

Задания к лекции 1

1. Описать тип-структуру «вектор на плоскости», хранящую координаты x и y.
Описание структуры может выглядеть примерно так:

```
struct Vector2
{
    double x;
    double y;
};
```

Описать функции для сложения и вычитания векторов, которые принимают в качестве параметров два вектора и возвращают вектор в качестве результата.

Описать функцию, принимающую вектор в качестве параметра и «красиво» выводящую его на консоль (например, координаты через запятую в круглых скобках).

Описать перечисление для задания квадранта координатной плоскости (I, II, III, IV).

Описать функцию, принимающую вектор в качестве параметра и возвращающую квадрант, в котором он находится (как элемент перечисления).

2. Описать тип-структуру «комплексное число», хранящую действительную и мнимую части комплексного числа.

Описание структуры может выглядеть примерно так:

```
struct ComplexNumber
{
    double re;
    double im;
};
```

Описать функции для сложения и умножения комплексных чисел, которые принимают в качестве параметров два комплексных числа и возвращают комплексное число в качестве результата.

Описать функцию, принимающую комплексное число в качестве параметра и «красиво» выводящую его на консоль (например, так: $2.71 + i*3.14$).

Описать перечисление для задания вида комплексного числа: действительное, мнимое, комплексное. Описать функцию, принимающую комплексное число в качестве параметра и возвращающую вид числа.

3. Описать тип-структуру «точка на плоскости в декартовых координатах». Описание структуры может выглядеть, например, так:

```
struct Point2
{
    double x;
    double y;
};
```

Описать функцию, принимающую в качестве параметров три точки и возвращающую периметр соответствующего треугольника.

Описать функцию, принимающую точку в качестве параметра и «красиво» выводящую её на консоль (например, координаты через запятую в круглых скобках).

Описать перечисление для задания вида треугольника: равнобедренный, равносторонний, общего вида. Описать функцию, принимающую в качестве параметров три точки и возвращающую вид соответствующего треугольника.