

Лабораторная работа №3. Пользовательские типы данных

В работе предполагается введение пользовательских типов данных в виде структуры (или класса). Определение структуры Point может выглядеть следующим образом:

```
struct Point { double x; double y; };
```

В зависимости от варианта, кроме типа Point, необходимо ввести пользовательский тип данных «Треугольник», «Прямоугольник», «Квадрат», «Ромб» или «Трапеция», которые содержат массив точек в виде внутреннего элемента (поля).

Номер варианта определяем по формуле:

int V = (int(c1) + int(c2)) % 11, где **c1** – первая буква фамилии на *английском* языке в *верхнем* регистре, **c2** – первая буква имени на *английском* языке в *верхнем* регистре. Например, вариант преподавателя V = (int('T') + int('K')) % 11 = 5.

Вариант	Типы данных	Функции
0	Point, Trapeze	1, 3, 4, 7
1	Point, Triangle	1, 2, 4, 8
2	Point, Triangle	1, 3, 5, 9
3	Point, Rectangle	1, 2, 5, 6
4	Point, Rectangle	1, 3, 4, 7
5	Point, Square	1, 2, 4, 6
6	Point, Square	1, 3, 5, 7
7	Point, Rhomb	1, 2, 5, 6
8	Point, Rhomb	1, 3, 4, 7
10	Point, Trapeze	1, 2, 5, 6
11	Point, Trapeze	1, 3, 4, 7

Функции для реализации определяются в соответствии с вариантом. Приведенные сигнатуры можно изменять. В реализации одной из функций использовать ссылочный тип данных (н-р, Point& или Rectangle&) для аргументов или возвращаемого значения.

№	Описание функции
1	Вычисление длины отрезка, соединяющего две точки double length(Point p1, Point p2);
2	Принадлежат ли три точки одной прямой? bool inLine(Point points[]);
3	Нахождение середины отрезка, соединяющего две точки Point midPoint(Point p1, Point p2);
4	Нахождение ближайших двух точек в массиве void getClosest(Point* points, int size, Point* p1, Point* p2);
5	Нахождение двух наиболее удаленных точек в массиве void getClosest(Point* points, int size, Point* p1, Point* p2);
6	Проверка корректности фигуры bool isValid(Rectangle& rect);
7*	Создание фигуры (квадрата, прямоугольника или ромба) bool getSquare(Point points[], Square& square);
8*	Нахождение прямоугольных треугольников int getRectTriangles(Triangles t[], int size, Triangles* selected);
9*	Нахождение равнобедренных треугольников

	int getIsoscelesTriangles(Triangles t[], int size, Triangles* selected);
10**	Проверка пересечения фигур
11**	Проверка вложенности фигур
12**	Предложить свою функцию для работы с фигурами