Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный технологический университет»

**Лабораторная работа**

На тему:

" Создание меню программы. Разбиение программы на файлы. STL контейнеры. Array “

**Выполнил:**

Волковец Сергей Николаевич

(подпись)

**Научный руководитель:**

Белодед Николай Иванович

(подпись)

**Теоретические сведения:**

Создание меню в базах данных необходимо для обеспечения удобного и интуитивно понятного интерфейса для пользователя, позволяет пользователю выбирать необходимые ему функции и операции из списка доступных вариантов. Это может существенно упростить работу с базой данных и снизить вероятность ошибок при вводе команд.  
 Меню может быть использовано для ограничения доступа к определенным функциям базы данных только для определенных пользователей. Например, администратор базы данных может иметь доступ ко всем функциям, в то время как обычный пользователь может иметь доступ только к просмотру и редактированию своих записей.  
 Управление целостностью данных. Создание меню может помочь в обеспечении целостности данных, поскольку пользователи могут выбирать только определенные операции и функции, которые были предварительно проверены и одобрены разработчиками базы данных. Это может предотвратить случайное удаление или изменение данных.

Синтаксис STL основан на использовании таких синтаксических конструкций языка C++ как шаблоны (templates) классов и шаблоны функций.   
 По своей сути STL — это сложный набор шаблонных классов и функций, реализующих многие популярные и часто используемые структуры данных и алгоритмы.  
 Центральным понятием STL, вокруг которого крутится всё остальное, это контейнер . **Контейнер** — это набор некоторого количества обязательно однотипных элементов, упакованных в контейнер определенным образом. Простейшим прототипом контейнера в классическом языке C++ является массив. Тот способ, которым элементы упаковываются в контейнер и определяет тип контейнера и особенности работы с элементами в таком контейнере. STL вводит целый ряд разнообразных типов контейнеров, основные из них:

* последовательные контейнеры — вектор (vector), двусвязный список (list), дэк (deque);
* ассоциативные контейнеры — множества (set), хэш-таблицы (map);

|  |  |  |
| --- | --- | --- |
| Контейнер | Описание | Требуемый заголовок |
| array | Массив | <array> |
| vector | Динамический массив | <vector> |
| deque | Двунаправленная очередь | <deque> |
| forward\_list | Односвязный список | <forward\_list> |
| list | Двусвязный список | <list> |
| set | Множество уникальных элементов | <set> |
| map | Хранит пары “ключ\значение”,где каждый ключ может быть ассоциирован с двумя и более значениями | <map> |

**Массив** — структура данных, хранящая набор значений (элементов массива), идентифицируемых по индексу или набору индексов, принимающих целые (или приводимые к целым) значения из некоторого заданного непрерывного диапазона.  
 **std::array** — это фиксированный массив, который не распадается в указатель при передаче в функцию.

#include <array>

int main()  
{  
std::array<int,6> numbers; // объявляем массив значений int длиной 6  
}

Основные функции array  
 В контейнер array нельзя добавлять новые элементы, так же как и удалять уже имеющиеся. Основные функции типа array, которые мы можем использоватьб:

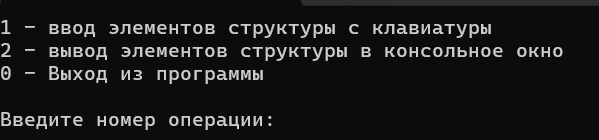
* **size()**: возвращает размер контейнера
* **at(index)**: возвращает элемент по индексу index
* **front()**: возвращает первый элемент
* **back()**: возвращает последний элемент
* **fill(n)**: присваивает всем элементам контейнера значение n

Для доступа к элементам контейнера array можно применять тот же синтаксис, что при работе с массивами - в квадратных скобках указывать индекс элемента, к которому идет обращение:

|  |
| --- |
| #include <array> #include <iostream>  using namespace std;  int main() {  array<int, 6> numbers{ 2, 3, 4, 5, 6, 7 };  int n = numbers[3];  cout << "n = " << n << std::endl; // получаем значение элемента n = 5  numbers[3] = 12; // меняем значение элемента  cout << "numbers[3] = " << numbers[3] << endl;//numbers[3] = 12 } |

Задание 1

Реализовать полноценное меню, показывающее весь функционал, описанный в предыдущей лабораторной работе.



Код:

//2.База клиентов. Содержится информация о фамилии, имени, его контактный телефон, адрес.

// Имеется информация потраченных за всё время средств, при достижении

// определённой суммы клиенту предоставляется скидка 10% на дальнейшие заказы.

#include <iostream>

#include <string>

#include <time.h>

#include <windows.h>

using namespace std;

struct person

{

string surname;

string name;

string phone;

string address;

int out\_money;

bool diskountt = false;

};

struct person clients[100];

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

srand(time(NULL));

setlocale(LC\_ALL, "Russian");

const int diskount = 200;

int n = 0;

int choice = 1;

while (choice != 0) {

cout << "1 - ввод элементов структуры с клавиатуры\n";

cout << "2 - вывод элементов структуры в консольное окно\n";

cout << "0 - Выход из программы\n\n";

cout << "Введите номер операции: ";

cin >> choice;

if (choice == 1) {

cout << "Введите данные: Фамилию, Имя,Контактный телефон, адрес" << endl;

cout << "Введите вашу фамилию ";

cin >> clients[n].surname;

cout << "Введите ваше имя ";

cin >> clients[n].name;

cout << "Введите ваш контактный телефон ";

cin >> clients[n].phone;

cout << "Введите ваш адрес ";

cin >> clients[n].address;

clients[n].out\_money = rand() % 1000;

if (clients[n].out\_money >= diskount) {

clients[n].diskountt = true;

cout << "Поздравляем вы получаете скидку в 10% " << endl;

}

n++;

}

if (choice == 2) {

cout << "База данных" << endl;

for (int i = 0; i < n; i++) {

cout << clients[i].surname << " ";

cout << clients[i].name << " ";

cout << clients[i].phone << " ";

cout << clients[i].address << " ";

if (clients[i].diskountt == true) {

cout << "Имеется скидка в десять процентов";

}

cout << endl;

}

}

}

}

**Задание 2.**

Разбить программу на файлы. Предлагаем разбить программу на следующие файлы:

1. main.cpp - содержит функцию main, в которой реализован switch case с меню программы
2. func.cpp - содержит определение функций

func.h - содержит объявление функций

2)main.cpp:

#include <iostream>

#include <string>

#include <time.h>

#include <windows.h>

#include "func.h"

using namespace std;

struct person clients[100];

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

srand(time(NULL));

setlocale(LC\_ALL, "Russian");

const int diskount = 200;

int n = 0;

int choice = 1;

cout << "\033[31mКрасный текст\n";

while (choice != 0) {

window();

cin >> choice;

if (choice == 1) {

choice1(clients, n, diskount);

}

if (choice == 2) {

choice2(clients, n);

}

}

}

2)func.cpp

#include <iostream>

#include "func.h"

using namespace std;

void window() {

cout << "1 - ввод элементов структуры с клавиатуры\n";

cout << "2 - вывод элементов структуры в консольное окно\n";

cout << "0 - Выход из программы\n\n";

cout << "Введите номер операции: ";

}

void choice1(struct person clients[100], int& n, int diskount) {

cout << "Введите данные: Фамилию, Имя, Контактный телефон, адрес" << endl;

cout << "Введите вашу фамилию ";

cin >> clients[n].surname;

cout << "Введите ваше имя ";

cin >> clients[n].name;

cout << "Введите ваш контактный телефон ";

cin >> clients[n].phone;

cout << "Введите ваш адрес ";

cin >> clients[n].address;

clients[n].out\_money = rand() % 1000;

if (clients[n].out\_money >= diskount) {

clients[n].diskountt = true;

cout << "Поздравляем вы получаете скидку в 10% " << endl;

}

n++;

}

void choice2(struct person clients[100], int n) {

cout << "База данных" << endl;

for (int i = 0; i < n; i++) {

cout << clients[i].surname << " ";

cout << clients[i].name << " ";

cout << clients[i].phone << " ";

cout << clients[i].address << " ";

if (clients[i].diskountt == true) {

cout << "Имеется скидка в десять процентов";

}

cout << endl;

}

}

3)func.h:

#ifndef FUNC\_H

#define FUNC\_H

#include <string>

using namespace std;

struct person {

string surname;

string name;

string phone;

string address;

int out\_money;

bool diskountt;

};

void window();

void choice1(struct person clients[100], int& n, int diskount);

void choice2(struct person clients[100], int n);

#endif

3) **Задание 3.**

Написать программу, в которой нужно найти сумму минимального и максимального элементов массива, с выводом на экран.

#include <iostream>

#include <array>

using namespace std;

int main()

{

setlocale(LC\_ALL, "ru");

int n;

cout << "Введите количество чисел n" << endl;

cin >> n;

array <int, 100> arr;

int max = INT\_MIN;

int min = INT\_MAX;

cout << "Введите массив ";

for (int i = 0; i < n; i++) {

cin >> arr[i];

if (arr[i] > max) {

max = arr[i];

}

if (arr[i] < min) {

min = arr[i];

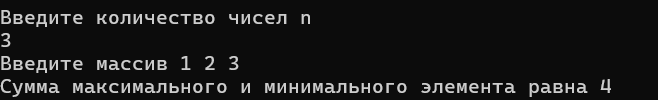
}

}

cout << "Сумма максимального и минимального элемента равна "<< min + max;

}

Пример работы:



**Задание 4.**

Написать программу, в которой нужно создать два массива , объединить их и вывести на экран сумму элементов.

#include <iostream>

#include <array>

using namespace std;

int main() {

setlocale(LC\_ALL, "ru");

int n, nn;

cout << "Введите количество чисел n 1 массива" << endl;

cin >> n;

cout << "Введите количество чисел n 2 массива" << endl;

cin >> nn;

array<int, 100> arr;

array<int, 100> arr2;

cout << "Введите массив 1 ";

for (int i = 0; i < n; i++) {

cin >> arr[i];

}

int sum = 0;

cout << "Введите массив 2 ";

for (int i = 0; i < nn + n; i++) {

if (i >= nn) {

arr2[i] = arr[i - nn];

}

else {

int p;

cin >> p;

arr2[i] = p;

}

sum += arr2[i];

}

cout << "Сумма всех членов объединённого массива равна " << sum << endl;

}

Пример работы кода: #include <iostream>

#include <array>

using namespace std;

int main() {

setlocale(LC\_ALL, "ru");

int n, nn;

cout << "Введите количество чисел n 1 массива" << endl;

cin >> n;

cout << "Введите количество чисел n 2 массива" << endl;

cin >> nn;

array<int, 100> arr;

array<int, 100> arr2;

cout << "Введите массив 1 ";

for (int i = 0; i < n; i++) {

cin >> arr[i];

}

int sum = 0;

cout << "Введите массив 2 ";

for (int i = 0; i < nn + n; i++) {

if (i >= nn) {

arr2[i] = arr[i - nn];

}

else {

int p;

cin >> p;

arr2[i] = p;

}

sum += arr2[i];

}

cout << "Сумма всех членов объединённого массива равна " << sum << endl;

}

**Дополнительное задание 1.**

Объедините два массива целых чисел в один и выведите его элементы:

//1.Объедините два массива целых чисел в один и выведите его элементы

#include <iostream>

#include <array>

using namespace std;

int main() {

setlocale(LC\_ALL, "ru");

array<int, 5> array1 = { 1, 2, 3, 4, 5 };

array<int, 5> array2 = { 6, 7, 8, 9, 10 };

array<int, 10> combinedArray;

for (int i = 0; i < 5; ++i) {

combinedArray[i] = array1[i];

combinedArray[i + 5] = array2[i];

}

cout << "Объединенный массив:" << endl;

for (int i = 0; i < 10; ++i) {

cout << combinedArray[i] << " ";

}

cout << endl;

return 0;

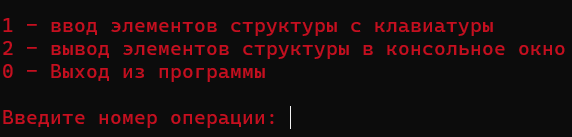
}

Пример работы кода:



**Дополнительное задание 2.**

Улучшить визуальное представление меню, используя изменение цвета консоли. (Приложение 1).



3) **Дополнительное задание 3.**

Напишите программу, которая запрашивает у пользователя 10 целых чисел и сохраняет их в контейнере array в STL C++. Затем программа должна вычислить среднее арифметическое этих чисел и вывести на экран все числа, которые больше среднего арифметического.

#include <iostream>

#include <array>

using namespace std;

int main() {

setlocale(LC\_ALL, "ru");

array<int, 10> numbers;

int sum = 0;

cout << "Введите 10 целых чисел:" << endl;

for (int i = 0; i < 10; ++i) {

cin >> numbers[i];

sum += numbers[i];

}

double average = sum / 10.0;

cout << "Числа, которые больше среднего арифметического (" << average << "):" << endl;

for (int i = 0; i < 10; ++i) {

if (numbers[i] > average) {

cout << numbers[i] << " ";

}

}

cout << endl;

return 0;

}

Пример работы кода:

