БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ Факультет прикладной математики и информатики Кафедра технологий программирования

Черепенников Роман Михайлович

Отчет по учебной практике

студента 1 курса 8 группы

Преподаватель

Комаровский Игорь Венедиктович старший преподаватель кафедры ДМиА

Оглавление

| 1. | Список выполненных лабораторных работ | 3 |
|----|---------------------------------------|----|
| 2. | Отчет по лабораторной работе № 8 | 9 |
| 3. | Отчет по лабораторной работе № 15 | 13 |
| 4. | Список использованных источников | 19 |

Список выполненных лабораторных работ

№1. Линейние алгоритмы (С++)

Вариант 15

Задание: Разработайте приложения для вычисления выражений. Независимые переменные – входные данные в приложении:

$$a = (1+y)^{\frac{1}{3}} \frac{x+y(x^2+4)}{e^{-x-2}+\frac{1}{2}(x^2+4)}; \qquad b = \frac{1+\cos(y-2)}{x^{\frac{4}{2}}+\sin^{\frac{4}{2}}z};$$

№2. Операции над целыми числами (С++)

Вариант 3

Определить, между какими двумя последующими степенями двойки расположено заданное число.

№3 Первое задание на ассемблере (Asm)

Вариант 32

По заданным целым $k \ge 0$ и $n \ge 0$ найти такое целое $x \ge 0$, что:

$$x^{k} + x^{2k} \le n$$
 W $(x+1)^{k} + (x+1)^{2k} > n$.

№4 Операции над целыми числами (Asm)

Вариант 3

Определить, между какими двумя последующими степенями двойки расположено заданное число.

№5 Полином (Asm)

Задана рекуррентная формула типа чисел фибоначчи f_n . Для заданных n>=0, целого x вычислить значение полиномов P1 и P2:

$$P1(n,x)=f_0x^n+f_1x^{n-1}+...+f_{n-1}x+f_n$$
,

$$P2(n,x)=f_nx^n+f_{n-1}x^{n-1}+...+f_1x+f_0.$$

№6 Указатели. Динамическое распределение памяти. С++

1. Реализовать два способа представления матрицы размера NxM в памяти компьютера. Первый — память под матрицу выделяется одним "куском" и матрица расположена построчно (либо по столбцам). Второй — вначале выделяется память под массив указателей на строки, а затем для каждой строки. Выбрать для себя какую-нибудь задачу на матрице и реализовать ее с помощью функции, в которую все необходимые данные передаются через параметры.

2.

3. Исследовать различные способы а) передачи по ссылке в качестве параметра статической двумерной (3-х мерной) матрицы, б) другие способы передачи в функцию многомерных матриц.

Примечание. В любых ситуациях, в формальном параметре, определяющем тип многомерного массива можно не указывать границу только по первой размерности

В частности, решить следующие ЗАДАЧИ:

а). Заголовок функции суммирования элементов 3-х мерного массива имеет вид:

int sum3 (int * pa, int n, int m, int l);

Пусть

int a[2][3][4];

3-х мерный массив. Как вызвать функцию sum3, чтобы она выполнила суммирование

элементов этого массива. Рассмотреть другие способы описания фактического параметра

для многомерного массива

б). Заголовок функции суммирования элементов 3-х мерного массива иммеет вид:

int sum3(int aa[cn][cm][cl]);

где cn, cm, cl - константы типа int. Пусть

int * p;

указатель на начало области расположения 3-х мерного массива int a[cn][cm][cl]; и константы cn, cm, cl доступны в момент вызова функции sum3. Как вызвать функцию sum3, используя указатель р для задания информации об области расположения массива а. в). Реализовать функцию MultMatrix умножения матриц целых чисел void MultMatrix (int * pa, int * pb, int * pc, int cn, int cm, int cl); Размерности матриц A, B, C соответственно сп на сm, сm на cl, сп на cl. Память для результирующей матрицы С уже выделена. 4. Пусть заданы 3-х мерные матрицы int a[cn][cm][cl]; int b[cn][cl][ck]; Вычислить матрицу int c[cn][cm][ck]; по формуле: для каждого і in [0..cn-1] c[i] = (a[i][*][*] умножить на b[i][*][*]Это вычисление реализовать в функции MultMatrix3

void MultMatrix3 (int * pa, int * pb, int * pc, int cn, int cm, int cl, int ck);

Память для результирующей матрицы С уже выделена.

№7 declspec(naked) Матрицы (Asm)

Вариант 1

Решить следующую задачу (на ассемблере) : в матрице $A_{m \times n}$ вычислить суммы строк верхнего треугольника

№8 Матрицы. Отдельный Asm-модуль (Asm)

Вариант 1

Реализовать мультимодульную программу (два или более модулей, по крайней мере один из них на ассемблере) для решения следующей задачи: в матрице $A_{m\times n}$ вычислить суммы строк верхнего треугольника

№9 Строковые команды (Asm)

Написать на ассемблере функцию (условно назовем ее FWords)

int FWords(char * s);

которая вычисляет количество слов в строке s. Для решения задачи необходимо использовать строковые команды ассемблера. Слова в строке отделяются друг от друга одним или более пробелов. Пробелы могут быть как в начале строки, так и в конце. Слово может содержать любые символы кроме пробела.

№10. Задание по простым структурам данных (стеки, очереди, деки и др.) (C++)

Вариант 4

Реализовать класс, представляющий дек. Класс для реализации дека использует динамический массив. В случае переполнения выделяется новая память, размер которой в два раза больше размера исходной памяти.

№11. Наследование. Иерархия геометрических фигур 1 (С++)

Вариант 14

Реализовать иерархию классов для плоских геометрических фигур, таких как выпуклый четырехугольник, выпуклый четырехугольник с диагоналями, круг, кольцо, точка (точечный объект) и т.д.

Класс этой иерархии должен поддерживать следующие методы: конструктор[ы], деструктор, MoveTo (для размещения фигуры по заданным координатам), Show (показать фигуру), Hide (спрятать фигуру), Rotate (повернуть на заданный угол вокруг центра тяжести), Explode (пропорционально сжать или растянуть

относительно центра тяжести). Кроме того, класс может содержать некоторые другие методы, используемые в реализации вышеупомянутых методов.

Данные, характеризующие фигуру, можно разделить на два вида:

- описывающие фигуру.
- определяющие расположение фигуры на экране. Расположение любой плоской фигуры можно задать координатами центра тяжести фигуры и углом какого-либо отрезка фигуры с осью Ох.

№12 Задачи по WinAPI 1 (C++)

Вариант 10

Управление перемещением кнопки с помощью клавиатуры.

№13 Задачи по WinAPI 2 (C++)

Вариант 16

Построить приложение, демонстрирующее процесс вычисления определенного интеграла функции y=f(x) на заданном промежутке [a,b]. Предоставить возможность: (a) выбора функции из меню; (б) ввода выражения, задающего функцию, из текстового файла (например, x*sin(x)).

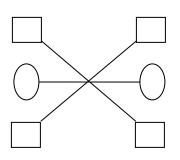
№14 Задачи по WinAPI 3 (C++)

Вариант 14

Разработка приложения с графическим интерфейсом

Приложение включает **меню**, по командам из которого вызывается **пользовательский диалог** и **стандартный диалог открытия файла** для определения условий выполнения задачи.

В диалоге в текстовых полях ввода задаются размеры задаются размеры рисуемой фигуры (большого квадрата и маленьких прямоугольников).



Выбранные размеры отображаются в строке статуса. Нажатием на кнопку в диалоге вызывается стандартный диалог выбора цвета. Три кнопки-флажка определяют второй альтернативный способ выбора цвета комбинацией красный-синий-зеленый. Группа из двух радиокнопок определяет способ получения размеров: считыванием из файла или вручную.

В главном окне рисуем фигуру вида (в основе – квадрат с различным числом окружностей по периметру)

По таймеру вершины фигуры закрашиваются выбранным цветом, при этом размеры элементов вначале увеличиваются в два раза а затем уменьшается до исходного, и цвет возвращается к исходному. Вершины закрашиваются поочередно по контуру фигуры.

Предоставить возможность запускать и останавливать вращение фигуры.

№15 Наследование. Иерархия геометрических фигур 2 (С++)

Вариант 14

Реализовать иерархию классов для плоских геометрических фигур, таких как выпуклый четырехугольник, выпуклый четырехугольник с диагоналями, круг, кольцо, точка (точечный объект) и т.д.

Класс этой иерархии должен поддерживать следующие методы: конструктор[ы], деструктор, MoveTo (для размещения фигуры по заданным координатам), Show (показать фигуру), Hide (спрятать фигуру), Rotate (повернуть на заданный угол вокруг центра тяжести), Explode (пропорционально сжать или растянуть относительно центра тяжести). Кроме того, класс может содержать некоторые другие методы, используемые в реализации вышеупомянутых методов.

Данные, характеризующие фигуру, можно разделить на два вида:

- описывающие фигуру.
- определяющие расположение фигуры на экране. Расположение любой плоской фигуры можно задать координатами центра тяжести фигуры и углом какого-либо отрезка фигуры с осью Ох

Отчет по лабораторной работе №8

1) Постановка задачи

Вариант 1

Реализовать мультимодульную программу (два или более модулей, по крайней мере один из них на ассемблере) для решения следующей задачи: в матрице $A_{m\times n}$ вычислить суммы строк верхнего треугольника.

2) Описание решения

Алгоритм

- 1. Считывание данных о размере матрицы и ее элементах, введенных пользователем
- 2. Представление матрицы в памяти компьютера как одномерного массива
- 3. Вычисление суммы верхнего треугольника с использованием соответствующей функции
- 4. Вывод результата на экран

Код некоторых фрагментов программы

Ввод данных от пользователя и отображение матрицы, введенной пользователем, а затем вызов функции. В функцию передается указатель на массив, количество строк и столбоцов:

```
□int main()
    setlocale(0, "rus");
    cout << "Введите размеры матрицы n,m: ";
    int n, m;
    cin >> n >> m;
    int* array = new int[n * m];
    int array size = n * m;
    for (int i = 0; i < n * m; ++i) { ... }
    cout << "=======" << endl;
    for (int i = 0; i < n * m; ++i)
       cout << array[i] << "     ";</pre>
       if ((i + 1) \% m == 0)
           cout << endl;</pre>
    cout << "=========;
    int x = AsmModule(array, n, m);
    cout << endl << "Результат: " << х;
```

Реализация функции выполняющей вычисления:

```
.386
    .model flat
    .data
    .code
   PUBLIC _AsmModule
_AsmModule PROC
    push ebp
 8 mov ebp,esp
10 push ebx
            push esi
            push edi
14
            mov ebx,[ebp+16]
                                ;в еbх количество столбцов
            imul ebx,[ebp+12]
            cmp ebx, 1
            je end1
            cmp ebx,0
            je end1
            xor eax,eax
                                  ;переменная для результата
            xor esi,esi
                                  ;адресная переменная
            mov edx,1
                                  ;номер строки
            mov ebx,[ebp+8]
            while_begin:
            cmp edx,[ebp+12]
            jg while_end
                                  ;мы прошли по всем строкам
            add esi,edx
                                  ;esi на первом в строке элементе который нужно суммировать
            mov ecx,[ebp+16]
            sub ecx,edx
            cmp ecx,0
            jle while_end
                                ;если в последующих строках не будет элементов для суммирования
            sumir:
            add eax,[ebx][esi*4]
            inc esi
            loop sumir
            add edx,1
            jmp while_begin
            end1:
            mov eax,0
40
            while_end:
                 pop edi
                 pop esi
                 pop ebx
                 mov esp,ebp
                 pop ebp
   ret
   _AsmModule endp
```

3) Результаты

50 end

Входные данные:

```
1 8 15
1)
    6 17 9
    0
       7
          23
           12 18 42 18
    1
        2
2)
    47
        54
           36
               3
                  10
                      4
        95
               7 12
                     7
    11
           84
    11
        99
3)
    0
        4
```

Тесты:

1)

2)

```
Консоль отладки Microsoft Visual Studio
Введите размеры матрицы n,m: 3 6
1 строка: 1 2 12 18 42 18
2 строка: 47 54 36 3 10 4
3 строка: 11 95 84 7 12 7
            12
                    18
                           42
                                   18
      2
       54
                      3
47
              36
                            10
                                   4
11
       95
              84
                      7
                            12
                                    7
Результат: 171
```

3)

Консоль отладки Microsoft Visual Studio

Отчет по лабораторной работе №15

1)Постановка задачи

Вариант 14

Реализовать иерархию классов для плоских геометрических фигур, таких как выпуклый четырехугольник, выпуклый четырехугольник с диагоналями, круг, кольцо, точка (точечный объект) и т.д.

Класс этой иерархии должен поддерживать следующие методы: конструктор[ы], деструктор, MoveTo (для размещения фигуры по заданным координатам), Show (показать фигуру), Hide (спрятать фигуру), Rotate (повернуть на заданный угол вокруг центра тяжести), Explode (пропорционально сжать или растянуть относительно центра тяжести). Кроме того, класс может содержать некоторые другие методы, используемые в реализации вышеупомянутых методов.

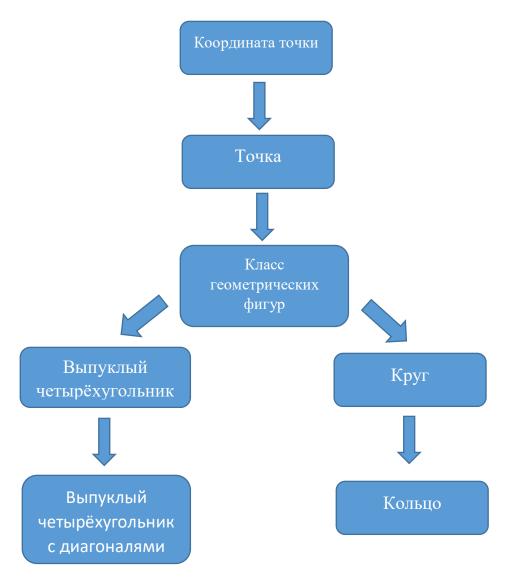
Данные, характеризующие фигуру, можно разделить на два вида:

- описывающие фигуру.
- определяющие расположение фигуры на экране. Расположение любой плоской фигуры можно задать координатами центра тяжести фигуры и углом какого-либо отрезка фигуры с осью Ох

2)Описание решения

Алгоритм

1. Реализация следующей полиморфной иерархии с использованием виртуальных функций:



- 2. Реализация методов Draw, Shift, MoveTo, Rotate, Explode, Hide, Show. Эти методы позволяют графически представить объекты.
- 3. Создание оконного приложения, позволяющего создавать и изменять объекты следующих классов: выпуклый четырехугольник, выпуклый четырехугольник с диагоналями, круг, кольцо.

Код некоторых фрагментов программы

Базовый класс полиморфной иерархии (KPoint):

Класс для представления выпуклого четырехугольника(KQuad):

Класс для представления кольца (KRing):

```
class KRing :public KCircle

| KRing(int _x, int _y, int _radius,int _subradius, bool _vis = true, COLORREF color=RGB(0,0,0),
| int _angle =0, double _coef = 1.0) :KCircle(_x,_y,_radius,_vis,color,_angle,_coef), subradius(_subradius) {}
| virtual ~KRing() { }
| protected:
| virtual void Draw(HDC& hdc) override;
| private:
| int subradius;
| int subradius;
| int subradius;
```

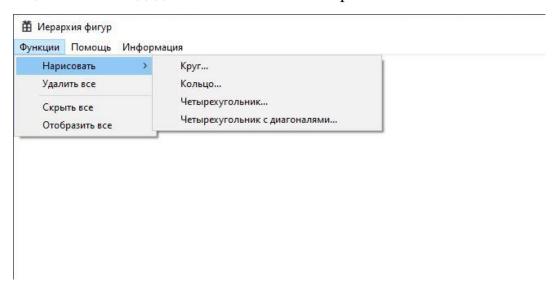
Реализация метода Draw для круга:

Для хранения множества фигур, созданных пользователем, используется контейнер vector указателей на базовый класс KPoint

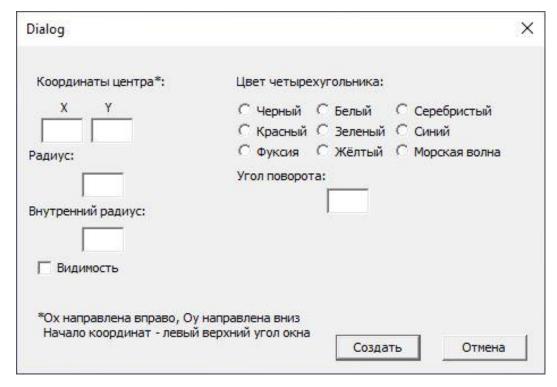
std::vector <KPoint *> shapes_vec;

3)Результаты

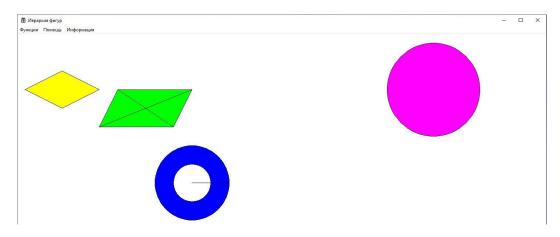
С помощью меню создадим объекты всех четырех классов:



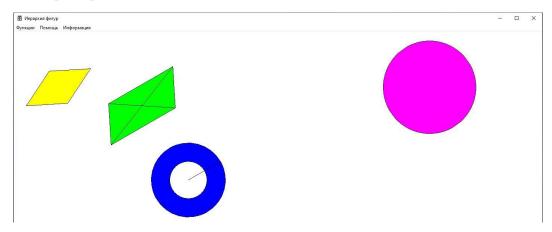
Окно создания кольца:



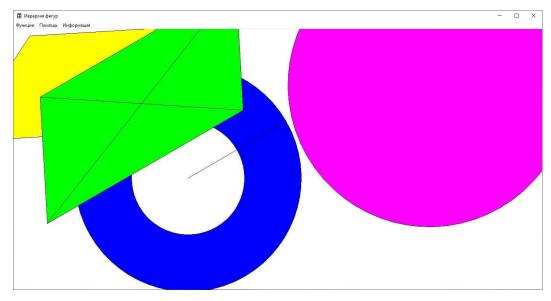
После их добавления будет изображено следующее:



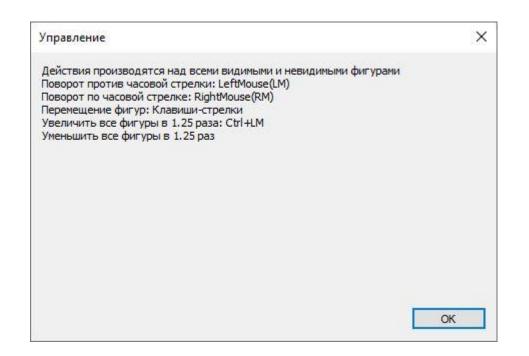
Вращение фигур:



Масштабирование и наложение фигур (наверху отображается последняя добавленная пользователем фигура):



Пользователю доступны следующие возможности:



Список использованных источников

- 1. Мейерс, Скотт Эффективный и современный C++. 42 рекомендации по использованию C++11 и C++14 / Скотт Мейерс. М.: Вильямс, 2015. 304
- 2. Щупак Ю.А. Win32 API. Эффективная разработка приложений. СПб.: Питер, 2007-572с.
- 3. Юань Фень. Программирование графики для Windows(+CD). СПб.: Питер, 2002-1072с.
- 4. Чарльз Петзольд. Программирование для Windows 95, том 1—СПб.: ВНV-Санкт-Петербург, 1997-495с.
- 5. Головиц Я. С 17 STL. Стандартная библиотека шаблонов. -СПб., 2018.