**Федеральное государственное автономное образовательное учреждение высшего образования**

**"Московский Авиационный Институт (Национальный Исследовательский Университет"**

Институт “Информационные технологии и прикладная математика”

**Лабораторная работа № 3**

по курсу «ООП»

Тема: Механизмы наследования С++

Студент: Николаева Елизавета

Группа: 80-201

Преподаватель: Чернышов Л.Н.

Дата: 15.01.2021

Оценка:

Москва, 2021

1. Постановка задачи

Вариант 5

Ромб, 5-угольник, 6-угольник

Разработать классы: вычисления геометрического центра, вывода координат вершин фигуры, вычисления площади фигуры. Классы должны наследоваться от базового класса Figure.

Программа должна иметь следующие возможности:

1) вводить фигуры, согласно моему варианту

2) сохранять созданные фигуры в динамический массив

3) вызывать для всего массива общие функции, т.е. распечатывать для каждой фигуры в массиве геометрический центр, координаты вершин и площадь

4) вычислять общую площадь фигур в массиве

5) удалять из массива фигуру по индексу

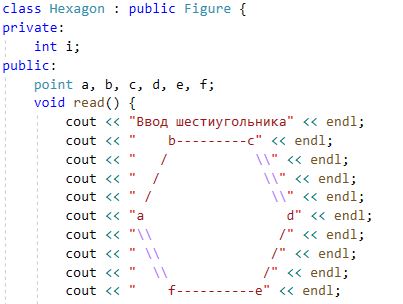
1. Решение

Репозиторий: [https://github.com/esnikolaeva/oop/tree/main/oop\_exercise\_03](https://github.com/esnikolaeva/oop/tree/main/oop_exercise_032)

Для решения задачи необходимо создать класс Figure

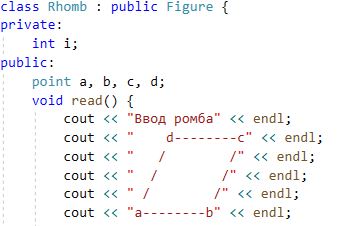
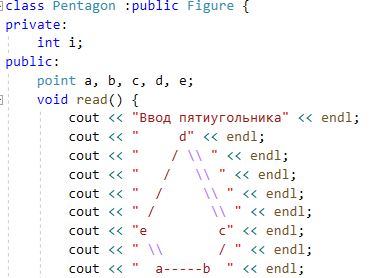
из которого в дальнейшем следуют классы шестиугольника, пятиугольника и ромба.

Класс шестиугольника содержит в себе ввод его вершин, проверку правильности их ввода, вычисление площади, центра координат и вывод фигуры:



С целью экономии листов в отчёте, в дальнейшем прикрепляю скриншоты с созданием классов. Полное их описание можно увидеть в листинге программы!

Аналогично создаются классы ромба и пятиугольника соответственно.



Далее в программе идет проверка на непустоту ввода и реализация меню.

Затем, создадим файл CMakeLists:

cmake\_minimum\_required(VERSION 3.20)

project(oop\_exercise\_03)

set(CMAKE\_CXX\_STANDARD 23)

add\_executable(oop\_exercise\_03 main.cpp)

1. Тестирование

**Тест 01:**

1. Ввести фигуру

2. Вывести координаты точек фигуры

3. Вывести площадь фигуры

4. Вывести центр фигуры

5. Вывести общую площадь

6. Удалить фигуру из массива

0. Выход

1

Введите фигуру:

1. Шестиугольник

2. Ромб

3. Пятиугольник

1

Ввод шестиугольника

b---------c

/ \

/ \

/ \

a d

\ /

\ /

\ /

f----------e

Введите координаты точки a

0

8

Введите координаты точки b

4

15

Введите координаты точки c

12

15

Введите координаты точки d

16

8

Введите координаты точки e

12

1

Введите координаты точки f

4

1

1. Ввести фигуру

2. Вывести координаты точек фигуры

3. Вывести площадь фигуры

4. Вывести центр фигуры

5. Вывести общую площадь

6. Удалить фигуру из массива

0. Выход

2

Фигура 0

Вывод фигуры шестиугольник вида:

b---------c

/ \

/ \

/ \

a d

\ /

\ /

\ /

f----------e

Координаты точки a: 0 8

Координаты точки b: 4 15

Координаты точки c: 12 15

Координаты точки d: 16 8

Координаты точки e: 12 1

Координаты точки f: 4 1

1. Ввести фигуру

2. Вывести координаты точек фигуры

3. Вывести площадь фигуры

4. Вывести центр фигуры

5. Вывести общую площадь

6. Удалить фигуру из массива

0. Выход

3

Фигура 0

Площадь шестиугольника равна 20.7846

1. Ввести фигуру

2. Вывести координаты точек фигуры

3. Вывести площадь фигуры

4. Вывести центр фигуры

5. Вывести общую площадь

6. Удалить фигуру из массива

0. Выход

4

Фигура 0

Координаты центра шестиугольника: 4 7

1. Ввести фигуру

2. Вывести координаты точек фигуры

3. Вывести площадь фигуры

4. Вывести центр фигуры

5. Вывести общую площадь

6. Удалить фигуру из массива

0. Выход

5

Общая площадь равна 20.7846

1. Ввести фигуру

2. Вывести координаты точек фигуры

3. Вывести площадь фигуры

4. Вывести центр фигуры

5. Вывести общую площадь

6. Удалить фигуру из массива

0. Выход

6

Введите номер удаляемого элемента

2

Данного элемента не существует

1. Ввести фигуру

2. Вывести координаты точек фигуры

3. Вывести площадь фигуры

4. Вывести центр фигуры

5. Вывести общую площадь

6. Удалить фигуру из массива

0. Выход

0

**Тест 02:**

1. Ввести фигуру

2. Вывести координаты точек фигуры

3. Вывести площадь фигуры

4. Вывести центр фигуры

5. Вывести общую площадь

6. Удалить фигуру из массива

0. Выход

1

Введите фигуру:

1. Шестиугольник

2. Ромб

3. Пятиугольник

2

Ввод ромба

d--------c

/ /

/ /

/ /

a--------b

Введите координаты точки a

-2

-1

Введите координаты точки b

0

-1

Введите координаты точки c

2

1

Введите координаты точки d

0

1

1. Ввести фигуру

2. Вывести координаты точек фигуры

3. Вывести площадь фигуры

4. Вывести центр фигуры

5. Вывести общую площадь

6. Удалить фигуру из массива

0. Выход

2

Фигура 0

Вывод фигуры ромб вида:

d--------c

/ /

/ /

/ /

a--------b

Координаты точки a: -2 -1

Координаты точки b: 0 -1

Координаты точки c: 2 1

Координаты точки d: 0 1

1. Ввести фигуру

2. Вывести координаты точек фигуры

3. Вывести площадь фигуры

4. Вывести центр фигуры

5. Вывести общую площадь

6. Удалить фигуру из массива

0. Выход

3

Фигура 0

Площадь ромба равна 4.47214

1. Ввести фигуру

2. Вывести координаты точек фигуры

3. Вывести площадь фигуры

4. Вывести центр фигуры

5. Вывести общую площадь

6. Удалить фигуру из массива

0. Выход

4

Фигура 0

Координаты центра ромба: 0 0

1. Ввести фигуру

2. Вывести координаты точек фигуры

3. Вывести площадь фигуры

4. Вывести центр фигуры

5. Вывести общую площадь

6. Удалить фигуру из массива

0. Выход

5

Общая площадь равна 4.47214

1. Ввести фигуру

2. Вывести координаты точек фигуры

3. Вывести площадь фигуры

4. Вывести центр фигуры

5. Вывести общую площадь

6. Удалить фигуру из массива

0. Выход

6

Введите номер удаляемого элемента

0

Данного элемента не существует

1. Ввести фигуру

2. Вывести координаты точек фигуры

3. Вывести площадь фигуры

4. Вывести центр фигуры

5. Вывести общую площадь

6. Удалить фигуру из массива

0. Выход

0

**Тест 03:**

1. Ввести фигуру

2. Вывести координаты точек фигуры

3. Вывести площадь фигуры

4. Вывести центр фигуры

5. Вывести общую площадь

6. Удалить фигуру из массива

0. Выход

1

Введите фигуру:

1. Шестиугольник

2. Ромб

3. Пятиугольник

3

Ввод пятиугольника

d

/ \

/ \

/ \

/ \

e c

\ /

a-----b

Введите координаты точки a

-1

-2

Введите координаты точки b

1

-2

Введите координаты точки c

2

0

Введите координаты точки d

0

2

Введите координаты точки e

-2

0

1. Ввести фигуру

2. Вывести координаты точек фигуры

3. Вывести площадь фигуры

4. Вывести центр фигуры

5. Вывести общую площадь

6. Удалить фигуру из массива

0. Выход

2

Фигура 0

Вывод фигуры пятиугольник вида:

d

/ \

/ \

/ \

/ \

e c

\ /

a-----b

Координаты точки a: -1 -2

Координаты точки b: 1 -2

Координаты точки c: 2 0

Координаты точки d: 0 2

Координаты точки e: -2 0

1. Ввести фигуру

2. Вывести координаты точек фигуры

3. Вывести площадь фигуры

4. Вывести центр фигуры

5. Вывести общую площадь

6. Удалить фигуру из массива

0. Выход

3

Фигура 0

Площадь пятиугольника равна 10

1. Ввести фигуру

2. Вывести координаты точек фигуры

3. Вывести площадь фигуры

4. Вывести центр фигуры

5. Вывести общую площадь

6. Удалить фигуру из массива

0. Выход

4

Фигура 0

Координаты центра пятиугольника: 0 -0.4

1. Ввести фигуру

2. Вывести координаты точек фигуры

3. Вывести площадь фигуры

4. Вывести центр фигуры

5. Вывести общую площадь

6. Удалить фигуру из массива

0. Выход

5

Общая площадь равна 10

1. Ввести фигуру

2. Вывести координаты точек фигуры

3. Вывести площадь фигуры

4. Вывести центр фигуры

5. Вывести общую площадь

6. Удалить фигуру из массива

0. Выход

6

Введите номер удаляемого элемента

1

1. Ввести фигуру

2. Вывести координаты точек фигуры

3. Вывести площадь фигуры

4. Вывести центр фигуры

5. Вывести общую площадь

6. Удалить фигуру из массива

0. Выход

0

4. Листинг

/\*Николаева Е.С. М8О-201Б-20

Вариант 5: Ромб, 5-угольник, 6-угольник\*/

#include <iostream>

#include <vector>

#include <cmath>

using namespace std;

class Figure {

public:

struct point {

double x, y;

};

double S;

point center;

point read\_point() {

point p;

while (true) {

cin >> p.x >> p.y;

if (cin.fail()) {

cin.clear();

cin.ignore(32767, '\n');

cout << "Координаты введены неверно. Попробуйте снова" << endl;

}

else {

cin.ignore(32767, '\n');

return p;

}

}

}

virtual void read() = 0;

virtual double square() = 0;

virtual void print\_square() = 0;

virtual void coordinates\_center() = 0;

virtual void print() = 0;

};

class Hexagon : public Figure {

private:

int i;

public:

point a, b, c, d, e, f;

void read() {

cout << "Ввод шестиугольника" << endl;

cout << " b---------c" << endl;

cout << " / \\" << endl;

cout << " / \\" << endl;

cout << " / \\" << endl;

cout << "a d" << endl;

cout << "\\ /" << endl;

cout << " \\ /" << endl;

cout << " \\ /" << endl;

cout << " f----------e" << endl;

while (true) {

cout << "Введите координаты точки a" << endl;

a = read\_point();

cout << "Введите координаты точки b" << endl;

b = read\_point();

cout << "Введите координаты точки c" << endl;

c = read\_point();

cout << "Введите координаты точки d" << endl;

d = read\_point();

cout << "Введите координаты точки e" << endl;

e = read\_point();

cout << "Введите координаты точки f" << endl;

f = read\_point();

if (!((b.x < c.x) && (b.y == c.y) && (c.x < d.x) && (c.y > d.y) && (d.x > e.x) && (d.y > e.y) && (e.x > f.x) && (e.y == f.y) && (f.x > a.x) && (f.y < a.y) && (a.x < b.x) && (a.y < b.y))) {

cout << "Координаты введены неверно." << endl;

while (true) {

cout << "Попробовать снова?" << endl << "1. Да" << endl << "2. Нет" << endl;

cin >> i;

if (cin.fail()) {

cin.clear();

cin.ignore(32767, '\n');

cout << "Неверный ввод" << endl;

}

else {

cin.ignore(32767, '\n');

break;

}

}

if (i == 2) return;

}

else

return;

}

}

double square() {

S = (6 \* (e.x - f.x) \* sqrt(3)) / 4;

return S;

}

void print\_square() {

cout << "Площадь шестиугольника равна " << square() << endl;

return;

}

void coordinates\_center() {

center.x = (e.x - f.x) / 2;

center.y = (b.y - f.y) / 2;

cout << "Координаты центра шестиугольника: " << center.x << " " << center.y << endl;

return;

}

void print() {

cout << "Вывод фигуры шестиугольник вида:" << endl;

cout << " b---------c" << endl;

cout << " / \\" << endl;

cout << " / \\" << endl;

cout << " / \\" << endl;

cout << "a d" << endl;

cout << "\\ /" << endl;

cout << " \\ /" << endl;

cout << " \\ /" << endl;

cout << " f----------e" << endl;

cout << "Координаты точки a: " << a.x << " " << a.y << endl;

cout << "Координаты точки b: " << b.x << " " << b.y << endl;

cout << "Координаты точки c: " << c.x << " " << c.y << endl;

cout << "Координаты точки d: " << d.x << " " << d.y << endl;

cout << "Координаты точки e: " << e.x << " " << e.y << endl;

cout << "Координаты точки f: " << f.x << " " << f.y << endl;

return;

}

};

class Rhomb : public Figure {

private:

int i;

public:

point a, b, c, d;

void read() {

cout << "Ввод ромба" << endl;

cout << " d--------c" << endl;

cout << " / /" << endl;

cout << " / /" << endl;

cout << " / /" << endl;

cout << "a--------b" << endl;

while (true) {

cout << "Введите координаты точки a" << endl;

a = read\_point();

cout << "Введите координаты точки b" << endl;

b = read\_point();

cout << "Введите координаты точки c" << endl;

c = read\_point();

cout << "Введите координаты точки d" << endl;

d = read\_point();

if (!((a.x < b.x) && (d.x < c.x) && (d.y == c.y) && (a.y == b.y) && (c.y > a.y) && ((b.x - a.x) == (c.x - d.x) == (sqrt((d.y - a.y) \* (d.y - a.y) + (d.x - a.x) \* (d.x - a.x))) == (sqrt((d.y - a.y) \* (d.y - a.y) + (c.x - b.x) \* (c.x - b.x)))))) {

cout << "Координаты введены неверно." << endl;

while (true) {

cout << "Попробовать снова?" << endl << "1. Да" << endl << "2. Нет" << endl;

cin >> i;

if (cin.fail()) {

cin.clear();

cin.ignore(32767, '\n');

cout << "Неверный ввод" << endl;

}

else {

cin.ignore(32767, '\n');

break;

}

}

if (i == 2) return;

}

else

return;

}

}

double square() {

S = (sqrt((d.y - b.y) \* (d.y - b.y) + (b.x - d.x) \* (b.x - d.x))) \* (sqrt((d.y - b.y) \* (d.y - b.y) + (c.x - a.x) \* (c.x - a.x))) / 2;

return S;

}

void print\_square() {

cout << "Площадь ромба равна " << square() << endl;

return;

}

void coordinates\_center() {

center.x = (a.x + b.x + c.x + d.x) / 4;

center.y = (a.y + b.y + c.y + d.y) / 4;

cout << "Координаты центра ромба: " << center.x << " " << center.y << endl;

return;

}

void print() {

cout << "Вывод фигуры ромб вида:" << endl;

cout << " d--------c" << endl;

cout << " / /" << endl;

cout << " / /" << endl;

cout << " / /" << endl;

cout << "a--------b" << endl;

cout << "Координаты точки a: " << a.x << " " << a.y << endl;

cout << "Координаты точки b: " << b.x << " " << b.y << endl;

cout << "Координаты точки c: " << c.x << " " << c.y << endl;

cout << "Координаты точки d: " << d.x << " " << d.y << endl;

return;

}

};

class Pentagon :public Figure {

private:

int i;

public:

point a, b, c, d, e;

void read() {

cout << "Ввод пятиугольника" << endl;

cout << " d" << endl;

cout << " / \\ " << endl;

cout << " / \\ " << endl;

cout << " / \\ " << endl;

cout << " / \\ " << endl;

cout << "e c" << endl;

cout << " \\ / " << endl;

cout << " a-----b " << endl;

while (true) {

cout << "Введите координаты точки a" << endl;

a = read\_point();

cout << "Введите координаты точки b" << endl;

b = read\_point();

cout << "Введите координаты точки c" << endl;

c = read\_point();

cout << "Введите координаты точки d" << endl;

d = read\_point();

cout << "Введите координаты точки e" << endl;

e = read\_point();

if (!((a.y == b.y) && (e.x < a.x) && (e.x < b.x) && (e.x < c.x) && (e.x < d.x) && (a.x - b.x) && (c.x > a.x) && (c.x > b.x) && (c.x > e.x) && (c.x > d.x) && (c.y > b.y) && (d.y > c.y) && (d.y > e.y) && (e.y > a.y))) {

cout << "Координаты введены неверно." << endl;

while (true) {

cout << "Попробовать снова?" << endl << "1. Да" << endl << "2. Нет" << endl;

cin >> i;

if (cin.fail()) {

cin.clear();

cin.ignore(32767, '\n');

cout << "Неверный ввод" << endl;

}

else {

cin.ignore(32767, '\n');

break;

}

}

if (i == 2) return;

}

else

return;

}

}

double square() {

S = (c.x - e.x) \* (d.y - a.y) - (d.x - e.x) \* (d.y - e.y) / 2 - (c.x - d.x) \* (d.y - c.y) / 2 - (c.x - b.x) \* (c.y - b.y) / 2 - (a.x - e.x) \* (e.y - a.y) / 2;

return S;

}

void print\_square() {

cout << "Площадь пятиугольника равна " << square() << endl;

return;

}

void coordinates\_center() {

center.x = (a.x + b.x + c.x + d.x + e.x) / 5;

center.y = (a.y + b.y + c.y + d.y + e.y) / 5;

cout << "Координаты центра пятиугольника: " << center.x << " " << center.y << endl;

return;

}

void print() {

cout << "Вывод фигуры пятиугольник вида:" << endl;

cout << " d" << endl;

cout << " / \\ " << endl;

cout << " / \\ " << endl;

cout << " / \\ " << endl;

cout << " / \\ " << endl;

cout << "e c" << endl;

cout << " \\ / " << endl;

cout << " a-----b " << endl;

cout << "Координаты точки a: " << a.x << " " << a.y << endl;

cout << "Координаты точки b: " << b.x << " " << b.y << endl;

cout << "Координаты точки c: " << c.x << " " << c.y << endl;

cout << "Координаты точки d: " << d.x << " " << d.y << endl;

cout << "Координаты точки e: " << e.x << " " << e.y << endl;

return;

}

};

int prov() {

if (cin.fail()) {

cin.clear();

cin.ignore(32767, '\n');

cout << "Неверный ввод" << endl;

return 0;

}

else {

cin.ignore(32767, '\n');

return 1;

}

}

int main() {

setlocale(LC\_ALL, "Russian");

int k, n;

double Si, sum = 0;

vector <Figure\*> fig;

while (true) {

cout << "1. Ввести фигуру" << endl << "2. Вывести кооординаты точек фигуры" << endl << "3. Вывести площадь фигуры" << endl << "4. Вывести центр фигуры" << endl << "5. Вывести общую площадь" << endl << "6. Удалить фигуру из массива" << endl << "0. Выход" << endl;

cin >> k;

if (prov()) {

if (k == 1) {

cout << "Введите фигуру:" << endl << "1. Шестиугольник" << endl << "2. Ромб" << endl << "3. Пятиугольник" << endl;

cin >> n;

if (prov()) {

if (n == 1) {

Hexagon\* f = new Hexagon();

f->read();

fig.push\_back(f);

}

if (n == 2) {

Rhomb\* f = new Rhomb();

f->read();

fig.push\_back(f);

}

if (n == 3) {

Pentagon\* f = new Pentagon();

f->read();

fig.push\_back(f);

}

}

}

if (k == 2) {

for (int i = 0; i < fig.size(); i++) {

cout << "Фигура " << i << endl;

fig[i]->print();

}

}

if (k == 3) {

for (int i = 0; i < fig.size(); i++) {

cout << "Фигура " << i << endl;

fig[i]->print\_square();

}

}

if (k == 4) {

for (int i = 0; i < fig.size(); i++) {

cout << "Фигура " << i << endl;

fig[i]->coordinates\_center();

}

}

if (k == 5) {

for (int i = 0; i < fig.size(); i++)

sum += fig[i]->square();

cout << "Общая площадь равна " << sum << endl;

}

if (k == 6) {

cout << "Введите номер удаляемого элемента" << endl;

cin >> n;

if (prov()) {

if ((n > fig.size()) || (n <= 0))

cout << "Данного элемента не существует" << endl;

else {

delete fig[n - 1];

fig.erase(fig.begin() + n - 1);

}

}

}

if (k == 0) {

for (int i = 0; i < fig.size(); i++)

delete fig[i];

break;

}

}

}

};

5. Список литературы

Справочник по языку [Электронный ресурс]. URL: <https://ravesli.com/uroki-cpp/> (дата обращения 15.01.2021).

Справочник по языку [Электронный ресурс]. URL:

<https://metanit.com/cpp/tutorial/> (дата обращения 15.01.2021).