**Московский авиационный институт**

**(Национальный исследовательский университет)**

Институт: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Объектно-ориентированное программирование»

**Лабораторная работа № 3**

Тема: Механизмы наследования в С++

Студент: Калугин Кирилл Алексеевич

Группа: 80-207

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2020

1. Постановка задачи

Разработать классы согласно варианту задания, классы должны наследоваться от базового класса Figure. Фигуры являются фигурами вращения. Все классы должны поддерживать набор общих методов:

1. Вычисление геометрического центра фигуры;
2. Вывод в стандартный поток вывода std::cout координат вершин фигуры;
3. Вычисление площади фигуры.

Создать программу, которая позволяет:

* Вводить из стандартного ввода std::cin фигуры, согласно варианту задания.
* Сохранять созданные фигуры в динамический массив std::vector<Figure\*>.
* Вызывать для всего массива общие функции (1-3 см. выше).Т.е. распечатывать для каждой фигуры в массиве геометрический центр, координаты вершин и площадь.
* Необходимо уметь вычислять общую площадь фигур в массиве.
* Удалять из массива фигуру по индексу.

Используемые фигуры: пяти-, шести- и восьмиугольник.

1. Описание программы

Программа создает класс Figure и наследуемые от него классы для пяти-, шести- и восьмиугольников. Взаимодействие с пользователем происходит при помощи интерактивного меню. программа может создавать выбранные пользователем фигуры и хранить их в векторе, печатать координаты центров фигур, их вершины, а также площадь, в том числе и общую. Также возможно удаление фигуры из массивы по индексу.

1. Набор тестов

Тест №1

1.Новый пятиугольник

2.Новый шестиугольник

3.Новый восьмиугольник

4.Вывод всех вершин

5.Вывод всех площадей

6.Вывод всех центров

7.Вывод общей площади

8.Удаление элемента

9.Вывести это меню

10.Выход

Введите команду: 1

Введите координаты центра и радиус: 0 0 1

Введите команду: 2

Введите координаты центра и радиус: 1 1 77

Введите команду: 3

Введите координаты центра и радиус: 5 2 8/6 / .8

Введите команду: 1

Введите координаты центра и радиус: 1 1 8

Введите команду: 4

Вершины фигуры № 1:

A = (0,1)

B = (0.882948,0.469472)

C = (0.71934,-0.694658)

D = (-0.438371,-0.898794)

E = (-0.990268,0.139173)

Вершины фигуры № 2:

A = (1,78)

B = (67.684,39.5)

C = (67.684,-37.5)

D = (1,-76)

E = (-65.684,-37.5)

F = (-65.684,39.5)

Вершины фигуры № 3:

A = (5,10.8)

B = (11.2225,8.22254)

C = (13.8,2)

D = (11.2225,-4.22254)

E = (5,-6.8)

F = (-1.22254,-4.22254)

G = (-3.8,2)

H = (-1.22254,8.22254)

Вершины фигуры № 4:

A = (1,9)

B = (8.06358,4.75577)

C = (6.75472,-4.55727)

D = (-2.50697,-6.19035)

E = (-6.92214,2.11338)

Введите команду: 5

Площадь фигуры № 1: 2.37764

Площадь фигуры № 2: 15404

Площадь фигуры № 3: 219.033

Площадь фигуры № 4: 152.169

Введите команду: 6

Центр фигуры № 1: Центр O = (0,0)

Центр фигуры № 2: Центр O = (1,1)

Центр фигуры № 3: Центр O = (5,2)

Центр фигуры № 4: Центр O = (1,1)

Введите команду: 7

Общая площадь всех фигур равна:15777.6

Введите команду: 8

Введите номер элемента: 1

Введите команду: 4

Вершины фигуры № 1:

A = (1,78)

B = (67.684,39.5)

C = (67.684,-37.5)

D = (1,-76)

E = (-65.684,-37.5)

F = (-65.684,39.5)

Вершины фигуры № 2:

A = (5,10.8)

B = (11.2225,8.22254)

C = (13.8,2)

D = (11.2225,-4.22254)

E = (5,-6.8)

F = (-1.22254,-4.22254)

G = (-3.8,2)

H = (-1.22254,8.22254)

Вершины фигуры № 3:

A = (1,9)

B = (8.06358,4.75577)

C = (6.75472,-4.55727)

D = (-2.50697,-6.19035)

E = (-6.92214,2.11338)

Введите команду: 8

Введите номер элемента: 1

Введите команду: 8

Введите номер элемента: 1

Введите команду: 4

Вершины фигуры № 1:

A = (1,9)

B = (8.06358,4.75577)

C = (6.75472,-4.55727)

D = (-2.50697,-6.19035)

E = (-6.92214,2.11338)

Введите команду: 8

Введите номер элемента: 1

Введите команду: 4

Массив пуст!

Введите команду:

1. Листинг программы

//Калугин Кирилл М8О-207Б-19

//Разработать классы согласно варианту задания, классы должны наследоваться от базового класса Figure. Фигуры являются фигурами вращения. Все классы должны поддерживать набор общих методов:

//1.Вычисление геометрического центра фигуры;

//2.Вывод в стандартный поток вывода std::cout координат вершин фигуры;

//3.Вычисление площади фигуры.

//Создать программу, которая позволяет:

//• Вводить из стандартного ввода std::cin фигуры, согласно варианту задания.

//• Сохранять созданные фигуры в динамический массив std::vector<Figure\*>.

//• Вызывать для всего массива общие функции (1-3 см. выше).Т.е. распечатывать для каждой фигуры в массиве геометрический центр, координаты вершин и площадь.

//• Необходимо уметь вычислять общую площадь фигур в массиве.

//• Удалять из массива фигуру по индексу.

//Используемые фигуры: пяти-, шести- и восьмиугольник.

#include <iostream>

#include <vector>

#include <cmath>

using namespace std;

const double PI = 3.141592653589793238462643383279502884197169399375105820974944;

struct coord {

double x;

double y;

};

class Figure {//создание класса фигур

public:

virtual void centre () {}

virtual void points () {}

virtual double square () {}

};

class five: public Figure {//создание класса-наследника для пятиугольников

public:

coord O;

double R;

double S;

virtual void points () {//метод вывода вершин

cout << "\nA = (" << O.x << "," << O.y + R << ")\n" << "B = ("

<< O.x + R \* cos (28.0 / 180.0 \* PI) << "," << O.y + R \* sin (28.0 / 180.0 \* PI)

<< ")\n" << "C = (" << O.x + R \* cos (- 44.0 / 180.0 \* PI) << ","

<< O.y + R \* sin (- 44.0 / 180.0 \* PI) << ")\n" << "D = ("

<< O.x + R \* cos (- 116.0 / 180.0 \* PI) << "," << O.y + R \* sin (- 116.0 / 180.0 \* PI)

<< ")\n" << "E = (" << O.x + R \* cos (- 188.0 / 180.0 \* PI)

<< "," << O.y + R \* sin (- 188.0 / 180.0 \* PI) << ")\n";

}

virtual double square () {//метод подсчета площади

return (2.5 \* pow (R, 2) \* sin (72.0 / 180.0 \* PI));

}

virtual void centre () {//метод вывода координат центра

cout << "Центр O = " << "(" << O.x << "," << O.y << ")" << "\n";

}

};

class six: public Figure {//создание класса-наследника для шестиугольников

public:

coord O;

double R;

double S;

virtual void points () {//метод вывода вершин

cout << "\nA = (" << O.x << "," << O.y + R << ")\n" << "B = ("

<< O.x + R \* cos (PI / 6.0) << "," << O.y + R \* sin (PI / 6.0)

<< ")\n" << "C = (" << O.x + R \* cos (- PI / 6.0) << ","

<< O.y + R \* sin (- PI / 6.0) << ")\n" << "D = (" << O.x << ","

<< O.y - R << ")\n" << "E = (" << O.x + R \* cos (- 5.0 \* PI / 6.0)

<< "," << O.y + R \* sin (- 5.0 \* PI / 6.0) << ")\n" << "F = ("

<< O.x + R \* cos (- 7.0 \* PI / 6.0) << ","

<< O.y + R \* sin (- 7.0 \* PI / 6.0) << ")\n";

}

virtual double square () {//метод подсчета площади

return (1.5 \* sqrt (3) \* pow (R, 2));

}

virtual void centre () {//метод вывода координат центра

cout << "Центр O = " << "(" << O.x << "," << O.y << ")" << "\n";

}

};

class eight: public Figure {//создание класса-наследника для восьмиугольник

public:

coord O;

double R;

double S;

virtual void points () {//метод вывода вершин

cout << "\nA = (" << O.x << "," << O.y + R << ")\n" << "B = ("

<< O.x + R \* cos (PI / 4) << "," << O.y + R \* sin (PI / 4)

<< ")\n" << "C = (" << O.x + R << "," << O.y << ")\n"

<< "D = (" << O.x + R \* cos (- PI / 4) << "," << O.y + R \* sin (- PI / 4)

<< ")\n" << "E = (" << O.x << "," << O.y - R << ")\n"

<< "F = (" << O.x + R \* cos (- 3 \* PI / 4) << "," << O.y + R \* sin (- 3 \* PI / 4)

<< ")\n" << "G = (" << O.x - R << "," << O.y << ")\n" << "H = ("

<< O.x + R \* cos (- 5 \* PI / 4) << "," << O.y + R \* sin (- 5 \* PI / 4) << ") \n";

}

virtual double square () {//метод подсчета площади

return (4 \* pow (R, 2) \* sin (PI / 4));

}

virtual void centre () {//метод вывода координат центра

cout << "Центр O = " << "(" << O.x << "," << O.y << ")" << "\n";

}

};

int main () {

vector <Figure\*> vault;

coord o;

double r;

int c = 0;

cout << "1.Новый пятиугольник\n2.Новый шестиугольник\n3.Новый восьмиугольник\n4.Вывод всех вершин\n5.Вывод всех площадей\n6.Вывод всех центров\n7.Вывод общей площади\n8.Удаление элемента\n9.Вывести это меню\n10.Выход";

while (c != 10) {

cout << "\nВведите команду: ";

cin >> c;

switch (c) {

case 1: {

cout << "Введите координаты центра и радиус: ";

cin >> o.x >> o.y >> r;

five\* nf = new five;

nf->O.x = o.x;

nf->O.y = o.y;

nf->R = r;

vault.push\_back (nf);

break;

}

case 2: {

cout << "Введите координаты центра и радиус: ";

cin >> o.x >> o.y >> r;

six\* ns = new six;

ns->O.x = o.x;

ns->O.y = o.y;

ns->R = r;

vault.push\_back (ns);

break;

}

case 3: {

cout << "Введите координаты центра и радиус: ";

cin >> o.x >> o.y >> r;

eight\* ne = new eight;

ne->O.x = o.x;

ne->O.y = o.y;

ne->R = r;

vault.push\_back (ne);

break;

}

case 4: {

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

cout << "Вершины фигуры № " << i + 1 << ": ";

vault [i]->points();

}

if (vault.size() == 0) {

cout << "\nМассив пуст!";

}

break;

}

case 5: {

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

cout << "Площадь фигуры № " << i + 1 << ": ";

double S1 = vault [i]->square();

cout << S1 << "\n";

}

if (vault.size() == 0) {

cout << "\nМассив пуст!";

}

break;

}

case 6: {

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

cout << "Центр фигуры № " << i + 1 << ": ";

vault [i]->centre();

}

if (vault.size() == 0) {

cout << "\nМассив пуст!";

}

break;

}

case 7: {

if (vault.size() == 0) {

cout << "\nМассив пуст!";

} else {

double S0 = 0;

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

S0 += vault [i]->square();

}

cout << "\nОбщая площадь всех фигур равна:" << S0;

}

break;

}

case 8: {if (vault.size() == 0) {

cout << "\nМассив пуст!";

} else {

cout << "\nВведите номер элемента: ";

int j = 0;

cin >> j;

vault.erase(vault.begin() + j - 1);

}

break;

}

case 9: {

cout << "1.Новый пятиугольник\n2.Новый шестиугольник\n3.Новый восьмиугольник\n4.Вывод всех вершин\n5.Вывод всех площадей\n6.Вывод всех центров\n7.Вывод общей площади\n8.Удаление элемента\n9.Вывести это меню\n10.Выход";

break;

}

case 10: {

break;

}

default:

cout << "Неверная команда\n";

}

}

return 0;

}

1. Вывод

Наследование классов позволяет создавать более универсальные классы, и, как следствие, упрощает работу с разнотипными данными.

1. Литература

Наследование классов [Электронный ресурс]

URL:http://cppstudio.com/post/10103/ дата обращения (20.10.2020)

Наследование(программирование)[Электронный ресурс]

URL:https://ru.wikipedia.org/wiki/Наследование\_(программирование (20.10.2020)