ОДЕССКИЙ НАЦИОНАЛЬНЫЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ИНСТИТУТ КОМПЬЮТЕРНЫХ СИСТЕМ

Кафедра информационных систем и технологий

**Лабораторная работа №11**

по дисциплине: «Алгоритмизация и программирование»

Вариант № 1

**Выполнил:**

АД-192

Березовский В. А.

**Проверили:**

Одесса 2020

**Тема:** векторы в С++.

**Цель:** ознакомиться с векторами, какие методы существуют для работы с векторами, и понять, как работает итератор и новые функции.

**Ход работы**

**Задание:**

Сформировать вектор, добавить в него элементы в начало, в конец, в середину, изменить элемент вектора, удалить элемент, на который указывает итератор.

Вывести на консоль элементы вектора после каждого изменения.

**Листинг:**

#include <iostream>

#include <vector> // подключаем данную библиотеку, для работы с векторами

#include <string> // подключаем данную библиотеку, для работы со строками

using namespace std;

void add\_end() { // функция для добавления элемента в конец вектора

vector<int> sample\_1{ 1,2,3,4,5 }; // сформировали вектор и присвоили значения

int value\_1;

cout << "\nEnter the number you want to put at the end of the vector: ";

cin >> value\_1;

sample\_1.push\_back(value\_1); // добавляем элемент в конец вектора

for (int i = 0; i < sample\_1.size(); i++) { // выводим значения вектора на экран

cout << sample\_1[i] << " ";

}

cout << endl << endl;

}

void add\_start() { // функция для добавления элемента в начало вектора

vector<int> sample\_2{ 1,2,3,4,5 }; // сформировали вектор и присвоили значения

int value\_2;

cout << "\nEnter the number you want to put at the start of the vector: ";

cin >> value\_2;

sample\_2.insert(sample\_2.begin(), value\_2); // добавляем элемент к началу вектора

for (int i = 0; i < sample\_2.size(); i++) { // выводим значения вектора на экран

cout << sample\_2[i] << " ";

}

cout << endl << endl;

}

void add\_middle() { // функция для добавления элемента в середине вектора

vector<int> sample\_3{ 1,2,3,4,5 }; // сформировали вектор и присвоили значения

auto iter = sample\_3.cbegin(); // создаем итератор

int value\_3;

cout << "\nEnter the number you want to put at the middle of the vector: ";

cin >> value\_3;

sample\_3.insert(iter + 3, value\_3); // добавляем элемент в середину вектора через итератор

for (int i = 0; i < sample\_3.size(); i++) { // выводим значения вектора на экран

cout << sample\_3[i] << " ";

}

cout << endl << endl;

}

void change() { // функция для изменения элементов вектора

vector<int> sample\_4{ 1,2,3,4,5 }; // сформировали вектор и присвоили значения

int value\_4;

cout << "\nEnter the value which you want to change the vectors: ";

cin >> value\_4;

sample\_4.assign(5, value\_4); // изменения элементов в векторе через итератор

for (int i = 0; i < sample\_4.size(); i++) { // выводим значения вектора на экран

cout << sample\_4[i] << " ";

}

cout << endl << endl;

}

void delet() { // функиция для удаления элемента вектора на который указывает итератор

std::vector<int> sample\_5{ 1,2,3,4,5 }; // сформировали вектор и присвоили значения

auto iter\_1 = sample\_5.cbegin(); // создаем итератор

int value\_5;

cout << "\nEnter the number which you want to delete the vectors: ";

cin >> value\_5;

sample\_5.erase(iter\_1 + (value\_5 - 1)); // удоляем элемент в векторе через итератор

for (int i = 0; i < sample\_5.size(); i++) { // выводим значения вектора на экран

cout << sample\_5[i] << " ";

}

cout << endl << endl;

}

void vectors() { // функция для вывода значений вектора на экран

vector<int> sample{ 1,2,3,4,5 }; // формируем вектор и присваем элементы

cout << "Vector Elements: " << endl;

for (int i = 0; i < sample.size(); i++) { // выводим значения вектора на экран

cout << sample[i] << " ";

}

cout << endl << endl;

}

int main() {

int menu;

vectors(); // вызываем функция

cout << "Menu:\n1) Add item at the end;\n2) Add item at the beginning;\n3) Add item at the middle;\n4) Edit item;\n5) Delete item;\n";

tryAgain:

cout << "\nEnter number: ";

cin >> menu;

switch (menu){ // меню для выбора нужной функции

case 1: {

add\_end(); // вызов функции

break;

}

case 2: {

add\_start(); // вызов функции

break;

}

case 3: {

add\_middle(); // вызов функции

break;

}

case 4: {

change(); // вызов функции

break;

}

case 5: {

delet(); // вызов функции

break;

}

default:

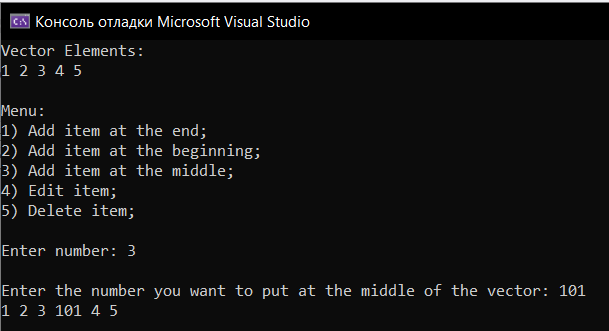
cout << "\nError, please choose the right way!" << endl;

goto tryAgain; // если не правильно выбрали то возвращаемся назад к меню

}

}

**Тестирование программы:**

****

**Вывод:** я ознакомился с векторами, какие методы существуют для работы с векторами, и понял, как работает итератор и новые функции.