

ГУАП

КАФЕДРА № 43

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

\_\_\_\_\_  
должность, уч. степень, звание

\_\_\_\_\_  
подпись, дата

\_\_\_\_\_  
инициалы, фамилия

## ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

Классы, конструкторы, деструкторы  
по курсу: ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

\_\_\_\_\_  
4136

\_\_\_\_\_  
подпись, дата

\_\_\_\_\_  
Бобрович Н. С.  
инициалы, фамилия

Санкт-Петербург 2022

**Цель работы:**

Изучить принципы создания классов с конструкторами, применив на практике знания базовых синтаксических конструкций языка C++ и объектно-ориентированного программирования.

**Закрепить знания по теме:**

Классы, конструкторы, деструкторы.

**Условие:**

В работе необходимо реализовать класс в соответствии с вариантом задания и создать приложение. В классе должны быть предусмотрены:

- конструктор для установки начальных значений полей,
- конструктор по умолчанию,
- конструктор копирования, деструктор.

Поля класса должны иметь спецификатор доступа `private`. Доступ к полям осуществляется через `public` методы. В функции `main()` создается не менее 3 объектов класса (с использованием всех конструкторов) и осуществляется вызов методов класса.

**Вариант 2.**

2. Определите класс для нахождения углов треугольника. Вещественные поля `a, b, c` - стороны треугольника. Метод `Angles` находит углы.

**Листинг программы:**

```

1  #include <iostream>
2  #include <math.h>
3
4  using namespace std;
5
6  class Smth {
7  private:
8      double a, b, c;
9  public:
10     Smth() {
11         a = 0;
12         b = 0;
13         c = 0;
14     }
15     Smth(double a, double b, double c) {
16         this->a = a;
17         this->b = b;
18         this->c = c;
19     }
20     Smth(const Smth& other) {
21         a = other.a;
22         b = other.c;
23         c = other.c;
24     }
25     double Angles(Smth x)
26     {
27         double r;
28         const double pi = acos(-1.0);
29         double y;
30         y = ((x.b * x.b + x.c * x.c) - x.a * x.a) / (2 * x.c * x.b);
31         r = acos(y) * 180 / pi;
32         return r;
33     }
34     ~Smth() {}
35 };
36
37 int main()
38 {
39     system("color F0");
40     setlocale(LC_ALL, "Rus");
41     double a, b, c;
42     Smth B;
43     cin >> a;
44     cin >> b;
45     cin >> c;
46     if ((a > 0) && (b > 0) && (c > 0)) {
47         Smth A(a, b, c);
48         cout << A.Angles(A) << endl;
49         Smth B(b, a, c);
50         cout << A.Angles(B) << endl;

```

```
50         cout << A.Angles(B) << endl;
51         Smth C(c, b, a);
52         cout << A.Angles(C) << endl;
53     }
54     else {
55         cout << "Error!" << endl;
56     }
57     Smth C = B;
58     return 0;
59 }
60
```

**Результат работы:**

Консоль отладки Microsoft Visual Studio

```
3
4
5
36.8699
53.1301
90
```

C:\Users\User\source\repos\00PLR2\x64\Debug\00PLR2.exe (процесс 21584) завершил работу с кодом 0.  
Нажмите любую клавишу, чтобы закрыть это окно:■

Консоль отладки Microsoft Visual Studio

```
-3
1
1
Error!
```

C:\Users\User\source\repos\00PLR2\x64\Debug\00PLR2.exe (процесс 19844) завершил работу с кодом 0.  
Нажмите любую клавишу, чтобы закрыть это окно:■

Консоль отладки Microsoft Visual Studio

```
10
6
6
112.885
33.5573
33.5573
```

C:\Users\User\source\repos\00PLR2\x64\Debug\00PLR2.exe (процесс 21452) завершил работу с кодом 0.  
Нажмите любую клавишу, чтобы закрыть это окно:■

**Вывод:**

В результате выполнения работы были получены навыки обращения с конструкторами и деконструкторами.