ЛАБОРАТОРНАЯ РАБОТА №7

“Программирование с использованием конструкторов и деструкторов”

**Цель работы**: сформировать знания и умения по создание и использованию конструкторов и

деструкторов классов в ЯП Си++, приобрести практические навыки создания программ на языке Си++ с

использованием классов, содержащих конструкторы и деструкторы.

**Задание 2:**

#include<iostream>

using namespace std;

class dinamicArray {

int\* p = new int;// выделили память для переменной целого типа

int lenght;// размер

int\* a;//указатель на массив целых чисел в динамической памяти

public: dinamicArray(int n);//конструктор

int sum();// сумма элементов массива

void output();//вывод массива

~dinamicArray() { delete a; };//деструктор

};

dinamicArray::dinamicArray(int n) {//конструктор

lenght = n;//устанавливаем длину

a = new int[n];// выделяем память

for (int i = 0; i < lenght; i++)a[i] = i;//заполняем массив

}

int dinamicArray::sum() {//сумма элементов массива

int s = 0; // вспомогательная переменная для накопления суммы

for (int i = 0; i < lenght; i++)s += a[i];

return s;

}

void dinamicArray::output() {//вывод элементов массива

cout << endl;

for (int i = 0; i < lenght; i++)cout << a[i] << " ";

delete p; //освободили память, на которую указывает указатель p.

}

int main() {

dinamicArray A(5);//создаем массив из пяти элементов

cout << "Array A" << endl;

A.output();//вывод элементов массива

cout << "Sum of elements " << A.sum();//вывод суммы элементов

return 0;

}

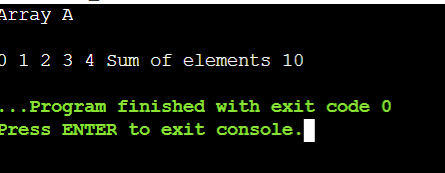


Рисунок 1 – выполнение кода примера 1

**Задание 3:**

#include<iostream>

using namespace std;

class fraction {

int m; //числитель

int n; //знаменатель

public:

fraction();

void output();// вывод дроби

void mult(int a);// умножение дроби на целое число

~fraction();

};

//Программируем методы класса fraction

fraction::fraction() {

cout << "Vvedite chislitel: "; cin >> m;

cout << "Vvedite znamenatel: "; cin >> n;

}

void fraction::output() {// вывод дроби

cout << m << "/" << n;

}

void fraction::mult(int a) {// умножение дроби на число

m = m \* a;

}

//Программируем алгоритм решения основной задачи

int main() {

// ввод дроби

cout << "Vvedite drob': " << endl;

fraction A; //дробь

int k; // число на которое нужно умножить дробь

/\*Чтобы ввести дробь А, нужно у объекта A вызвать метод input. \*/

//умножение дроби на целое число

cout << "Vvedite chislo na kotoroe nuzno ymnozit' drob'";

cin >> k;

/\*Чтобы умножить дробь А на число k нужно у объекта A вызвать \*/

A.mult(k);//вызываем метод mult для дроби A

// вывод дроби

A.output();//вызываем метод output для дроби A

return 0;

}

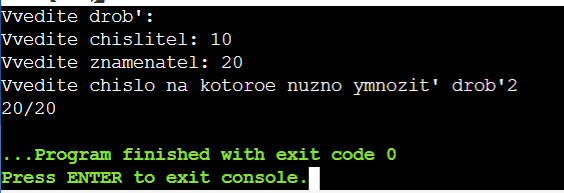


Рисунок 2 – выполнение кода

**Контрольные вопросы**

**1.Какие названия должны иметь функция конструктор и функция деструктор?**

Название класса.

**2.Как вызвать конструктор и деструктор в главной программе?**

Конструктор срабатывает при выполнении операции определения типа переменной. Вызывается функция-конструктор в тот момент, когда создается объект, т.е. для объекта выделяется место в памяти.

**3.Приведите примеры классов, в которых обязательно должны присутствовать конструктор и**

**деструктор?**

Рассмотрим пример, в котором невозможно обойтись без использования конструктора и деструктора. Необходимо создать класс динамический массив. Свойства – длина и указатель на область памяти, занятую под элементы. Методы: заполнение массива, нахождение суммы элементов массива, вывод на экран элементов массива.

**4.Может ли в классе быть несколько конструкторов?**

Нет, может быть только один конструктор.