Лабораторная работа №1

По дисциплине «Информатика»

Тема: Объявление класса при помощи С++

Выполнил:

студент группы [EMPTY]

[YOUR NAME]

Принял:

[TEACHER NAME]

**Цель работы**: изучить структуру классов, особенности реализации методов класса и создания объектов; приобрести навыки в работе с классами на языке C++.

Пример определения класса и работы с его объектом

**Структура программы**

* deps.h - содержит макросы, что позволяет программе работать как на Windows так и на Unix-подобных платформах
* deps.cpp - содержит определения Windows-зависимых функций
* array.h - содержит объявления функций для работы с массивом
* array.cpp - содержит определения функций для работы с массивом
* main.cpp - содержит интерфейс для работы с массивами

**Ниже представлен код программы**

// deps.h

#ifndef \_\_DEPS\_H

#define \_\_DEPS\_H

# ifdef \_\_WIN32

# include <windows.h>

# include <conio.h>

# define KB\_ENTER 13

# else

# include <unistd.h>

# include <termios.h>

# include <sys/ioctl.h>

# define KB\_ENTER 10

int putch (int c);

int getch (void);

int kbhit (void);

# endif

#define KB\_ESCAPE 27

#endif

// deps.cpp

#include <cstdio>

#include "deps.h"

#ifndef \_\_WIN32

int putch (int c)

{

return putchar (c);

}

int kbhit(void) {

static const int STDIN = 0;

static bool initialized = false;

if (! initialized) {

// Use termios to turn off line buffering

termios term;

tcgetattr(STDIN, &term);

term.c\_lflag &= ~ICANON;

tcsetattr(STDIN, TCSANOW, &term);

setbuf(stdin, NULL);

initialized = true;

}

int bytesWaiting;

ioctl(STDIN, FIONREAD, &bytesWaiting);

return bytesWaiting;

}

int getch (void)

{

int ch;

struct termios oldtc;

struct termios newtc;

tcgetattr(STDIN\_FILENO, &oldtc);

newtc = oldtc;

newtc.c\_lflag &= ~(ICANON | ECHO);

tcsetattr(STDIN\_FILENO, TCSANOW, &newtc);

ch=getchar();

tcsetattr(STDIN\_FILENO, TCSANOW, &oldtc);

return ch;

}

#endif

// array.h

#ifndef \_\_LAB1\_H

#define \_\_LAB1\_H

// объявление класса

class CArrayRemoveIfOdd {

//объявление приватных членов класса

//поля класса:

int nCount; //реальная длина массива; если массив пуст – nCount=0

int \*arr; // указатель на массив

public: //объеялание общедоступных членов класса

// методы класса::

CArrayRemoveIfOdd(); //конструктор; вызвается, когда создаётся объект класса

~CArrayRemoveIfOdd();//деструктор; вызвается, когда уничтожается объект класса

void CreateArray(int N); //создание массива

void SetValue(int k, int value); //установка элемента массива

void FillArrayRandValues(); //заполнение массива случайными числами

void DisplayArray(); // вывод элементов массива на экран

void RemoveIfOdd (void); // удалить четные элементы массива

};

#endif

// array.cpp

#include <stdlib.h>

#include <cstdio>

#include "array.h"

CArrayRemoveIfOdd::CArrayRemoveIfOdd() //вызов конструктора

{

nCount=0; //начальные значения полей класса: массив пуст

arr=NULL;

}

CArrayRemoveIfOdd::~CArrayRemoveIfOdd()

{

delete [] arr; //освобождение памяти, занимаемой массивом

}

void CArrayRemoveIfOdd::CreateArray(int N)

{

if (N>0) { //если задан корректно новый размер массива – создаём его

if (arr!=NULL) //если массив уже существовал – уничтожаем его

delete [] arr;

arr=new int[N];

nCount=N;

printf("Массив создан\n");

}

else {

printf("Ошибка! Размер массива должен быть больше 0\n");

}

}

void CArrayRemoveIfOdd::SetValue(int k, int value)

{

if (k>=0 && k<nCount) //если элемент с индексом k существует, меняем его

arr[k]=value;

else

printf("Ошибка! Задан неверный индекс массива\n");

}

void CArrayRemoveIfOdd::FillArrayRandValues()

{

if(nCount>0) { //если массив создан

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

arr[i]=rand()%100;

}

DisplayArray();

}

else {

printf("Массив не создан\n");

}

}

void CArrayRemoveIfOdd::DisplayArray()

{

if(nCount > 0) { //если массив создан

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

printf("%d ", arr[i]);

printf("\n");

}

}

else {

printf("Массив не создан\n");

}

}

void CArrayRemoveIfOdd::RemoveIfOdd (void)

{

int i = 0;

int j = 0;

int \*tmparr, tmpsize;

tmpsize = nCount / 2 + 1;

tmparr = new int[tmpsize];

for ( ; i < nCount; ++i) {

if (i % 2 == 1) {

/\* copy odd index values into temp array. \*/

tmparr[j] = arr[i];

j++;

}

}

/\* Delete previously allocated memory for array. \*/

delete [] arr;

/\* Assign pointer to newly created array. \*/

arr = tmparr;

nCount = j;

}

// main.cpp

#include <iostream>

#include <locale.h>

#include <cstdlib>

#include <cstdio>

#include <time.h>

#include "array.h"

#include "deps.h"

int main (void)

{

int key, n, k, val;

CArrayRemoveIfOdd ar; // создаётся объект класса, при этом автоматически вызывается конструктор

srand(time(NULL)); // включаем генератор случайных чисел

setlocale(0, "RUS"); // обеспечиваем вывод в консольное окно символов кириллицы

do {

printf("\n1 - создание массива\n");

printf("2 - изменение элементов массива\n");

printf("3 - заполнение массива случайными числами\n");

printf("4 - вывод на экран элементов массива\n");

printf("5 - удаление элементов массива на четных местах\n");

printf("ESC - выход\n");

key = getch();

switch(key)

{

case '1':

printf("Введите количество элементов массива: ");

scanf("%d", &n);

ar.CreateArray(n);

break;

case '2':

printf("Введите номер элемента массива: ");

scanf("%d", &k);

printf("Введите значение элемента массива: ");

scanf("%d",&val);

ar.SetValue(k, val);

break;

case '3':

ar.FillArrayRandValues();

break;

case '4':

ar.DisplayArray();

break;

case '5':

ar.RemoveIfOdd ();

break;

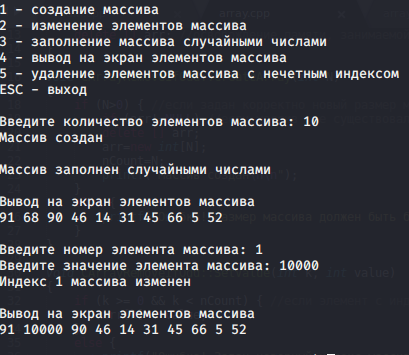
}

} while (key != KB\_ESCAPE); // код клавиши ESC

return 0; // объект класса уничтожается, автоматически вызывается деструктор

}

Экранные формы с примерами работы программы



Pис. 1

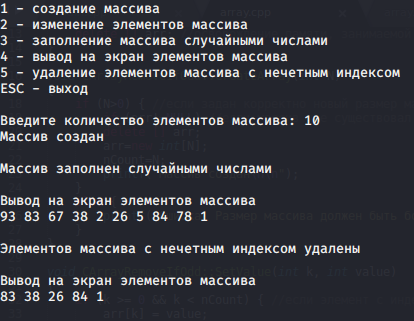


Рис. 2

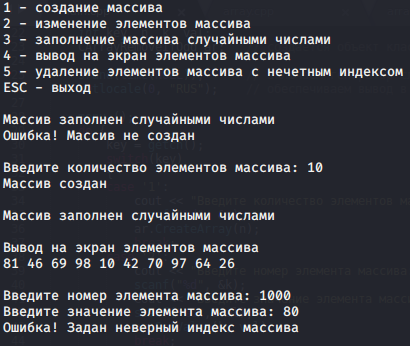


Рис. 3

Рис. 1 показывает базовые операции, который можно произвести над массивами, такие как создание массива, заполнение случайными числами, добавление и удаление элементов.

Рис. 2 - удаление элементов массива стоящих на нечетных местах (вар 6). После создания массива и заполнения случайными числами (не обязательно) можно приступить к уделению данных элементов. Операция совершается при нажатии клавиши 5. На Рис. 2 можно увидеть что после данной операции количество элементов массива уменьшилось вдвое.

Рис. 3 показывает способность программы корректно обрабатывать ошибки, такие как удаление не существующего элемента массива, удаление элементов массива, который еще не создан. Программа способна не только выявить данные ошибки, но и вывести соответствующее сообщение пользователю.

Заключение

В данной лабораторной работе я научился работать с массивами, производить операции добавления, удаления элементов, вывода элементов на экран с помощью программы написанной мною. Изучил структуру классов, особенности реализации методов класса и создания объектов. Приобрел навыки в работе с классами на языке C++