Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования «Рязанский государственный радиотехнический

университет имени В. Ф. Уткина»

Кафедра «Вычислительная и прикладная математика»

**Отчет**

**о лабораторной работе №5**

**по дисциплине**

«Объектно-ориентированное программирование»

**на тему**

«Шаблонные классы»

Выполнил: студент гр. 243

Трынкин Михаил Александрович

Проверил:

Каширин Игорь Юрьевич

Рязань 2023

**Цель работы:**

Получить навыки составления алгоритмов и программ на языке С++, используя шаблонные классы.

**Задание**

В данной лабораторной работе необходимо разработать шаблонный класс в соответствии с вариантом задания. Любой класс должен иметь конструктор копии и перегруженный оператор =. Также класс должен содержать перегрузку оператора << для класса iostream, чтобы его можно было выводить в консоль через стандартный поток вывода cout с указанием количества элементов и их значений. В главной функции main обеспечить консольный интерфейс для тестирование всех функций шаблонного класса с типами int, float, char\*, struct Vec2 {float x; float y;} (аналогично лаб 1)

**Вариант 20.**

**Динамически-инициализированный массив,** количество элементов задаётся через non-type параметр, выделение памяти происходит в конструкторе при создании объекта. Необходимые методы:

**sort(bAscending) – сортировать элементы по возрастанию (bAscending=true**), либо по убыванию. Для типа char\* должна быть своя специализация шаблона, сортирующая по длине строк. **getMaxSize()** – получить максимальный размер массива

**getLength()** – получить текущее кол-во элементов

**insert(elem,n)** – добавить элемент на позицию n

**remove(elem,n)** – удалить элемент на позиции n

Перегрузить следующие операции: **+** – добавить элемент в конец массива **-** – удалить первое вхождение элемента в массиве (для типа char\* должна быть своя специализация шаблона) **[]** – доступ к элементу по индексу **< и >** – проверка, что в двух одинакового размера массивов все элементы больше или меньше элементов другого массива (для типа char\* и Vec2 должна быть своя специализация шаблона: char\* исходя из длины строки, для Vec2 – исходя из длины вектора).

**Листинг программы:**

struct Vec2

{

float x = 0, y = 0;

// перегрузка для =

Vec2& operator=(float value)

{

x = y = value;

return \*this;

}

// перегрузка для ==

bool operator == (Vec2 b){

return (sqrt(this->x\*this->x + this->y\*this->y) == sqrt(b.x\*b.x + b.y\*b.y));

}

// перегрузка для <=

bool operator <= (Vec2 b){

return (sqrt(this->x\*this->x + this->y\*this->y) < sqrt(b.x\*b.x + b.y\*b.y));

}

// перегрузка для >=

bool operator >= (Vec2 b){

return (sqrt(this->x\*this->x + this->y\*this->y) > sqrt(b.x\*b.x + b.y\*b.y));

}

};

template <typename T, int maxSize>

class DynArray

{

public:

T\* array;

int len;

DynArray()

{

array = new T[maxSize];

//занулим элементы

for(int i = 0; i < maxSize; ++i)

array[i] = 0;

}

DynArray(DynArray& arr)

{

array = new T[maxSize];

this->setLen(arr.getLen());

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

array[i] = arr.array[i];

}

}

~DynArray()

{

delete[] array;

}

void setLen(int n){ this->len = n;}

int getLen(){ return len; }

int getMaxSize() {

return maxSize;

}

int chooseOperation(){

int operation = 0;

cout<<("Какая операция вас интересует?")<< endl;

do{

printChoise();

cin>>operation;

if ((operation > 9) or (operation < 0)) cout<<("Такая операция не предусмотрена")<< endl;

} while ((operation > 9) or (operation < 0));

cout<<endl;

return operation;

}

void sort(bool asc){

if (asc){

int i, j;

bool swapped;

for (i = 0; i < len - 1; i++) {

swapped = false;

for (j = 0; j < len - i - 1; j++) {

if (array[j] >= array[j + 1]) {

swap(array[j], array[j + 1]);

swapped = true;

}

}

// Если во вложенном цикле ничего не изменилось

// то прерываем

if (swapped == false)

break;

}

}

else{

int i, j;

bool swapped;

for (i = 0; i < len - 1; i++) {

swapped = false;

for (j = 0; j < len - i - 1; j++) {

if (array[j] <= array[j + 1]) {

swap(array[j], array[j + 1]);

swapped = true;

}

}

// Если во вложенном цикле ничего не изменилось

// то прерываем

if (swapped == false)

break;

}

}

}

void insertToN(T elem, int n){

for (int i = len-1; i >= n; --i)

swap(array[i+1], array[i]);

array[n] = elem;

++len;

}

void removeFromN(int n){

array[n] = 0;

for (int i = n; i < len; ++i)

swap(array[i+1], array[i]);

--len;

}

void insertToEnd(T elem){

array[len] = elem;

++len;

}

DynArray<T, maxSize>& operator +(const T elem);

void removeFirst(T elem){

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

if (array[i] == elem) {

removeFromN(i);

break;

}

}

}

DynArray<T, maxSize>& operator -(const T elem);

T& operator[](int index)

{

return array[index];

}

bool biggerThenAll( T\* arr2, bool bigger){

bool rez = true;

if (bigger == true){

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

for (int j = 0; j < len; ++j){

if (array[i] < arr2[j]){

rez = false;

break;

}

}

}

}

else{

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

for (int j = 0; j < len; ++j){

if (array[i] > arr2[j]){

rez = false;

break;

}

}

}

}

return rez;

}

bool operator >(const DynArray<T, maxSize> arr2);

bool operator <(const DynArray<T, maxSize> arr2);

void print();

void operations(short type){

short operation = chooseOperation();

switch (operation){

case 1:

{

print();

break;

}

case 2:

{

bool asc;

short how = sortType();

(how == 1) ? asc = true : asc = false;

sort(asc);

print();

break;

}

case 3:

{

cout<<("Максимальная длина массива = ")<<NONTYPE<<endl;

break;

}

case 4:

{

T elem;

string elem1;

cout<<("Введите элемент и позицию")<<endl;

cin>>elem;

int n = position(getLen());

print();

insertToN(elem,n);

print();

break;

}

case 5:

{

int n = position(getLen());;

print();

removeFromN(n);

print();

break;

}

case 6:

{

cout<<("Введите элемент ")<<endl;

T elem;

cin>>elem;

\*this + elem;

print();

break;

}

case 7:

{

cout<<("Имеется массив: ")<<endl;

print();

cout<<("Введите элемент ")<<endl;

T elem;

cin>>elem;

\*this - elem;

print();

break;

}

case 8:

{

switch (type){

case 1:{

DynArray<T, NONTYPE> array2;

array2.setLen(this->getLen());

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

array2[i] = rand() % 100;

}

cout<<("Исходный массив")<<endl;

print();

cout<<("Второй массив")<<endl;

array2.print();

cout<<("0 - проверка, больше ли все элементы 1 массива, чем 2")<<endl;

cout<<("1 - проверка, меньше ли все элементы 1 массива, чем 2")<<endl;

short ch;

cin>>ch;

switch(ch){

case 0:{

ch = \*this > array2;

break;

}

case 1:{

ch = \*this < array2;

break;

}

}

switch(ch){

case 0:{

cout<<("Неверно!")<<endl;

break;

}

case 1:{

cout<<("Верно!")<<endl;

break;

}

}

break;

}

case 2:{

DynArray<T, NONTYPE> array2;

array2.setLen(this->getLen());

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

array2[i] = rand() % 100 + rand() % 100 \* 0.1;

}

cout<<("Исходный массив")<<endl;

print();

cout<<("Второй массив")<<endl;

array2.print();

cout<<("0 - проверка, больше ли все элементы 1 массива, чем 2")<<endl;

cout<<("1 - проверка, меньше ли все элементы 1 массива, чем 2")<<endl;

short ch;

cin>>ch;

switch(ch){

case 0:{

ch = \*this > array2;

break;

}

case 1:{

ch = \*this < array2;

break;

}

}

switch(ch){

case 0:{

cout<<("Неверно!")<<endl;

break;

}

case 1:{

cout<<("Верно!")<<endl;

break;

}

}

break;

}

}

break;

}

case 9:

break;

}

}

};

// объявление метода вне описания класса-шаблона

template <typename T, int maxSize>

void DynArray<T, maxSize>::print()

{

cout<<"количество элементов = "<<len<<endl;

for(int i = 0; i < len; ++i)

cout << array[i] << " ";

cout << endl;

}

template < int maxSize>

void DynArray<Vec2, maxSize>::print()

{

cout<<"количество элементов = "<<len<<endl;

for(int i = 0; i < len; ++i)

cout << " vec: "<< array[i].x << ";"<< array[i].y << " ";

cout << endl;

}

//перегрузка операторов для основного шаблона

template<typename T, int maxSize>

DynArray<T, maxSize>& DynArray<T, maxSize>::operator +(const T elem){

this->insertToEnd(elem);

return \*this;

}

template<typename T, int maxSize>

DynArray<T, maxSize>& DynArray<T, maxSize>::operator -(const T elem){

this->removeFirst(elem);

return \*this;

}

template<typename T, int maxSize>

bool DynArray<T, maxSize>::operator >(const DynArray<T, maxSize> array2){

return this->biggerThenAll(array2.array, true);

}

template<typename T, int maxSize>

bool DynArray<T, maxSize>::operator <(const DynArray<T, maxSize> array2){

return this->biggerThenAll(array2.array, false);

}

//перегрузка операторов для строк

template<int maxSize>

DynArray<char\*, maxSize>& DynArray<char\*, maxSize>::operator +(const char\* elem){

this->insertToEnd(elem);

return \*this;

}

template<int maxSize>

DynArray<char\*, maxSize>& DynArray<char\*, maxSize>::operator -(const char\* elem){

this->removeFirst(elem);

return \*this;

}

template<int maxSize>

bool DynArray<char\*, maxSize>::operator >(const DynArray<char\*, maxSize> array2){

return this->biggerThenAll(array2.array, true);

}

template<int maxSize>

bool DynArray<char\*, maxSize>::operator <(const DynArray<char\*, maxSize> array2){

return this->biggerThenAll(array2.array, false);

}

//перегрузка операторов для векторов

template<int maxSize>

DynArray<Vec2, maxSize>& DynArray<Vec2, maxSize>::operator +(const Vec2 elem){

this->insertToEnd(elem);

return \*this;

}

template<int maxSize>

DynArray<Vec2, maxSize>& DynArray<Vec2, maxSize>::operator -(const Vec2 elem){

this->removeFirst(elem);

return \*this;

}

template<int maxSize>

bool DynArray<Vec2, maxSize>::operator >(const DynArray<Vec2, maxSize> array2){

return this->biggerThenAll(array2.array, true);

}

template<int maxSize>

bool DynArray<Vec2, maxSize>::operator <(const DynArray<Vec2, maxSize> array2){

return this->biggerThenAll(array2.array, false);

}

int main(int argc, char\*\* argv)

{

system("chcp 1251");

srand(time(0));

int len = 0;

int type;

cout<<("С каким типом данных хотите работать?")<< endl;

do{

cout<<("1 - целые числа")<< endl;

cout<<("2 - числа с плавающей точкой")<< endl;

cout<<("3 - строки")<< endl;

cout<<("4 - вектора vec(float x; float y)")<< endl;

cin>>type;

if ((type > 4) or (type <= 0)) cout<<("Такой вариант не предусмотрен!")<< endl;

} while ((type > 4) or (type <= 0));

cout << endl;

cout<<("Введите длину массива")<< endl;

do{

cin>>len;

if (len > 20) cout<<("Длина больше допустимой! Повторите ввод")<< endl;

if (len < 1) cout<<("Длина меньше допустимой! Повторите ввод")<< endl;

}while ((len > 20) or (len <= 0));

cout << endl;

switch (type){

case 1:

{

cout<<("Рассматривается тип данных int")<< endl;

DynArray<int, NONTYPE> array;

array.setLen(len);

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

array[i] = rand() % 100;

}

array.operations(1);

break;

}

case 2:

{

cout<<("Рассматривается тип данных float")<< endl;

DynArray<float, NONTYPE> array;

array.setLen(len);

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

array[i] = rand() % 100 + rand() % 100 \*0.01 ;

}

array.operations(2);

break;

}

case 3:

{

cout<<("Рассматривается тип данных char\*")<< endl;

DynArray<char\*, NONTYPE> array;

array.setLen(len);

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

short a = rand() % 7 + 1;

char \*temp = new char[a];

strcpy(temp,"");

array[i] = new char[a];

char const \*s = "S";

for (int j = 0; j < a; ++j){

strcat(temp, s);

}

strcpy(array[i], temp);

cout<<array[i]<< endl;

delete[] temp;

}

array.operations(3);

break;

}

case 4:

{

cout<<("Рассматривается тип данных Vec2")<< endl;

DynArray<Vec2, NONTYPE> array;

array.setLen(len);

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

array[i].x = rand() % 100;

array[i].y = rand() % 100;

}

array.operations(4);

break;

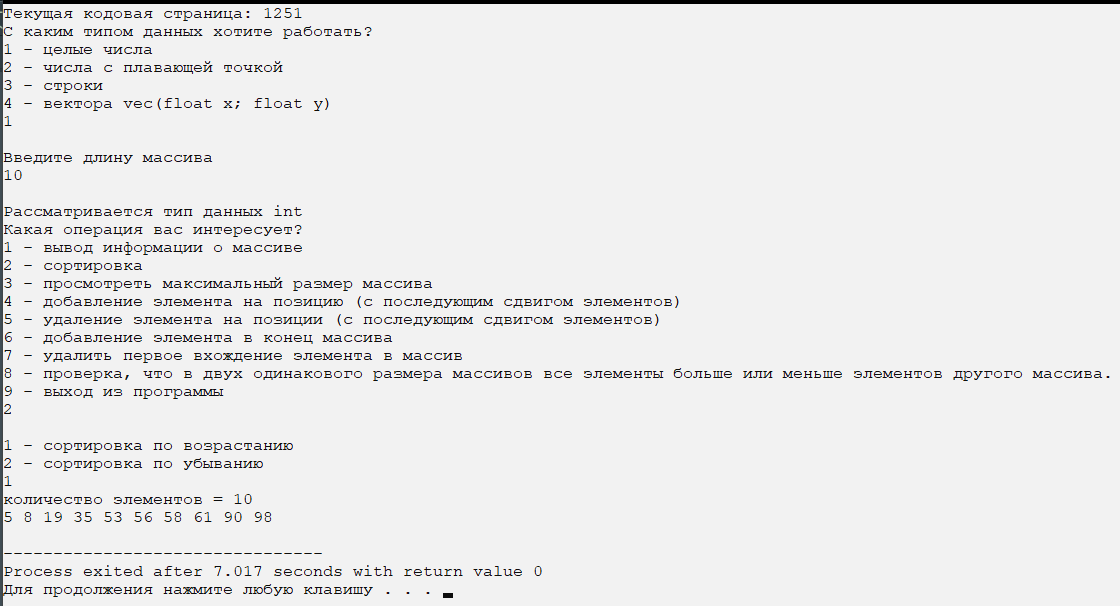
}

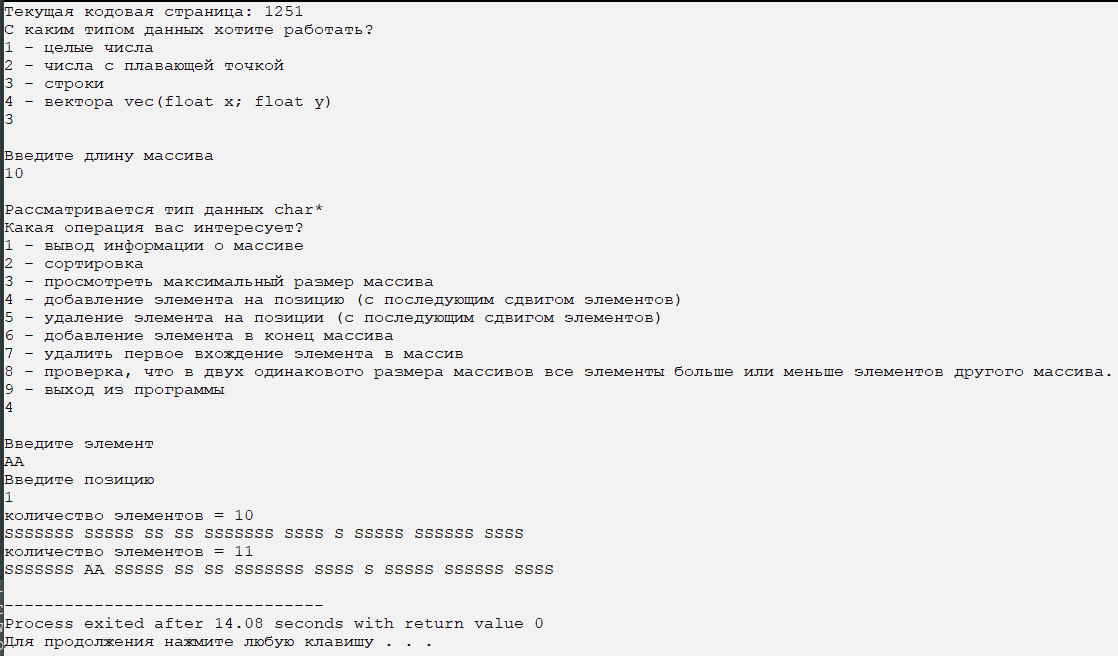
}

return 0;

}

**Результат работы программы:**

****

****

**Проверка:**

Реальные данные совпадают с ожидаемыми, программа работает корректно.

**Вывод:**

Я научился составлять алгоритмы с использованием шаблонных классов реализовывать их на языке программирования С++.