Учебный курс: Технологии программирования

Кафедра: ИС Факультет: ИТиП

Лабораторная работа №2. Ссылки

Выполнил: Трофимов В.А. Группа: 1511

Преподаватель: Повышев В.В.

# Задание

1. Объявите (в отельном заголовочном файле) и реализуйте (в другом файле) процедуры (они не возвращают значений!) согласно варианту.

Изменяет большую из двух переменных на её остаток от деления на вторую переменную.

Изменяют вещественную переменную на обратное к ней число.

Уменьшает радиус окружности на заданное число.

Меняет в матрице местами две указанные строчки.

2. Все процедуры должны быть написаны в двух вариантах – один вариант использует указатели, второй вариант – ссылки.

3. Напишите программу, проверяющую и демонстрирующую правильность работы процедур.

# Входные/Выходные данные

## Консоль

>>2 5

2 1

>>2 5

2 1

>>2

0.5

>>2

0.5

>>4

2

>>4

2

>>0 0 0 1 1 1 2 2 2

2.000000 2.000000 2.000000

1.000000 1.000000 1.000000

0.000000 0.000000 0.000000

2.000000 2.000000 0.000000

1.000000 1.000000 1.000000

0.000000 0.000000 0.000000

# Исходный текст

## Файл main.cpp

// Course: Programming technologies (C Plus Plus)

// Lab 2. References.

// Student: Trofimov V.A. Group: 1511

// Teacher: Povishev V.V.

// Created 21.02.2013 Modified: 22.02.2013

// Description: Program entry point.

#include "reference.h"

#include <iostream>

using namespace std;

int main(){

int \* a = (int\*)malloc(sizeof(int)); \*a = 2;

int \* b = (int\*)malloc(sizeof(int)); \*b = 5;

MaxModul(a, b);

cout << \*a << " " << \*b << endl;

//

int temp1, temp2;

int & ar = temp1 = 2;

int & br = temp2 = 5;

MaxModul(ar, br);

cout << ar << " " << br << endl;

//

double \* c = (double\*)malloc(sizeof(double)); \*c = 2;

Inverse(c);

cout << \*c << endl;

double temp3;

double & cr = temp3 = 2;

Inverse(cr);

cout << cr << endl;

//

circle \* d = (circle\*)malloc(sizeof(circle));

d -> x = 0; d -> y = 0; d -> r = 4; \*c = 2;

Reduce(d, c);

cout << d -> r << endl;

//

circle temp4; temp4.x = temp4.y = 0; temp4.r = 4;

circle & e = temp4; cr = 2;

Reduce(e, cr);

cout << e.r << endl;

//

mtrx \* f = (mtrx\*)malloc(sizeof(mtrx));

f->data[0][0] = 0; f->data[0][1] = 0; f->data[0][2] = 0;

f->data[1][0] = 1; f->data[1][1] = 1; f->data[1][2] = 1;

f->data[2][0] = 2; f->data[2][1] = 2; f->data[2][2] = 2;

\*a = 0; \*b = 2;

Swap(f, a, b);

print(f);

//

mtrx g;

g.data[0][0] = 0; g.data[0][1] = 0; g.data[0][2] = 0;

g.data[1][0] = 1; g.data[1][1] = 1; g.data[1][2] = 1;

g.data[2][0] = 2; g.data[2][1] = 2; g.data[2][3] = 2;

mtrx & gr = g; ar = 0; br = 2;

Swap(gr, ar, br);

print(gr);

return 0;

## }

## Файл reference.h

// Course: Programming technologies (C Plus Plus)

// Lab 2. References.

// Student: Trofimov V.A. Group: 1511

// Teacher: Povishev V.V.

// Created 21.02.2013 Modified: 22.02.2013

// Description: Reference header file.

#pragma once

class mtrx {

public: double data[3][3];

};

class circle {

public: double x, y, r;

};

void MaxModul(int \* a, int \* b);

void MaxModul(int & a, int & b);

void Inverse(double \* a);

void Inverse(double & a);

void Reduce(circle \* a, double \* b);

void Reduce(circle & a, double & b);

void Swap(mtrx \* a, int \* b, int \* c);

void Swap(mtrx & a, int & b, int & c);

void print(mtrx \* a);

void print(mtrx & a);

## Файл reference.h

// Course: Programming technologies (C Plus Plus)

// Lab 1. Namespaces.

// Student: Trofimov V.A. Group: 1511

// Teacher: Povishev V.V.

// Created 21.02.2013 Modified: 22.02.2013

// Description: Functions header file.

#pragma once

#include "declarations.h"

namespace matrix {

mtrx fill(mtrx & a, double value);

mtrx cpy(const mtrx & a);

mtrx mul(const mtrx & a, double b);

mtrx mul(const mtrx & a, const mtrx & b);

int print(const mtrx & a);

int read(mtrx & a);

}

## Файл functions.cpp

// Course: Programming technologies (C Plus Plus)

// Lab 2. References.

// Student: Trofimov V.A. Group: 1511

// Teacher: Povishev V.V.

// Created 21.02.2013 Modified: 22.02.2013

// Description: Reference source code file.

#include "reference.h"

#include <stdio.h>

void MaxModul(int \* a, int \* b){

\*a > \*b ? \*a %= \*b : \*b %= \*a;

}

void MaxModul(int & a, int & b){

a > b ? a %= b : b %= a;

}

void Inverse(double \* a){

\*a = 1 / \*a;

}

void Inverse(double & a){

a = 1 / a;

}

void Reduce(circle \* a, double \* b){

a -> r -= \*b;

a -> r = a -> r < 0 ? 0 : a -> r;

}

void Reduce(circle& a, double& b){

a.r -= b;

a.r = a.r < 0 ? 0 : a.r;

}

void Swap(mtrx \* a, int \* b, int \* c){

double temp;

for(int i = 0; i < 3; i++) {

temp = a -> data[\*b][i];

a -> data[\*b][i] = a -> data[\*c][i];

a -> data[\*c][i] = temp;

}

}

void Swap(mtrx & a, int & b, int & c){

double temp;

for(int i = 0; i < 3; i++) {

temp = a.data[b][i];

a.data[b][i] = a.data[c][i];

a.data[c][i] = temp;

}

}

void print(mtrx \* a) {

printf("\n%lf %lf %lf\n%lf %lf %lf\n%lf %lf %lf\n\n",

a->data[0][0], a->data[0][1], a->data[0][2],

a->data[1][0], a->data[1][1], a->data[1][2],

a->data[2][0], a->data[2][1], a->data[2][2]);

}

void print(mtrx & a) {

printf("\n%lf %lf %lf\n%lf %lf %lf\n%lf %lf %lf\n\n",

a.data[0][0], a.data[0][1], a.data[0][2],

a.data[1][0], a.data[1][1], a.data[1][2],

a.data[2][0], a.data[2][1], a.data[2][2]);

}