а в случае одинаковых названий — по артикулам товаров (также в алфавитном порядке).

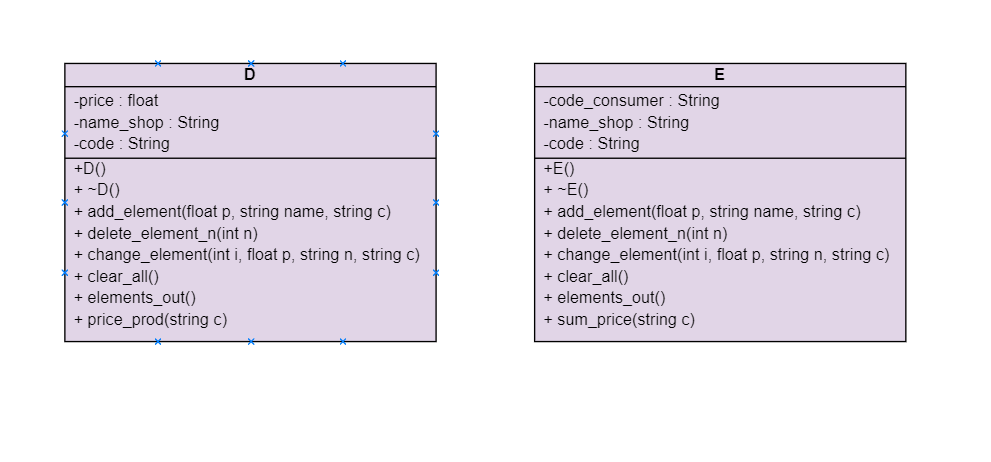
БЛОК-СХЕМА АЛГОРИТМА

Блок-схема алгоритма представлена на рисунке 1.

*Рисунок 1 Блок-схема программы*

ДИАГРАММА КЛАССОВ

Диаграмма классов представлена на рисунке 2.



*Рисунок 2 Диаграмма классов*

ЛИСТИНГ

#include <iostream>

#include <string>

#include <vector>

#include <algorithm>

#pragma warning(disable: 4018)

using namespace std;

class D { // различные комбинации name\_shop и code

private:

vector <float> price;

vector <string> name\_shop;

vector <string> code; //формат артикула AAddd-dddd

public:

D() {};

~D() {};

void add\_element(float p, string name, string c) {

try {

for (int i = 0; i < code.size(); i++) { //проверка на то, существует ли добавляемый элемент

if (code[i]==c && name\_shop[i]==name) {

string exe = "Такой товар в этом магазине уже существует, введите другой";

throw runtime\_error(exe);

}

}

price.push\_back(p);

name\_shop.push\_back(name);

code.push\_back(c);

}

catch(runtime\_error& r){

cout << r.what() << endl;

cout << "Введите цену товара: ";

cin >> p;

price.push\_back(p);

cout << "Введите название магазина: ";

cin >> name;

name\_shop.push\_back(name);

cout << "Артикул введите в формате «AAddd-dddd» ('A'- буква , 'd' - цифра): " ;

cin >> c;

code.push\_back(c);

}

}

void delete\_element\_n(int n) {

int i = n - 1;

price.erase(price.begin() + i);

name\_shop.erase(name\_shop.begin() + i);

code.erase(code.begin() + i);

}

void clear\_all() {

price.clear();

name\_shop.clear();

code.clear();

}

void change\_element(int i, float p, string n, string c) {

price[i-1] = p;

name\_shop[i-1] = n;

code[i-1] = c;

}

void elements\_out() {

if (code.size() != 0) {

for (int i = 0; i < code.size(); i++) {

for (int j = i + 1; j < code.size(); j++) {

string s1, s2;

s1 = name\_shop[i];

s2 = name\_shop[j];

if (s1[1] == s2[1] && s1[2] > s2[2]) {

swap(name\_shop[i], name\_shop[j]);

swap(price[i], price[j]);

swap(code[i], code[j]);

}

if (s1[1] > s2[1]) {

swap(name\_shop[i], name\_shop[j]);

swap(price[i], price[j]);

swap(code[i], code[j]);

}

}

}

}

for (int i = 0; i < code.size(); i++) {

cout << i+1 << " ";

cout << price[i] << " ";

cout << name\_shop[i] << " ";

cout << code[i] << endl;

}

}

float price\_prod(string c) {

for (int i = 0; i < code.size(); i++) {

if (code[i]==c) {

return price[i];

}

}

}

};

class E {

private:

vector <string> code;

vector <string> name\_shop;

vector <int> code\_consumer;

public:

E() {};

~E() {};

void add\_element(string c, string n, int cc) {

code.push\_back(c);

name\_shop.push\_back(n);

code\_consumer.push\_back(cc);

}

void delete\_element\_n(int n) {

int i = n - 1;

code.erase(code.begin() + i);

name\_shop.erase(name\_shop.begin() + i);

code\_consumer.erase(code\_consumer.begin() + i);

}

void clear\_all() {

code.clear();

name\_shop.clear();

code\_consumer.clear();

}

void change\_element(int i, string c, string n, int cc) {

code[i-1] = c;

name\_shop[i-1] = n;

code\_consumer[i-1] = cc;

}

void elements\_out() {

for (int i = 0; i < code.size(); i++) {

string s;

cout << i+1 << " ";

cout << code[i] << " ";

cout << name\_shop[i] << " ";

cout << code\_consumer[i] << endl;

}

}

int sum\_price(string c) {//количество проданного товара определенного типа

float sum=0;

for (int i = 0; i < code.size(); i++) {

if (code[i] == c) {

sum++;

}

}

return sum;

}

};

int main(void)

{

setlocale(LC\_ALL, "Russian");

float price, sum;

int trigger=1, action ,n;

string shop, code, code\_customer;

D D; //товары

E E; //покупатели

while (trigger == 1) {

cout << "1 - Добавить товар" << endl

<< "2 - Добавить покупателя" << endl

<< "3 - Удалить товар" << endl

<< "4 - Изменить товр" << endl

<< "5 - Вывести список всех товаров" << endl

<< "6 - Вывести список всех покупателей" << endl

<< "7 - Сумма продаж товара в магазине"<<endl

<< "8 - Рассчет суммы скидки на товар"<<endl

<< "9 - Завершить работу" << endl;

cin >> action;

if (action == 1) {

cout << "Введите цену товара: ";

cin >> price;

cout << "Введите название магазина: ";

cin >> shop;

cout << "Артикул введите в формате «AAddd-dddd»: ";

cin >> code;

D.add\_element(price, shop, code);

cout << "Товар успешно добавлен!" << endl;

}

if (action == 2) {

cout << "Артикул введите в формате «AAddd-dddd»: ";

cin >> code;

cout << "Введите название магазина: ";

cin >> shop;

cout << "Введите код покупателя: ";

cin >> code\_customer;

D.add\_element(price, shop, code);

cout << "Покупатель успешно добавлен!" << endl;

}

if (action == 3) {

cout <<"Введите номер элемента для удаление: ";

cin >> n;

D.delete\_element\_n(n);

cout <<"Элемент успешно удален!" <<endl;

}

if (action == 4) {

cout <<"Введите номер элемента к изменению: ";

cin >> n;

cout << "Введите цену товара: ";

cin >> price;

cout << "Введите название магазина: ";

cin >> shop;

cout << "Артикул введите в формате «AAddd-dddd»: ";

cin >> code;

D.change\_element(n, price, shop, code);

}

if (action == 5) {

D.elements\_out();

}

if (action == 6) {

E.elements\_out();

}

if (action == 7) {

int num;//количесто товара

float mul; //итоговая стоимость

cout << "Введите код товара, для которого требуется рассчитать общую стоимость продаж: ";

cin >> code;

num=E.sum\_price(code);

mul = D.price\_prod(code) \* num;

cout <<"Стоимость продаж данного товара: " << mul << endl;

}

if (action == 8) {

int disc;

cout << "Введите код товара: ";

cin >> code;

cout << "Введите размер скидки на товар: " ;

cin >> disc;

if (disc > 50 || disc < 5) {

cout << "Неверно введена скидка, введите заново: ";

cin >> disc;

}

cout << "Результирующая скидка на товар: ";

cout.precision(0);

cout << (D.price\_prod(code)\*disc)/100 << endl;

}

if (action == 9) {

trigger = 0;

}

}

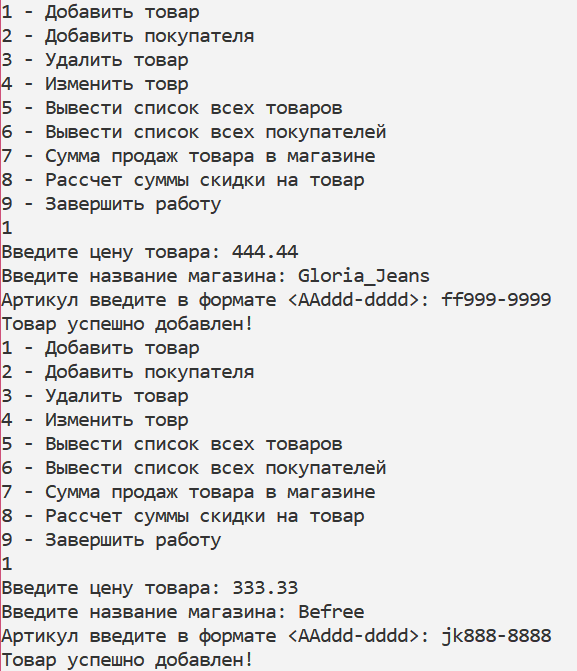
system("pause");

return 0;

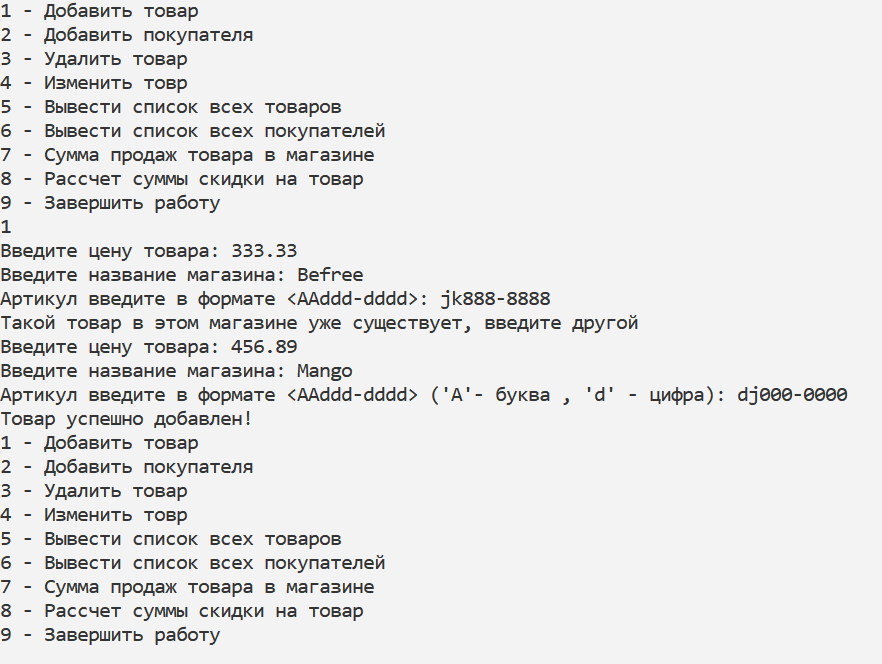
}

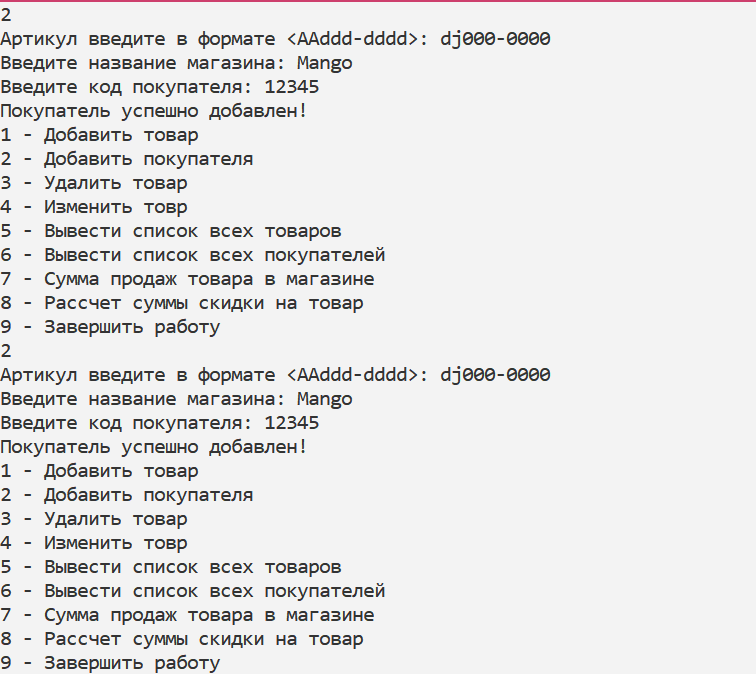
ПРОТОКОЛ ВЫПОЛНЕНИЯ ПРОГРАММЫ

Протокол выполнения программы представлен на рис. 3.

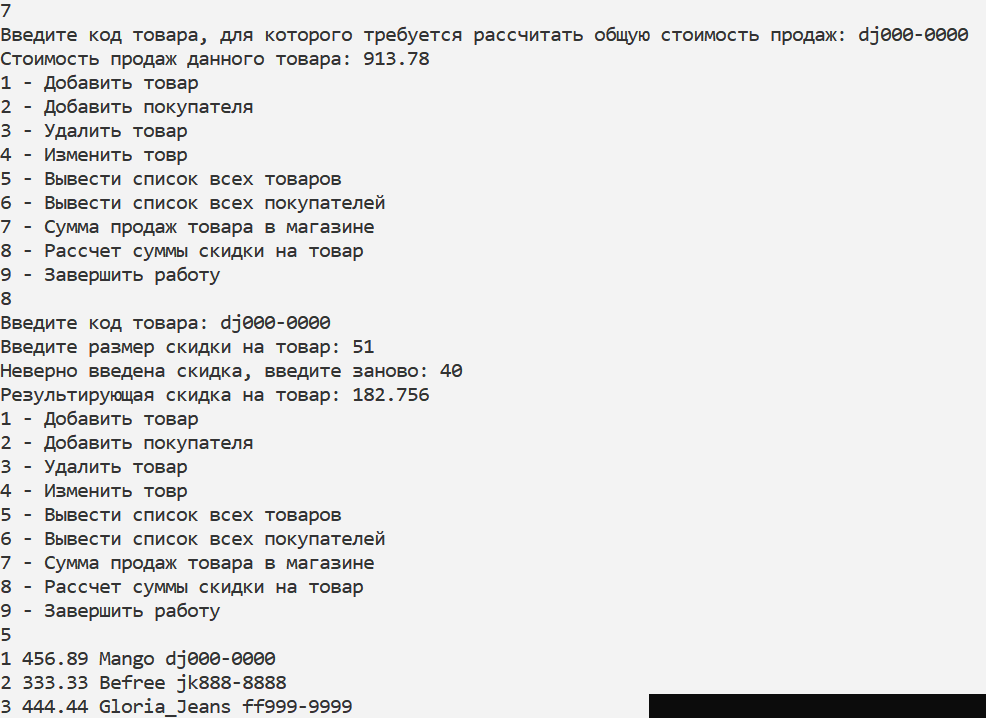


*Рисунок 3 Протокол программы*

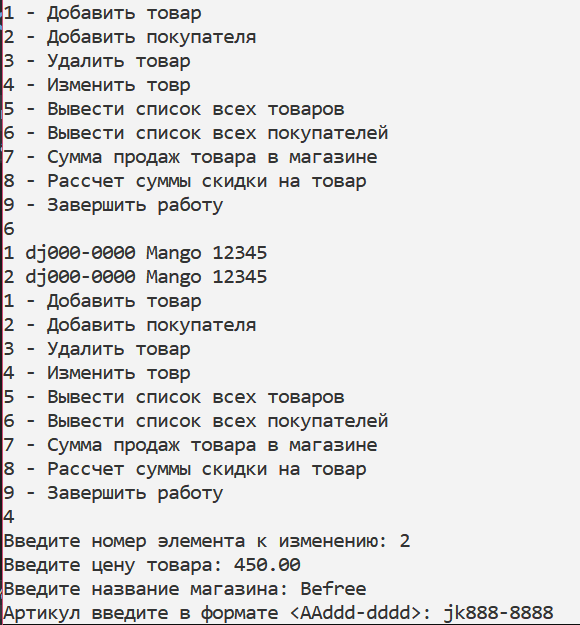
 *Рисунок 3 Протокол программы*



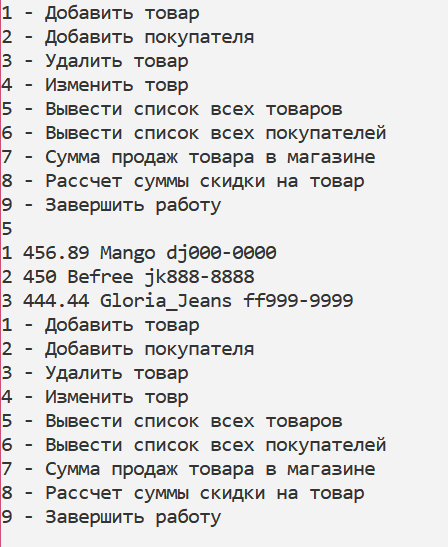
*Рисунок 3 Протокол программы*



*Рисунок 3 Протокол программы*



*Рисунок 3 Протокол программы*



*Рисунок 3 Протокол программы*