|  |  |
| --- | --- |
| *voenmeh* | МИНОБРНАУКИ РОССИИ  федеральное государственное бюджетное образовательное учреждение  высшего образования  **«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова»**  **(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова»)** |
| БГТУ.СМК-Ф-4.2-К5-02 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Факультет |  | О |  | Естественнонаучный |
|  |  | шифр |  | наименование |
| Кафедра |  | О7 |  | Информационные системы и программная инженерия |
|  |  | шифр |  | наименование |
| Дисциплина |  | Информационные технологии и программирование | | |

|  |
| --- |
| Практическая работа №3 |
| Шаблоны |
| Вариант №25 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил студент группы | | | |  | И924Б |
| Фокин В. К. | | | | | |
| Фамилия И.О. | | | | | |
| **Преподаватель:** | | | | | |
| Гладевич А. А. | |  |  | | |
| Фамилия И.О. Подпись | | | | | |
| Оценка |  | | | |  |
| «\_\_\_\_\_» |  | | | | 2023г. |

САНКТ-ПЕТЕРБУРГ

2023 г.

**Цель работы**

Научиться создавать шаблоны функций и шаблоны классов для работы с любыми типами данных без переписывания кода программы.

**Постановка задачи**

Написать шаблон функции, выполняющей указанные в вариативной части задания действия. Написать программу тестирования шаблонных функций, созданных на основе этого шаблона, с аргументами указанных типов. Разработать шаблон класса, описывающий указанный в вариативной части задания абстрактный тип данных, и написать программу тестирования объектов двух шаблонных классов. Выбор тестируемого метода должен осуществляться с помощью меню.

**Уровень сложности** – Средний.

Заданный АТД реализовать с помощью указанной структуры хранения.

**Вариант 25**

Типы аргументов: unsigned int, int.

1. Сортировка элементов одномерного массива по неубыванию.
2. АТД Дек. Структура хранения – связанный список.

**Текст программы:**

Файл deque.h

#include <iostream>

using namespace std;

//узел двусвязного списка

template <typename T>

struct Node {

T data;

Node<T>\* prev;

Node<T>\* next;

};

//класс дек

template <typename T>

class Deque {

Node<T>\* front;

Node<T>\* rear;

int size;

public:

//конструктор

Deque(int max\_size) {

front = rear = NULL;

size = 0;

}

//добавление элемента в начало

void pushFront(T data) {

Node<T>\* newNode = new Node<T>;

newNode->data = data;

newNode->prev = NULL;

newNode->next = front;

if (isEmpty()) {

rear = newNode;

}

else {

front->prev = newNode;

}

front = newNode;

size++;

}

//удаления элемента из начала

void popFront() {

if (isEmpty()) {

cout << "Deque is empty" << endl;

return;

}

Node<T>\* temp = front;

if (front == rear) {

front = rear = NULL;

}

else {

front = front->next;

front->prev = NULL;

}

delete temp;

size--;

}

//добавления элемента в конец

void pushRear(T data) {

Node<T>\* newNode = new Node<T>;

newNode->data = data;

newNode->prev = rear;

newNode->next = NULL;

if (isEmpty()) {

front = newNode;

}

else {

rear->next = newNode;

}

rear = newNode;

size++;

}

//удаления элемента из конца

void popRear() {

if (isEmpty()) {

cout << "Deque is empty" << endl;

return;

}

Node<T>\* temp = rear;

if (front == rear) {

front = rear = NULL;

}

else {

rear = rear->prev;

rear->next = NULL;

}

delete temp;

size--;

}

//получения первого элемента

T getFront() {

if (isEmpty()) {

cout << "Deque is empty" << endl;

return T();

}

return front->data;

}

//получения последнего элемента

T getRear() {

if (isEmpty()) {

cout << "Deque is empty" << endl;

return T();

}

return rear->data;

}

//проверки, пустой ли дек

bool isEmpty() {

return front == NULL;

}

//получения размера дека

int getSize() {

return size;

}

//вывод всех элементов в деке

void print() {

Node<T>\* current = front;

while (current != NULL) {

cout << current->data << " ";

current = current->next;

}

cout << endl;

}

};

Файл template.h

#include <iostream>

using namespace std;

//шаблон сортировки по неубыванию

template <typename T>

void sortArray(T arr[], int size) {

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

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

if (arr[i] > arr[j]) {

T temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

}

// функция для вывода массива на экран

template <typename T>

void printArray(T arr[], int size) {

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

cout << arr[i] << " ";

}

cout << endl;

}

Файл *program.cpp*

#include <iostream>

#include "deque.h"

#include "template.h"

using namespace std;

int main() {

const int size = 6;

int s = 0, x = 0;

int arr1[] = { 5, 2, -3, 6, -10, 8 };

unsigned int arr2[] = { 7, 14, 2, 24, 12, 5 };

Deque<int> deque1(size);

Deque<unsigned> deque2(size);

int front1, rear1;

unsigned front2, rear2;

char menu, nemu, unem;

do {

cout << "---------------------------------------------------------------" << endl;

cout << "1. Sorting |" << endl;

cout << "2. Test deque (nodes) |" << endl;

cout << "0. Exit |" << endl;

cout << "---------------------------------------------------------------" << endl;

cin >> menu;

switch (menu) {

case '1':

system("cls");

do {

cout << "---------------------------------------------------------------" << endl;

cout << "1. Sort elements in descending order in an int array |" << endl;

cout << "2. Sort elements in descending order in an unsigned int array |" << endl;

cout << "0. Back |" << endl;

cout << "---------------------------------------------------------------" << endl;

cin >> nemu;

switch (nemu) {

case '1':

cout << "---------------------------------------------------------------" << endl;

cout << "Int array: |" << endl;

printArray(arr1, size);

sortArray(arr1, size);

cout << "Sorted int array: |" << endl;

printArray(arr1, size);

cout << " |" << endl;

break;

case '2':

cout << "---------------------------------------------------------------" << endl;

cout << "Unsigned int array: |" << endl;

printArray(arr2, size);

sortArray(arr2, size);

cout << "Sorted unsigned int array: |" << endl;

printArray(arr2, size);

cout << " |" << endl;

break;

case '0':

system("cls");

break;

default:

cout << "!!!!! INVALID CHOICE !!!!! TRY AGAINE !!!!!" << endl;

}

} while (nemu != '0');

break;

case '2':

system("cls");

do {

cout << "---------------------------------------------------------------" << endl;

cout << "1. Test int deque |" << endl;

cout << "2. Test unsigned int deque |" << endl;

cout << "0. Back |" << endl;

cout << "---------------------------------------------------------------" << endl;

cin >> unem;

switch (unem) {

case '1':

cout << "---------------------------------------------------------------" << endl;

do {

cout << "Enter size deque: ";

cin >> s;

if (s < 1) {

cout << "Too small value\n";

}

} while (s < 1);

cout << "Enter " << s << " int elements: " << endl;

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

cin >> x;

deque1.pushRear(x);

}

cout << "Deque after push: |" << endl;

deque1.print();

front1 = deque1.getFront();

rear1 = deque1.getRear();

deque1.popFront();

deque1.popRear();

cout << "Deque after pop front and rear: |" << endl;

deque1.print();

cout << "Popped elements: |" << endl;

cout << "Front: " << front1 << " |" << endl;

cout << "Rear: " << rear1 << " |" << endl;

break;

case '2':

cout << "---------------------------------------------------------------" << endl;

do {

cout << "Enter size deque: ";

cin >> s;

if (s < 1) {

cout << "Too small value" << endl;

}

} while (s < 1);

cout << "Enter " << s << " int elements: " << endl;

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

do {

cin >> x;

if (x < 0) {

cout << "Too small value" << endl;

}

else {

deque2.pushRear(x);

}

} while (x < 0);

}

cout << "Deque after push: |" << endl;

deque2.print();

front2 = deque2.getFront();

rear2 = deque2.getRear();

deque2.popFront();

deque2.popRear();

cout << "Deque after pop front and rear: |" << endl;

deque2.print();

cout << "Popped elements: |" << endl;

cout << "Front: " << front2 << " |" << endl;

cout << "Rear: " << rear2 << " |" << endl;

break;

case '0':

system("cls");

break;

default:

cout << "!!!!! INVALID CHOICE !!!!! TRY AGAINE !!!!!" << endl;

}

} while (unem != '0');

break;

case '0':

cout << "--------------------- Exiting programm.... --------------------" << endl;

break;

default:

cout << "!!!!! INVALID CHOICE !!!!! TRY AGAINE !!!!!" << endl;

}

} while (menu != '0');

}

**Результат работы программы:**

При запуске программы появляется меню, что показано на рисунке 1.



Рисунок 1 − Меню программы

При нажатии на «1» в главном меню можно отсортировать два дека. Появляется меню выбора между int деком и unsigned int деком, что показано на рисунке 2.

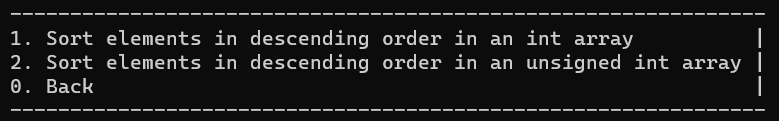


Рисунок 2 − Меню, появляющееся после нажатия «1» в главном меню

При выборе «1» можно отсортировать по неубыванию массив, состоящий из элементов типа int. Выводится имеющийся массив и отсортированный массив, что показано на рисунке 3.



Рисунок 3 − Неотсортированный и отсортированный массивы

При выборе «2» можно отсортировать по неубыванию массив, состоящий из элементов типа unsigned int. Выводится имеющийся массив и отсортированный массив, что показано на рисунке 4.

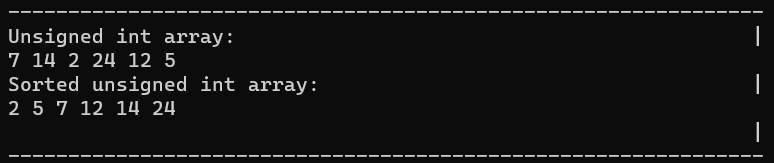


Рисунок 4 − Неотсортированный и отсортированный массивы

При выборе несуществующего пункта меню выведется сообщение об ошибке, что показано на рисунке 5.



Рисунок 5 − Выбор несуществующего пункта меню

При выборе «0» можно вернуться в главное меню.

При нажатии «2» в главном меню можно протестировать работу дека. Появится выбор между деком для int элементов и деком для unsigned int элементов, что показано на рисунке 6.

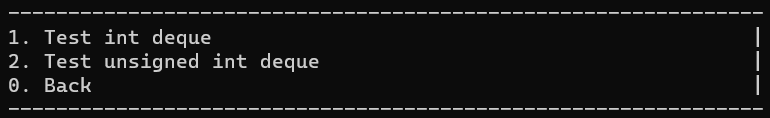


Рисунок 6 − Меню, появляющееся после нажатия «2» в главном меню

При выборе «1» можно проверить, что дек работает верно. С клавиатуры нужно ввести размер дека и его значения. Затем будет выведен дек без первого и последнего элемента, а потом будут выведены удаленный элемент с начала дека и удаленный элемент с конца дека, что показано на рисунке 7.

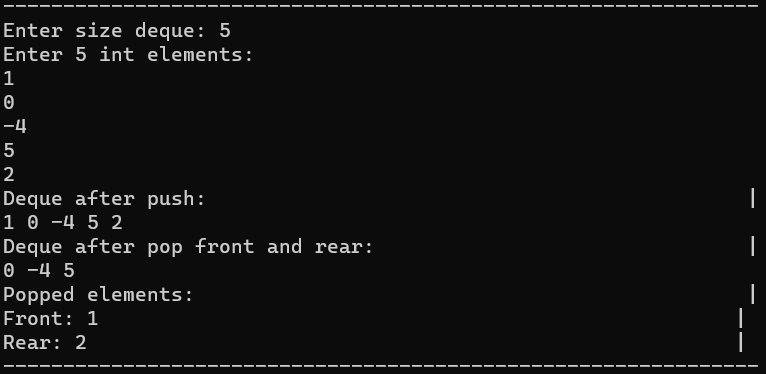


Рисунок 7 − Тестирование дека для int элементов

Если ввести размер, меньше 1, выведется ошибка, что показано  
на рисунке 8.

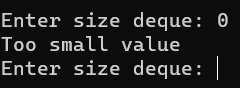


Рисунок 8 − Значение размера дека меньше 1

При выборе «2» можно проверить, что дек работает верно. С клавиатуры нужно ввести размер дека для unsigned int элементов и значения, затем будет выведен дек без первого и последнего элемента, а потом будут выведены удаленный элемент с начала дека и удаленный элемент с конца дека, что показано на рисунке 9.

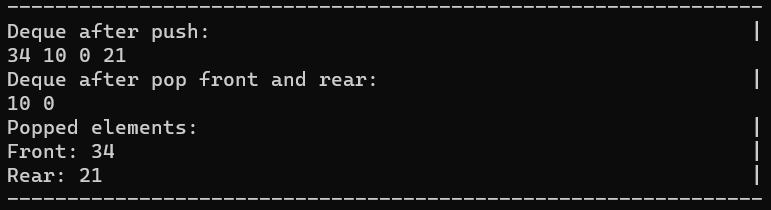


Рисунок 9 − Тестирование дека для unsigned int элементов

Если ввести размер меньше 1, выведется ошибка, что показано  
на рисунке 10.

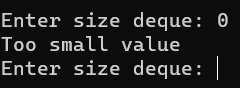


Рисунок 10 − Значение размера дека меньше 1

Если ввести значение меньше 0, появится ошибка, что показано  
на рисунке 11.

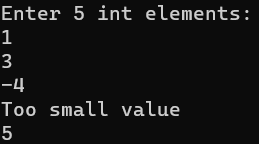


Рисунок 11 – Значение меньше 0

При выборе несуществующего пункта меню выведется сообщение об ошибке, что показано на рисунке 12.



Рисунок 12 − Выбор несуществующего пункта меню

При выборе «0» можно вернуться в главное меню.

При выборе несуществующего пункта в главном меню выведется сообщение об ошибке, что показано на рисунке 13.



Рисунок 13 − Выбор несуществующего пункта меню

При нажатии на «0» в главном меню, работа программы завершится, что показано на рисунке 14.



Рисунок 14 − Завершение программы