ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ



МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ВЫСШАЯ ШКОЛА ПЕЧАТИ И МЕДИАИНДУСТРИИ

*Институт Принтмедиа и информационных технологий*

*Кафедра Информатики и информационных технологий*

направление подготовки

09.03.02 «Информационные системы и технологии»,

ЛАБОРАТОРНАЯ РАБОТА № 4

Дисциплина: ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

Тема: Функции

Выполнил(а): студент(ка) группы 181-722

Голиков Р.О.

(Фамилия И.О.)

Дата, подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *\_\_\_\_\_\_\_\_\_\_\_*

(Дата) (Подпись)

Проверил: Ктн Доцент Арсентьев Д.А.*\_\_\_\_\_\_\_\_\_\_*

(Фамилия И.О. степень, звание) (Оценка)

Дата, подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *\_\_\_\_\_\_\_\_\_\_\_*

(Дата) (Подпись)

Замечания: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва

2019

**Задание 1**

Напишите функцию с именем circarea(), которая вычисляет площадь круга. Функция должна принимать один аргумент типа float и возвращать значение типа float. Напишите функцию main(), которая просит пользователя ввести значение радиуса, вызывает функцию circarea(), а затем отображает результат вычисления на экране.

**Код:**

#include <iostream>

using namespace std;

float circarea(float radius);

int main()

{

setlocale(LC\_ALL, "");

double rad;

cout << "\nВведите радиус окружности:";

cin >> rad;

cout << "Площадь равна " << circarea(rad) << endl;

return 0;

}

float circarea(float r)

{

const float PI = 3.14159F;

return r \* r \*PI;

}

**Вывод:**



**Задание 2**

Возведение числа n в степень р — это умножение числа n на себя р раз. Напишите функцию с именем power(), которая в качестве аргументов принимает значение типа double для n и значение типа int для р и возвращает значение типа double. Для аргумента, соответствующего степени числа, задайте значение по умолчанию, равное 2, чтобы при отсутствии показателя степени при вызове функции число n возводилось в квадрат. Напишите функцию main(), которая запрашивает у пользователя ввод аргументов для функции power(), и отобразите на экране результаты ее работы.

**Код:**

#include <iostream>

using namespace std;

double power(double n, int p = 2);//p has default value 2

int main()

{

setlocale(LC\_ALL, "");

double number, answer;

int pow;

char yeserno;

cout << "\nВведите число:";//get number

cin >> number;

cout << "Будете вводить степень (y/n)?";

cin >> yeserno;

if (yeserno == 'y')//пользователю нужен не квадрат числа?

{

cout << "Введите степень:";

cin >> pow;

answer = power(number, pow);//возвести число в степень

}

else

answer = power(number);//квадрат числа

cout << "Ответ " << answer << endl;

return 0;

}

//power()

//возвращает число n, возведенное в степень p

double power(double n, int p)

{

double result = 1.0;//начать с 1

for (int j = 0; j < p; j++)//умножить на n

result \*= n;//p раз

return result;

}

**Вывод:**  


**Задание 3**

Напишите функцию с именем zeroSmaller(), в которую передаются с помощью ссылок два аргумента типа int, присваивающую меньшему из аргументов нулевое значение. Напишите программу для проверки работы функции.

**Код:**

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "");

void zeroSmaller(int&, int&);

int а = 4, б = 7, в = 11, г = 9;

zeroSmaller(а, б);

zeroSmaller(в, г);

cout << "\na=" << а << " б==" << б

<< " в==" << в << " г==" << г;

return 0;

}

void zeroSmaller(int&first, int&second)

{

if (first < second)

first = 0;

else

second = 0;

}

**Вывод:**  


**Задание 4**

Напишите функцию, принимающую в качестве аргументов два значения типа Distance и возвращающую значение наибольшего из аргументов.

**Код:**

#include <iostream>

using namespace std;

struct Distance //Английские расстояния

{

int feet;

float inches;

};

Distance bigengl(Distance, Distance);//объявления

void engldisp(Distance);

int main()

{

setlocale(LC\_ALL, "");

Distance d1, d2, d3; //определение трех расстояний

//получить рассто€ние d1 от пользователя

cout << "\nВведите число футов:"; cin >> d1.feet;

cout << "Введите число дюймов:"; cin >> d1.inches;

//получить рассто€ние d2 от пользователя

cout << "\nВведите число футов:"; cin >> d2.feet;

cout << "Введите число дюймов:"; cin >> d2.inches;

d3 = bigengl(d1, d2); //d3 больше, чем d1 и d2

//вывести все значения

cout << "\nd1="; engldisp(d1);

cout << "\nd2="; engldisp(d2);

cout << "\nнаибольшее Ч "; engldisp(d3); cout << endl;

return 0;

}

//bigengl()

//сравнивает две структуры типа Distance, возвращает

//наибольшую

Distance bigengl(Distance dd1, Distance dd2)

{

if (dd1.feet > dd2.feet) //если число футов различается,

return dd1; //вернуть то, которое больше

if (dd1.feet < dd2.feet)

return dd2;

if (dd1.inches > dd2.inches)//если число дюймов

// различается,

return dd1; //вернуть наибольшее

else //дюймы или dd2, если равные значения

return dd2;

}

//engldisp()

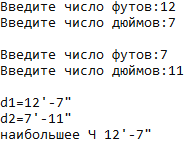
//выводит структуру типа Distance в футах и дюймах

void engldisp(Distance dd)

{

cout << dd.feet << "\'-" << dd.inches << "\"";

}

**Вывод:**  


**Задание 5**

Напишите функцию с именем hms\_to\_secs(), имеющую три аргумента типа int: часы, минуты и секунды. Функция должна возвращать эквивалент переданного ей временного значения в секундах (типа long). Создайте программу, которая будет циклически запрашивать у пользователя ввод значения часов, минут и секунд и выводить результат работы функции на экран.

**Код:**

#include <iostream>

using namespace std;

long hms\_to\_ses(int, int, int);

int main()

{

setlocale(LC\_ALL, "");

char ch;

int hours, minutes, seconds;

do

{

cout << "Введите время: ";

cin >> hours >> ch >> minutes >> ch >> seconds;

cout << "Всего секунд: " << hms\_to\_ses(hours, minutes, seconds);

cout << "\nХотите продолжить?(y/n) ";

cin >> ch;

} while (ch == 'y');

return 0;

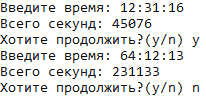
}

long hms\_to\_ses(int hours, int minutes, int seconds)

{

return hours \* 3600 + minutes \* 60 + seconds;

}

**Вывод:**  


**Задание 6**

Модифицируйте программу, описанную в упражнении 11 предыдущей лабораторной работы «Структуры», складывающую два структурных значения типа time. Теперь программа должна включать в себя две функции. Первая, time\_to\_secs(), принимает в качестве аргумента значение типа time и возвращает эквивалентное значение в секундах типа long. Вторая, secs\_to\_time(), в качестве аргумента принимает число секунд, имеющее тип long, а возвращает эквивалентное значение типа time.

**Код:**

#include <iostream>

using namespace std;

struct Time

{

int seconds;

int minutes;

int hours;

};

int time\_to\_secs(Time t)

{

return t.hours \* 3600 + t.minutes \* 60 + t.seconds;

}

Time time\_to\_secs(int totalSeconds)

{

Time t;

t.hours = totalSeconds / 3600;

int buf = totalSeconds % 3600;

t.minutes = buf / 60;

t.seconds = buf % 60;

return t;

}

int main()

{

setlocale(LC\_ALL, "");

Time t1, t2;

char ch;

cout << "Введите два время:\n";

cin >> t1.hours >> ch >> t1.minutes >> ch >> t1.seconds;

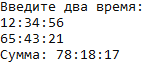
cin >> t2.hours >> ch >> t2.minutes >> ch >> t2.seconds;

Time resTime = time\_to\_secs(time\_to\_secs(t1) + time\_to\_secs(t2));

cout << "Сумма: " << resTime.hours << ch << resTime.minutes << ch << resTime.seconds;

return 0;

}

**Вывод:**  


**Задание 7**

Взяв в качестве основы функцию power() из упражнения 2, работающую только со значением типа double, создайте перегруженные функции с этим же именем, принимающими в качестве аргумента значения типа char, int, long и float. Напишите программу, вызывающую функцию power() со всеми возможными типами аргументов.

**Код:**

#include <iostream>

using namespace std;

double power(char n, int p = 2);

double power(int n, int p = 2);

double power(long n, int p = 2);

double power(float n, int p = 2);

double power(double n, int p = 2);

int main()

{

setlocale(LC\_ALL, "");

char a1;

int a2;

long a3;

float a4;

double a5;

double answer;

int pow;

cout << "Введите пять чисел char, int, long, float, double: ";

cin >> a1 >> a2 >> a3 >> a4 >> a5;

cout << "Введите степень: ";

cin >> pow;

cout << "Результаты: " << power(a1, pow) << " " << power(a2, pow) << " " <<

power(a3, pow) << " " << power(a4, pow) << " " << power(a5, pow);

return 0;

}

double power(char n, int p)

{

double result = 1.0;

for (int j = 0; j < p; j++)

result \*= n;

return result;

}

double power(int n, int p)

{

double result = 1.0;

for (int j = 0; j < p; j++)

result \*= n;

return result;

}

double power(long n, int p)

{

double result = 1.0;

for (int j = 0; j < p; j++)

result \*= n;

return result;

}

double power(float n, int p)

{

double result = 1.0;

for (int j = 0; j < p; j++)

result \*= n;

return result;

}

double power(double n, int p)

{

double result = 1.0;

for (int j = 0; j < p; j++)

result \*= n;

return result;

}

**Вывод:**  


**Задание 8**

Напишите функцию с именем swap(), обменивающую значениями два своих аргумента типа int (обратите внимание, что изменяться должны значения переменных из вызывающей программы, а не локальных переменных функции). Выберите способ передачи аргументов. Напишите вызывающую программу main(), использующую данную функцию.

**Код:**

#include <iostream>

using namespace std;

void swap(int &first, int &second)

{

int buf = first;

first = second;

second = buf;

}

int main()

{

setlocale(LC\_ALL, "");

cout << "Введите два числа: ";

int a, b;

cin >> a >> b;

swap(a, b);

cout << a << " " << b;

return 0;

}

**Вывод:**  


**Задание 9**

Переработайте программу из упражнения 8 так, чтобы функция swap() принимала в качестве аргументов значения типа time (см. упражнение 6).

**Код:**

#include <iostream>

using namespace std;

struct Time

{

int seconds;

int minutes;

int hours;

};

void swap(Time &first, Time &second)

{

Time buf = first;

first = second;

second = buf;

}

int main()

{

setlocale(LC\_ALL, "");

Time t1, t2;

char ch;

cout << "Введите два время:\n";

cin >> t1.hours >> ch >> t1.minutes >> ch >> t1.seconds;

cin >> t2.hours >> ch >> t2.minutes >> ch >> t2.seconds;

swap(t1, t2);

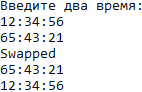
cout << "Swapped\n";

cout << t1.hours << ch << t1.minutes << ch << t1.seconds << endl;

cout << t2.hours << ch << t2.minutes << ch << t2.seconds << endl;

return 0;

}

**Вывод:**  


**Задание 10**

Напишите функцию, которая при каждом вызове будет выводить на экран количество раз, которое она вызывалась ранее. Напишите программу, которая будет вызывать данную функцию не менее 10 раз. Попробуйте реализовать данную функцию двумя различными способами: с использованием глобальной переменной и статической локальной переменной для хранения числа вызовов функции. Какой из способов предпочтительней? Почему для решения задачи нельзя использовать обычную локальную переменную?

**Код:**

#include <iostream>

using namespace std;

int Global;

void RecGlobal()

{

cout << "Глобальная переменная: " << Global++ << endl;

if (Global == 10) return;

RecGlobal();

}

// У обычной локальной переменной область видимости –

// только в функции, и с каждым вызовом обычная

// локальная переменная будет просто переопределяться

void RecLocalStatic()

{

static int local = 0;

cout << "Локальная статическая переменная: " << local++ << endl;

if (local == 10) return;

RecLocalStatic();

}

int main()

{

setlocale(LC\_ALL, "");

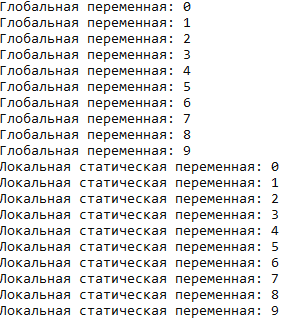
Global = 0;

RecGlobal();

RecLocalStatic();

return 0;

}

**Вывод:**  


**Задание 11**

Напишите программу, использующую структуру sterling, которая описана в упражнении 10 лабораторной работы 2 «Структуры». Программа должна получать от пользователя значения двух денежных сумм, выраженных в фунтах, шиллингах и пенсах, складывать эти значения и выводить результат на экран в том же формате. Необходимо разработать три функции. Первая из них должна получать от пользователя число фунтов, шиллингов и пенсов и возвращать соответствующее значение типа sterling. Вторая функция должна принимать в качестве аргументов два значения типа sterling, складывать их и возвращать значение, также имеющее тип sterling. Третья функция должна принимать аргумент типа sterling и выводить его значение на экран.

**Код:**

#include <iostream>

using namespace std;

struct Sterling

{

const int PoundsToOldShilling = 20;

const int OldShillingToOldPennies = 12;

int Pounds;

int Shillings;

int Pence;

Sterling(int pounds, int shillings, int pence)

{

Pence = pence % OldShillingToOldPennies;

Shillings = shillings + pence / OldShillingToOldPennies;

Pounds = pounds + Shillings / PoundsToOldShilling;

Shillings = Shillings % PoundsToOldShilling;

}

void Write()

{

cout << "Сумма старых фунтов: " << Pounds << '.' << Shillings << '.' << Pence;

}

};

Sterling Sum(Sterling s1, Sterling s2)

{

return Sterling(s1.Pounds + s2.Pounds, s1.Shillings + s2.Shillings, s1.Pence + s2.Pence);

}

int main()

{

setlocale(LC\_ALL, "");

int pounds1, pounds2, shillings1, shillings2, pence1, pence2;

char ch;

cout << "Введите два значения старых фунтов: \n";

cin >> pounds1 >> ch >> shillings1 >> ch >> pence1;

cin >> pounds2 >> ch >> shillings2 >> ch >> pence2;

Sterling s1 = Sterling(pounds1, shillings1, pence1);

Sterling s2 = Sterling(pounds2, shillings2, pence2);

Sterling sRes = Sum(s1, s2);

sRes.Write();

return 0;

}

**Вывод:**  


**Задание 12**

Модифицируйте калькулятор, созданный в упражнении 12 лабораторной работы 2 «Структуры», так, чтобы каждая арифметическая операция выполнялась с помощью функции. Функции могут называться fadd(), fsub(), fmul() и fdiv(). Каждая из функций должна принимать два структурных аргумента типа fraction и возвращать значение того же типа.

**Код:**

#include <iostream>

using namespace std;

struct fraction

{

int Dividend;

int Divider;

fraction(){}

fraction(int dividend, int divider)

{

dividend = dividend;

divider = divider;

}

};

fraction fsum(fraction f1, fraction f2)

{

fraction res;

res.Dividend = f1.Dividend \* f2.Divider + f1.Divider \* f2.Dividend;

res.Divider = f1.Divider \* f2.Divider;

return res;

}

fraction fsub(fraction f1, fraction f2)

{

fraction res;

res.Dividend = f1.Dividend \* f2.Divider - f1.Divider \* f2.Dividend;

res.Divider = f1.Divider \* f2.Divider;

return res;

}

fraction fmul(fraction f1, fraction f2)

{

fraction res;

res.Dividend = f1.Dividend \* f2.Dividend;

res.Divider = f1.Divider \* f2.Divider;

return res;

}

fraction fdiv(fraction f1, fraction f2)

{

fraction res;

res.Dividend = f1.Dividend \* f2.Divider;

res.Divider = f1.Divider \* f2.Dividend;

return res;

}

int main()

{

setlocale(LC\_ALL, "");

fraction f1, f2, res;

char oper, ch;

do {

cout << "\nВведите выражение с дробями: ";

cin >> f1.Dividend >> ch >> f1.Divider >> oper >> f2.Dividend >> ch >> f2.Divider;

switch (oper)

{

case '+':

res = fsum(f1, f2);

break;

case '-':

res = fsub(f1, f2);

break;

case '\*':

res = fmul(f1, f2);

break;

case '/':

res = fdiv(f1, f2);

break;

default: res.Dividend = res.Divider = 0;

}

cout << "Answer = " << res.Dividend << '/' << res.Divider;

cout << "\nDo another (Enter 'y' or 'n')? ";

cin >> ch;

} while (ch != 'n');

return 0;

}

**Вывод:**

