|  |  |
| --- | --- |
| для прик эмбл | |
| МИНОБРНАУКИ РОССИИ | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«Московский технологический университет»**  **МИРЭА** | |
|  | Институт информационных технологий |
|  | Кафедра инструментального и прикладного программного обеспечения |

|  |  |
| --- | --- |
| **ОТЧЁТ**  **ЛАБОРАТОРНАЯ РАБОТА №1** | |
| **дисциплина** | |
| **«Разработка программного приложения»**  *(наименование дисциплины)* | |
|  | |
| Выполнил студент группы \_ИКБО-06-16\_\_\_  *(учебная группа)* | *Кондратенко И.А.* |
| Принял доцент кафедры ИТС, доцент, к.т.н.  *должность, звание, ученая степень* | *Торхов Е.А.* |

|  |  |  |
| --- | --- | --- |
| Работы выполнены | «\_\_»\_\_\_\_\_\_\_201\_\_ г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_201\_\_ г. | *(подпись руководителя)* |

Москва 2018

1. **Цель**

Изучение постраничной навигации, корректность её отображения в пограничных ситуациях.

1. **Задание**

Отобразить постраничную навигацию, где общее количество страниц и номер текущей страницы задаёт пользователь.

1. **Описание программы**

Для реализации программы была выбрана следующая схема:

1 … k-1 [k] k+1 … N

Где k – номер выбранной страницы, а N – количество страницы (т.е. номер последней из них)

1. **Код программы**

#include <iostream>

#include <string.h>

#include <sstream>

using std::cin;

using std::cout;

using std::string;

const int prev = -1;

const int next = 1;

const int outPageSize = 5;

class Page{

private:

int \*pages;

int \*outPages;

int size;

void calc(int position);

public:

Page(int num\_pages);

string calculation(int position);

void showPageModel(string str);

~Page();

};

class System{

private:

bool repeat();

public:

void init();

void input(int &size, int &page);

string toString(int val);

};

int main(void){

System \*sys = new System();

sys->init();

return 0;

}

Page::Page(int num\_pages){

pages = new int[num\_pages];

outPages = new int[outPageSize];

for(int i = 0; i < num\_pages; i++){

pages[i] = i+1;

}

this->size = num\_pages;

}

void Page::showPageModel(string str){

cout << str;

}

string Page::calculation(int position){

string model;

System \*sys = new System;

if(position >= size) model = "Value is out of range!!!";

else{

if(size > 5){

if((size >= 7) && (position >= 3) && (position < size-3)){

model = model + sys->toString(pages[0]) + " ... ";

for(int i = prev; i < next + 1; i++){

model = model + " " + sys->toString(pages[position+i]);

}

model = model + " ... " + sys->toString(pages[size-1]);

}

else if(position < 3){

for(int i = 0; i < position + 2; i++){

model = model + sys->toString(pages[i]) + " ";

}

model = model + " ... " + sys->toString(pages[size-1]);

}

else if((position >= size - 3) && (position < size)){

model = model + sys->toString(pages[0]) + " ... ";

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

model = model + sys->toString(pages[i]) + " ";

}

}

}

else{

if(((size == 5) && (position < 2)) || ((size == 4) && (position == 0))){

for(int i = 0; i < position + 2; i++){

model = model + sys->toString(pages[i]);

if(i < position + 1) cout << " ";

else model = model + " ... " + sys->toString(pages[size-1]);

}

}

else if(((size == 5) && (position > size - 3)) ||((size == 4) && (position == size- 1))){

model = model + sys->toString(pages[0]) + " ... ";

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

model = model + sys->toString(pages[i]) + " ";

}

}

else{

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

model = model + sys->toString(pages[i]) + " ";

}

}

}

}

delete sys;

return model;

}

Page::~Page(){

delete []pages;

}

void System::input(int &size, int &page){

while(size < 1){

cout << "Enter the size of array: ";

cin >> size;

}

while(page < 1){

cout << "\n\nEnter the page: ";

cin >> page;

cout << "\n";

}

}

bool System::repeat(){

cout << "\n\nDo you want to repeat: ";

char asw = 0;

cin >> asw;

cout << "\n\n";

if(asw == 'Y' || asw == 'y') return true;

else if(asw == 'N' || asw == 'n') return false;

else repeat();

}

void System::init(){

int size = 0;

int page = 0;

input(size,page);

Page \*page\_model = new Page(size);

page\_model->showPageModel(page\_model->calculation(page-1));

delete page\_model;

if (repeat() == true) init();

else{}

}

string System::toString(int val){

std::ostringstream oss;

oss << val;

return oss.str();

}

**Скриншоты и примеры работы**

Результат работы программы представлен на Рис 1-2.

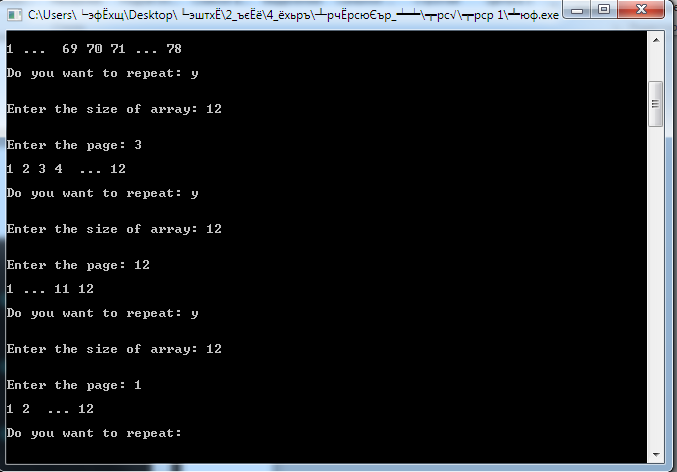


Рис. 1 – запрос 4 видов, отображающих все виды запросов, нуждающихся в отдельной обработке

1. **Заключение**

В результате данной лабораторной работы произошло знакомство с постраничной навигацией, были выявлены пограничные случаи и обработаны.

1. **Используемая литература**

*Герберт Шилдт.* Теория и практика С++ = Shildt's Expert C++. — СПб.: BHV — Санкт- Петербург, 1996.

Qt 5.3 Профессиональное программирование на C++. – СПб.:БХВ-Петербург, 2015. - 928 с.

Дейт К. Дж*.* Введение в системы баз данных — 8-е изд. — М.: Вильямс, 2005. — 1328 с.

КузнецовС. Д. Основы баз данных. — 2-е изд. — М.: Интернет-университет информационных технологий; БИНОМ. Лаборатория знаний, 2007. — 484 с