# **Лабораторная работа № 03**

**Студент**: Сазонов Вадим Кириллович , м8О-201Б-20

# **Тема: Механизмы наследования в С++**

**Цель:**

* Изучение основ работы с механизмами наследования в С++;

## **Требования к программе**

## Разработать классы согласно варианту задания, классы должны наследоваться от базового класса Figure.

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.

Программа должна иметь следующие возможности:

* 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.
* вычислять общую площадь фигур в массиве.
* удалять из массива фигуру по индексу.

## **Вариант 6 :**

|  |  |  |
| --- | --- | --- |
| 5-угольник | 6-угольник | 8-угольник |

Код программы:

#include <iostream>

#include <vector>

#include <math.h>

class Figure

{

public:

virtual float centerx() = 0;

virtual float centery() = 0;

virtual float area() = 0;

virtual void print() = 0;

};

class Fiveanglefigure : public Figure

{

public:

Fiveanglefigure(float a, float b, float length) : x1(a), y1(b), side(length){};

float centerx()

{

return x1 + side / 2;

}

float centery()

{

return y1 + side;

}

float area()

{

return 5 \* side \* side / 4 / tan(36 \* 3.14 / 180);

}

void print()

{

std::cout << x1 << " " << y1 << std::endl;

}

private:

float x1, y1;

float side;

};

class Sixanglefigure : public Figure

{

public:

Sixanglefigure(float a, float b, float length) : x1(a), y1(b), side(length){};

float centerx()

{

return x1 + side / 2;

}

float area()

{

return 6 \* side \* side / 4 / tan(30 \* 3.14 / 180);

}

float centery()

{

return y1 + sqrt(3) / 2 \* side;

}

void print()

{

std::cout << x1 << " " << y1 << std::endl;

}

private:

int x1, y1;

int side;

};

class Eightanglefigure : public Figure

{

public:

Eightanglefigure(float a, float b, float length) : x1(a), y1(b), side(length){};

float centerx()

{

return x1;

}

float centery()

{

return y1 + side \* cos(23 \* 3.14 / 180);

}

float area()

{

return 8 \* side \* side / 4 / tan(23);

}

void print()

{

std::cout << x1 << " " << y1 << std::endl;

}

private:

float x1, y1;

float side;

};

int main()

{

std::vector<Figure \*> vect;

float x1, y1, z1;

std::cin >> x1 >> y1 >> z1;

Figure \*a = new Fiveanglefigure(x1, y1, z1);

std::cin >> x1 >> y1 >> z1;

Figure \*b = new Sixanglefigure(x1, y1, z1);

std::cin >> x1 >> y1 >> z1;

Figure \*c = new Eightanglefigure(x1, y1, z1);

vect.push\_back(a);

vect.push\_back(b);

vect.push\_back(c);

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

{

vect[i]->print();

std::cout << vect[i]->centerx() << " " << vect[i]->centery() << std::endl

<< vect[i]->area() << std::endl;

}

return 0;

}

Тесты:

1. 1 2 3 0 0 1 2 2 5

Результаты тестов:

1 2

2.5 5

15.4947

0 0

0 0.866025

1.73311

2 2

2 6.60292

31.4831