Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №4

По дисциплине «Основы алгоритмизации и программирования»

За 1 семестр

Тема: «Знакомство с С++. Выполнение программы простой структуры»

Выполнил:

Студент 1 курса

Группы ИИ-21(1)

Кирилович А.А.

Проверил:

Скарубо А. О.

Брест 2021

**Цель работы:** научиться пользоваться языком C++, производить его анализ и коррекцию. Усвоить и закрепить базовые знания по работе с данным языком программирования.

Постановка задачи

1. 1.

#include <iostream>

using namespace std;

int main()

{

int x;

x = -3 + 4 \* 5 - 6; cout << x << endl; //(Операции 1.1)

//Выведет 11

x = 3 + 4 % 5 - 6; cout << x << endl; //(Операции 1.2)

//Выведет 1

x = -3 \* 4 % -6 / 5; cout << x << endl; //(Операции 1.3)

//Выведет 0

x = (7 + 6) % 5 / 2; cout << x << endl; //(Операции 1.4)

//Выведет 1

return 0;

}

1. 2.

#include <iostream>

using namespace std;

int main()

{

int x = 2, y, z;

x \*= 3 + 2; cout << x << endl; //(Операции 2.1)

//Выведет 10

x \*= y = z = 4; cout << x << endl; //(Операции 2.2)

//Выведет 40

x = y == z; cout << x << endl; //(Операции 2.3)

//Выведет 1

x == (y = z); cout << x << endl; //(Операции 2.4)

//Выведет 1

return 0;

}

1. 3.

#include <iostream>

using namespace std;

int main()

{

int x, y, z;

x = 2; y = 1; z = 0;

x = x && y || z; cout << x << endl; //(Операции 3.1)

//Выведет 1

cout << (x || y && z) << endl; //(Операции 3.2)

//Выведет 1

x = y = 1;

z = x++ - 1; cout << x << " " << z << endl; //(Операции 3.3)

//Выведет 2 и 0

z += -x++ + ++y; cout << x << " " << z << endl; ; //(Операции 3.4)

//Выведет 3 и 0

z = x / ++x; cout << z << endl; //(Операции 3.5)

//Выведет 1

return 0;

}

1. 4.

#include <iostream>

using namespace std;

int main()

{

int x, y, z;

x = 03; y = 02; z = 01;

cout << (x | y & z) << endl; //(Операции 4.1)

//Выведет 3

cout << (x | y & ~z) << endl; //(Операции 4.2)

//Выведет 3

cout << (x ^ y & ~z) << endl; //(Операции 4.3)

//Выведет 1

cout << (x & y && z) << endl; //(Операции 4.4)

//Выведет 1

x = 1; y = -1;

cout << (!x | x) << endl; //(Операции 4.5)

//Выведет 1

cout << (~x | x) << endl; //(Операции 4.6)

//Выведет -1

cout << (x ^ x) << endl; //(Операции 4.7)

//Выведет 0

x <<= 3; cout << x << endl; //(Операции 4.8)

//Выведет 8

y <<= 3; cout << y << endl; //(Операции 4.9)

//Вывведет -8

y >>= 3; cout << y << endl; //(Операции 4.10)

//Выведет -1

return 0;

}

1. 5.

#include <iostream>

using namespace std;

int main()

{

int x = 1, y = 1, z = 1;

x += y += z;

cout << (x < y ? y : x) << endl; //(Операции 5.1)

//Выведет 3

cout << (x < y ? x++ : y++) << endl;

cout << x << endl;

cout << y << endl; //(Операции 5.2)

/\*Выведет 2

3

3

\*/

cout << (z += x < y ? x++ : y++) << endl;

cout << y << endl;

cout << z << endl; //(Операции 5.3)

/\*Выведет 4

4

4

\*/

x = 3; y = z = 4;

cout << ((z >= y >= x) ? 1 : 0) << endl; //(Операции 5.4)

//Выведет 0

cout << (z >= y && y >= x) << endl; //(Операции 5.5)

//Выведет 1

return 0;

}

1. 6.

#include <iostream>

using namespace std;

int main()

{

int x, y, z;

x = y = z = 1;

++x || ++y && ++z; cout << x << y << z << endl; //(Операции 6.1)

//Выведет 211

x = y = z = 1;

++x&& ++y || ++z; cout << x << y << z << endl; //(Операции 6.2)

//Выведет 221

x = y = z = 1;

++x&& ++y&& ++z; cout << x << y << z << endl; //(Операции 6.3)

//Выведет 222

x = y = z = -1;

++x&& ++y || ++z; cout << x << y << z << endl; //(Операции 6.4)

//Выведет 0-10

x = y = z = -1;

++x || ++y && ++z; cout << x << y << z << endl; //(Операции 6.5)

//Выведет 00-1

x = y = z = -1;

++x&& ++y&& ++z; cout << x << y << z << endl; //(Операции 6.6)

//Выведет 0-1-1

return 0;

}

1. 7.

#include <iostream>

using namespace std;

int integer = 5;

char character = '5';

char sstring = \*"5";

int main()

{

cout << sstring << endl;

cout << character << endl;

cout << integer << endl;

cout << sstring << endl;

cout << ('5' > 5) << endl; //(Основные типы 1.1)

/\*Выведет 5

5

5

5

1

\*/

int sx = -8;

unsigned ux = -8;

cout << sx << endl;

cout << ux << endl;

cout << (sx >> 3) << endl;

cout << (ux >> 3) << endl;

cout << (sx >> 3) << endl;

cout << (ux >> 3) << endl; //(Основные типы 1.2)

/\*Выведет -8

4294967288

-1

536870911

-1

536870911

\*/

return 0;

}

1. 8.

#include <iostream>

using namespace std;

int main()

{

double d;

float f;

long l;

int i;

i = l = f = d = 100 / 3;

cout << (double)i << endl;

cout << (double)l << endl;

cout << (double)f << endl;

cout << (double)d << endl; //(Основные типы 2.1)

/\*Выведет 33

33

33

33

\*/

d = f = l = i = 100 / 3;

cout << (double)i << endl;

cout << (double)l << endl;

cout << (double)f << endl; //(Основные типы 2.2)

cout << (double)d << endl;

/\*Выведет 33

33

33

33

\*/

i = l = f = d = 100 / 3.;

cout << (double)i << endl;

cout << (double)l << endl;

cout << (double)f << endl; //(Основные типы 2.3)

cout << (double)d << endl;

/\*Выведет 33

33

33.3333

33.3333

\*/

d = f = l = i = (double)100 / 3;

cout << (double)i << endl;

cout << (double)l << endl;

cout << (double)f << endl; //(Основные типы 2.4)

cout << (double)d << endl;

/\*Выведет 33

33

33

33

\*/

i = l = f = d = (double)(100000 / 3);

cout << (double)i << endl;

cout << (double)l << endl;

cout << (double)f << endl; //(Основные типы 2.5)

cout << (double)d << endl;

/\*Выведет 33333

33333

33333

33333

\*/

d = f = l = i = 100000 / 3;

cout << (double)i << endl;

cout << (double)l << endl;

cout << (double)f << endl; //(Основные типы 2.6)

cout << (double)d << endl;

/\*Выведет 33333

33333

33333

33333

\*/

return 0;

}

1. 9.

#include <iostream>

using namespace std;

int main()

{

double d = 3.2, x;

int i = 2, y;

x = (y = d / i) \* 2;

cout << (double)(x) << endl;

cout << (double)(y) << endl;//(Основные типы 3.1)

/\*Выведет 2

1

\*/

y = (x = d / i) \* 2;

cout << (double)(x) << endl;

cout << (double)(y) << endl;//(Основные типы 3.2)

/\*Выведет 1.6

3

\*/

y = d \* (x = 2.5 / d);

cout << (double)(y) << endl;//(Основные типы 3.3)

//ВЫведет 2

x = d \* (y = ((int)2.9 + 1.1) / d);

cout << (double)(x) << endl;

cout << (double)(y) << endl;//(Основные типы 3.4/\*Выведет 2

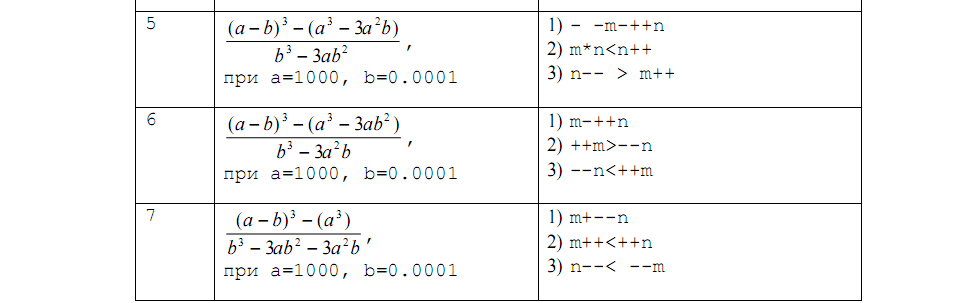
/\*Выведет 0

0

\*/

return 0;

}



Задание 1

#include <iostream>

#include <stdio.h>

#include <math.h>

using namespace std;

int main()

{

system("chcp>nul");

float a=1000, b=0.0001;

float c = pow(a - b, 3);

cout << c<<endl;

float d = pow(a, 3) - 3 \* pow(a, 2) \* b;

cout << d<<endl;

float e = pow(b, 3) - 3 \* pow(b, 2) \* a;

cout<<e<<endl;

float result = (c - d) / e;

cout << result;

system("pause>nul");

return 0;

}

Результат:

1e+09

1e+09

-3e-05

2.13333e+06

#include <iostream>

#include <stdio.h>

#include <math.h>

using namespace std;

int main()

{

system("chcp>nul");

double a=1000, b=0.0001;

double c = pow(a - b, 3);

cout << c<<endl;

double d = pow(a, 3) - 3 \* pow(a, 2) \* b;

cout << d<<endl;

double e = pow(b, 3) - 3 \* pow(b, 2) \* a;

cout<<e<<endl;

double result = (c - d) / e;

cout << result;

system("pause>nul");

return 0;

}

Результат:

1e+09

1e+09

-3e-05

-1.00136

Тип данных **float** занимает в памяти **4 байта**, содержит в себе числа с плавающей точкой, **до 7 знаков** после запятой.

Тип данных **double** занимает в памяти уже **8 байта** и также содержит в себе числа с плавающей точкой, но уже **до 15 знаков** после запятой.

Итого используя переменные типа **float** мы жертвуем точностью, получая взамен скорость и малый объем занимаемой памяти, **double** дает противоположный результат.

Задание 2

#include <iostream>

#include <cstdlib>

using namespace std;

int main() {

system("chcp 1251>nul");

int m, n;

cout << "Enter m and n:";

cin >> m >> n;

int first = m - ++n;

cout << "The first:\t" << first<<endl;

bool second = ++m > --n;

cout << "The second:\t" << second << endl;

bool third = --n < ++m;

cout << "The third:\t" << third;

system("pause>nul");

return 0;

}

Результат:

Enter m and n:6

6

The first: -1

The second: 1

The third: 1

first: к n прибавляем 1 и от m отнимаем n

second: если m+1>n-1 выводим 1, иначе 0

third: если n-1<m+1 выводим 1, иначе 0

**Вывод:** в ходе данной лабораторной работы я узнал новую информацию о функциях и их специфике, проработал принципы работы инкрементов и декрементов, а также получил новые данные о работе с логическими операторами.