

ГУАП

КАФЕДРА № 43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

должность, уч. степень, звание

подпись, дата

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

Наследование классов, базовый класс, производный класс.
по курсу: ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

4136

подпись, дата

Бобрович Н. С.
инициалы, фамилия

Санкт-Петербург 2022

Цель работы:

Изучить механизм создания нового класса на основе уже существующего, варианты доступа к элементам базового класса из производного.

Закрепить знания по теме:

Классы, наследование классов, варианты доступа.

Условие:

В работе необходимо реализовать базовый класс заданной структуры, на основе него создать производные классы. В нём предусмотреть конструктор для установки начальных значений полей. Создать объекты производных классов. Продемонстрировать работу всех методов, реализуемых в классах.

Вариант 2.

Создать класс Points для хранения координат четырёх точек A, B, C и D на плоскости. В классе предусмотреть возможность распечатки координат каждой точки по отдельности и всех разом. На основе класса Points создать класс Quadrilateral для работы с четырёхугольником. Предусмотреть методы для проверки существования четырёхугольника, нахождения площади и диагоналей.

Листинг программы:

```

1  #include <iostream>
2  #include <math.h>
3
4  using namespace std;
5
6  class Points
7  {
8  public:
9      double x1, x2, x3, x4, y1, y2, y3, y4, S, a, b;
10     void show();
11     void assign();
12     Points() {}
13     Points(double x1, double x2, double x3, double x4, double y1, do
14         this->x1 = x1;
15         this->x2 = x2;
16         this->x3 = x3;
17         this->x4 = x4;
18         this->y1 = y1;
19         this->y2 = y2;
20         this->y3 = y3;
21         this->y4 = y4;
22         this->S = S;
23         this->a = b;
24         this->a = b;
25     }
26     ~Points() {}
27 };
28
29 void Points::show()
30 {
31     cout << "x1= " << x1 << endl << "y1= " << y1 << endl << "x2= " <<
32 }
33 void Points::assign()
34 {
35     cout << "Введите координаты: " << endl;
36     cin >> x1 >> y1 >> x2 >> y2 >> x3 >> y3 >> x4 >> y4;
37     cout << endl;
38 }
39
40 class Quadrilateral : public Points {
41 public:
42     Quadrilateral operator/(Points t) {
43         Quadrilateral vivel;
44         vivel.S = sqrt((x1 - x2) * (x1 - x2) + (y1 - y3) * (y1 - y3));
45         vivel.a = sqrt(pow(x1 - x2, 2) + pow(y1 - y2, 2));
46         vivel.b = sqrt(pow(x1 - x3, 2) + pow(y1 - y3, 2));
47         if ((vivel.S != vivel.a) && (vivel.S != vivel.b)) {
48             return vivel;

```

```

49     }
50     else {
51         cout << "Неправильный четырёхугольник!" << endl;
52         return vivel;
53     }
54 }
55 Quadrilateral operator*(Points t) {
56     Quadrilateral vivel;
57     vivel.S = sqrt((x3 - x4) * (x3 - x4) + (y1 - y4) * (y1 - y4));
58     vivel.a = sqrt(pow(x1 - x2, 2) + pow(y1 - y2, 2));
59     vivel.b = sqrt(pow(x1 - x3, 2) + pow(y1 - y3, 2));
60     if ((vivel.S != vivel.a) && (vivel.S != vivel.b)) {
61         return vivel;
62     }
63     else {
64         cout << "Неправильный четырёхугольник!" << endl;
65         return vivel;
66     }
67 }
68 /*Quadrilateral operator+(Points t) {
69     Quadrilateral vivel;
70     vivel.S = (abs((x1 - x2)*(y1 + y2) + (x2 - x3)*(y2 + y3) + (x3 - x1)*(y3 + y4)));
71     return vivel;
72 }*/
73 void showS()
74 {
75     cout << S << endl;
76 }
77 };
78
79 int main()
80 {
81     system("color F0");
82     setlocale(0, "Rus");
83     Quadrilateral a, b, c;
84     a.assign();
85     b.assign();
86     c = a / b;
87     cout << "Первая диагональ первого четырёхугольника: ";
88     c.showS();
89     cout << endl;
90     c = a * b;
91     cout << "Вторая диагональ первого четырёхугольника: ";
92     c.showS();
93     cout << endl;
94     c = b / a;
95     cout << "Первая диагональ второго четырёхугольника: ";
96     c.showS();

```

```

97         cout << endl;
98         c = b * a;
99         cout << "Вторая диагональ второго четырёхугольника: ";
100        c.showS();
101        cout << endl;
102        return 0;
103    }

```

Результат работы:

```

Консоль отладки Microsoft Visual Studio
Введите координаты:
0
0
2
0
0
2
2
2
2

Введите координаты:
1
1
1
1
1
1
1
1
1

Первая диагональ первого четырёхугольника: 2.82843
Вторая диагональ первого четырёхугольника: 2.82843

Неправильный четырёхугольник!
Первая диагональ второго четырёхугольника: 0

Неправильный четырёхугольник!
Вторая диагональ второго четырёхугольника: 0

```

Вывод:

В результате выполнения работы были получены навыки обращения

