

# 1 Лабораторная

## *C++ - пространство имён, перегрузка функций*

За выполнение всех заданий лабораторной - **8 баллов** (до 19 марта, за сдачу после 19 марта ставлю половину).

Выполнить один вариант. Номер варианта = номер из списка подгруппы % 10

Также при выполнении заданий из своего варианта должны быть учтены следующие требования:

- В отдельном заголовочном файле .h разместить собственное namespace пространство имён, в котором объявить перегруженные функции из своего варианта ниже. В файле реализации .cpp, относящемся к данному заголовочному файлу – определить функции;
- В основной программе main продемонстрировать работоспособность своих функций и правильное использование собственного и стандартного std пространства имён;
- При подключении к main-у собственного заголовочного файла проверить, что в нём исключена возможность определения одних и тех же заголовочных файлов многократно (должна присутствовать в заголовочном файле директива условной компиляции header guards);
- В программе сделать по одной замене/определения псевдонима для типа, используя typedef, using и директиву препроцессора define. Объяснить различие между этими тремя возможностями.

### Вариант 0

- Перегрузить три функции, которые выводят символ '!' 50 раз, выводят заданный символ 50 раз, выводят заданный символ заданное число раз;
- Написать одну функцию, использующую аргументы по умолчанию, которая делает, что и предыдущие три;
- Перегрузить две функции для типов int, double и сделать ещё шаблонный вариант функции - в одномерном массиве вычислить сумму отрицательных элементов массива, возвращаемое значение – вещественное число;
- Написать функцию с неизвестным количеством параметров, которая ищет минимальный элемент в списке параметров.

### Вариант 1

- Перегрузить три функции, которые выводят символ '@' 20 раз, выводят заданный символ 30 раз, выводят заданный символ заданное число раз;

- Написать одну функцию, использующую аргументы по умолчанию, которая делает, что и предыдущие три;
- Перегрузить две функции для типов `int`, `double` и сделать ещё шаблонный вариант функции - в одномерном массиве вычислить сумму положительных элементов массива, возвращаемое значение – целое число;
- Написать функцию с неизвестным количеством параметров, которая ищет максимальный элемент в списке параметров.

## Вариант 2

- Перегрузить три функции, которые выводят символ `'&'` 15 раз, выводят заданный символ 5 раз, выводят заданный символ заданное число раз;
- Написать одну функцию, использующую аргументы по умолчанию, которая делает, что и предыдущие три;
- Перегрузить две функции для типов `int`, `double` и сделать ещё шаблонный вариант функции – в одномерном массиве вычислить сумму произведения элементов массива с четными номерами, возвращаемое значение – целое число;
- Написать функцию с неизвестным количеством параметров, которая считает количество четных элементов в списке параметров.

## Вариант 3

- Перегрузить три функции, которые выводят символ `'!'` 20 раз, выводят заданный символ 10 раз, выводят заданный символ заданное число раз;
- Написать одну функцию, использующую аргументы по умолчанию, которая делает, что и предыдущие три;
- Перегрузить две функции для типов `int`, `double` и сделать ещё шаблонный вариант функции - в одномерном массиве вычислить сумму произведения элементов массива с нечетными номерами, возвращаемое значение – целое число;
- Написать функцию с неизвестным количеством параметров, которая ищет максимальный из элементов в списке параметров, стоящих на четных местах.

## Вариант 4

- Перегрузить три функции, которые выводят символ `'a'` 100 раз, выводят заданный символ 100 раз, выводят заданный символ заданное число раз;
- Написать одну функцию, использующую аргументы по умолчанию, которая делает, что и предыдущие три;

- Перегрузить две функции для типов `int`, `double` и сделать ещё шаблонный вариант функции - в одномерном массиве вычислить максимальный элемент массива, возвращаемое значение – вещественное число;
- Написать функцию с неизвестным количеством параметров, которая ищет максимальный из элементов в списке параметров, стоящих на нечетных местах.

#### **Вариант 5**

- Перегрузить три функции, которые выводят символ '\*' 50 раз, выводят заданный символ 50 раз, выводят заданный символ заданное число раз;
- Написать одну функцию, использующую аргументы по умолчанию, которая делает, что и предыдущие три;
- Перегрузить две функции для типов `int`, `double` и сделать ещё шаблонный вариант функции - в одномерном массиве вычислить минимальный элемент массива, возвращаемое значение – целое число;
- Написать функцию с неизвестным количеством параметров, которая ищет среднее арифметическое из элементов в списке параметров, стоящих на четных местах.

#### **Вариант 6**

- Перегрузить три функции, которые выводят символ ':' 30 раз, выводят заданный символ 40 раз, выводят заданный символ заданное число раз;
- Написать одну функцию, использующую аргументы по умолчанию, которая делает, что и предыдущие три;
- Перегрузить две функции для типов `int`, `double` и сделать ещё шаблонный вариант функции - в одномерном массиве вычислить номер максимального элемента массива;
- Написать функцию с неизвестным количеством параметров, которая ищет среднее арифметическое из элементов в списке параметров, стоящих на нечетных местах.

#### **Вариант 7**

- Перегрузить три функции, которые выводят символ '(' 25 раз, выводят заданный символ 35 раз, выводят заданный символ заданное число раз;
- Написать одну функцию, использующую аргументы по умолчанию, которая делает, что и предыдущие три;
- Перегрузить две функции для типов `int`, `double` и сделать ещё шаблонный вариант функции - в одномерном массиве вычислить номер минимального элемента массива;

- Написать функцию с неизвестным количеством параметров, которая считает произведение всех элементов.

#### **Вариант 8**

- Перегрузить три функции, которые выводят символ ')' 80 раз, выводят заданный символ 40 раз, выводят заданный символ заданное число раз;
- Написать одну функцию, использующую аргументы по умолчанию, которая делает, что и предыдущие три;
- Перегрузить две функции для типов int, double и сделать ещё шаблонный вариант функции – в одномерном массиве вычислить максимальный по модулю элемент массива, возвращаемое значение – вещественное число;
- Написать функцию с неизвестным количеством параметров, которая ищет минимальный элемент в списке параметров.

#### **Вариант 9**

- Перегрузить три функции, которые выводят символ '%' 20 раз, выводят заданный символ 100 раз, выводят заданный символ заданное число раз;
- Написать одну функцию, использующую аргументы по умолчанию, которая делает, что и предыдущие три;
- Перегрузить две функции для типов int, double и сделать ещё шаблонный вариант функции - в одномерном массиве вычислить минимальный по модулю элемент массива, возвращаемое значение – целое число;
- Написать функцию с неизвестным количеством параметров, которая считает количество нечётных элементов в списке параметров.