МИНИСТЕРСТВО ТРАНСПОРТА И КОММУНИКАЦИЙ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования

«Белорусский государственный университет транспорта»

Кафедра «Информационно-управляющие системы и технологии»

Отчет  
по лабораторным работам

по дисциплине «Объектно-ориентированное программирование»

Выполнил Проверил

студент группы ГИ-21 ст. преп. каф. ИУСиТ

Зайцев И. С. Лыч Ю. П.

Гомель, 2021

Содержание

[Лабораторная работа №10 Использование последвательных контейнеров из стандартной библиотеки шаблонов STL 3](#_Toc65343672)

[1.1 Задания 3](#_Toc65343673)

[1.2 Контрольные вопросы 5](#_Toc65343674)

# Лабораторная работа №10 использование последвательных контейнеров из стандартной библиотеки шаблонов STL

### **Цель работы**

Изучить теоретические принципы и получить практические навыки разработки программ в среде Visual Studio с использованием типа vector, list, deque из стандартной библиотеки шаблонов STL.

## Задания

### **Задание 1**

### Условие

Дан вектор с положительными и отрицательными целыми числами. Записать в другой вектор сначала отрицательные элементы, а затем положительные.

### Листинг программы

#include <iostream>

#include <vector>

using namespace std;

int main()

{

int i, j, N=15;

vector<int> mass(N);

vector<int> mass2(N);

cout << "Vector with random values:" << endl;

for(i=0; i<N; i++)

{

mass[i]=rand()%21-10;

cout << mass[i] <<" ";

}

for(i=0, j=0; i<N; i++)

{

if(mass[i]<0)

{

mass2[j]=mass[i];

j++;

}

}

for(i=0, j; i<N; i++)

{

if(mass[i]>=0)

{

mass2[j]=mass[i];

j++;

}

}

cout << "\nFinal vector variant:" << endl;

for(i=0; i<N; i++)

{

cout << mass2[i] << " ";

}

cout << endl;

return 0;

}

### Результат выполнения программы

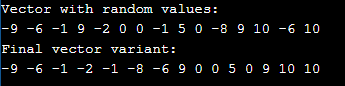


Рисунок 1.1 – Результат выполнения программы задания 1

### **Задание 2**

### Условие

### Листинг программы

#include <iostream>

#include <list>

#include <iterator>

#include <ctime>

using namespace std;

int main()

{

list<int> L;

int N = 10, i, key;

srand(time(NULL));

cout << "List of " << N <<" items:" << endl;

for(i=1; i<=N; i++)

{

key = rand() % 21-10;

L.push\_back(key);

}

for(int data : L) cout << data << " " ;

cout << endl;

for(auto i = L.begin(); i != L.end();)

{

i = L.erase(++i);

}

cout << "List with no even elements:" << endl;

for(int data : L) cout << data << " " ;

cout << endl;

return 0;

}

### Результат выполнения программы

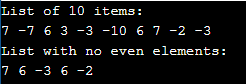


Рисунок 1.2 – Результат выполнения программы задания 2

## Контрольные вопросы

1. Из чего состоит ядро библиотеки шаблонов?

Ядро стандартной библиотеки шаблонов составляют три основных элемента: контейнеры, алгоритмы и итераторы. Они работают совместно, предоставляя готовые решения различных проблем программирования.

2. Дайте определения контейнеров, алгоритмов, итераторов, функциональных объектов.

Контейнеры – это объекты, содержащие другие объекты. Существует несколько типов контейнеров:

− последовательные. Представляют собой линейный список. Примеры: вектора, очереди, списки;

− ассоциативные. Позволяют эффективно находить нужные значения на основе заданных ключей. Примеры: отображения, множества. Каждый контейнерный класс определяет набор функций, которые можно применять к данному контейнеру. Например, контейнер списка включает функции, предназначенные для выполнения вставки, удаления и объединения элементов.

Алгоритмы действуют в контейнерах. Они позволяют выполнять инициализацию, сортировку, поиск и преобразовывать содержимое контейнеров. Многие алгоритмы работают с группой (или диапазоном) элементов внутри контейнера.

Итераторы – это объекты, которые действуют подобно указателям. Они позволяют получать доступ к содержимому контейнера так, как указатель позволяет получить доступ к элементу массива. Итераторы обрабатываются аналогично указателям. Их можно инкрементировать и декрементировать. К ним можно применять операцию разыменования. Итераторы объявляются с помощью типа iterator, определяемого различными контейнерами.

Существует 5 типов итераторов:

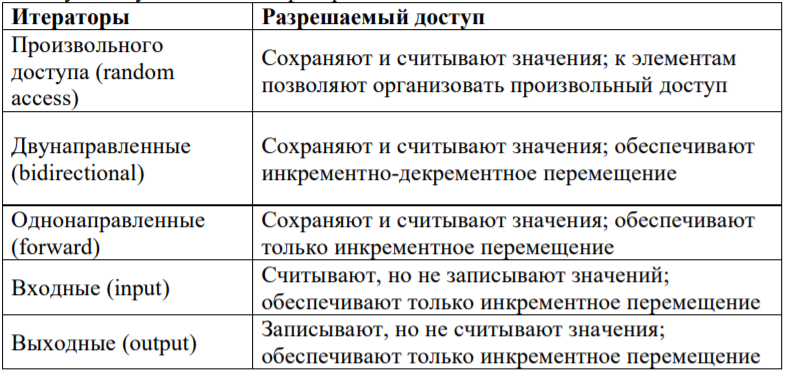


Рисунок 1.3 – Типы итераторов

Библиотека STL поддерживает реверсивные итераторы, которые являются либо двунаправленными, либо итераторами произвольного доступа, т.е. позволяют перемещаться по последовательности в обратном направлении. Следовательно, если реверсивный итератор указывает на конец последовательности, то после инкрементирования он будет указывать на элемент, расположенный перед концом последовательности.

Функциональные объекты – это объекты, действующие как функции. Они могут быть объектами класса или указателями на функции, включающими в себя имя функции, поскольку именно оно работает как указатель. Библиотека STL позволяет построить множество контейнеров, включая массивы, очереди и списки, и выполнить нал ними различные операции: заполнение случайным образом, поиск, сортировку и т. д.

3. Что представляет собой контейнерный класс?

Контейнерный класс – это абстрактный тип данных (АТД). Он обеспечивает абстрактные методы хранения данных. Например, стек (дек) – это АТД с функциями для запоминания и обработки данных по принципу «Последним вошел первым вышел». Он представляет собой просто метод программирования, который не зависит от конкретных деталей реализации способа хранения данных в нем.

4. Фундаментальная структура данных (ФСД) – это…

Фундаментальная структура данных (ФСД) – это любая структура, которая может использоваться для реализации АТД. Например, класс стека (АТД) можно реализовать с помощью связного списка (ФСД), или массива (ФСД), или другого фундаментального типа. В каждом контейнерном классе сочетаются один АТД и одна ФСД.

5. Перечислите методы, общие для всех контейнеров.

Методы, общие для всех контейнеров, приведены в таблице.

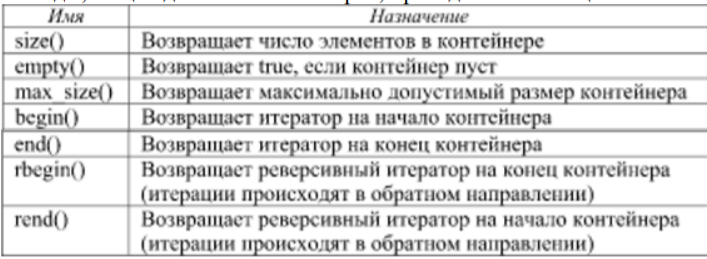


Рисунок 1.3 – Методы контейнеров

### **Выводы по работе**

В данной практической работе мы изучили теоретические принципы и получить практические навыки разработки программ в среде Visual Studio с использованием типа vector, list, deque из стандартной библиотеки шаблонов STL с использованием типа string из стандартной библиотеки шаблонов STL.