

Лабораторная работа 7. Функции с произвольным количеством входных и выходных аргументов

Задание 1

Написать функцию `Sum`, которая находит сумму произвольного количества аргументов, которые могут быть любыми объектами (использовать `varargin`), а также выводит количество входных аргументов. Проанализировать результаты, вызвав функцию следующим образом:

```
[sum1, num1]=Sum(1, 2, 3, 4, 5)
[sum2, num2]=Sum(10, 'a') %посмотреть ascii-код символа
[sum3, num3]=Sum(10, 'ab')
[sum4, num4]=Sum('abc', 'def')
[sum5, num5]=Sum(1, 2, 3, 4, 5, [10 20 30])
[sum6, num6]=Sum(1, 2, 3, 4, 5, [10;20;30])
[sum7, num7]=Sum(1, 2, 3, 4, 5, [10 20;30 40])
```

Написать выводы в комментариях к каждому вызову.

Задание 2

Написать функцию `MaxMinVector`, которая принимает произвольное количество (использовать `varargin`) векторов и возвращает два выходных параметра: максимальный и минимальный элементы среди элементов всех векторов.

Вызвать функцию с разным количеством выходных параметров (без выходных параметров, с одним, двумя, более двух) и проанализировать результат. Протестировать работу функции для вектор-строк, вектор-столбцов, матриц, скаляров, других объектов (например типа `char` или `string`).

Указание. Организовать проверку, что все входные параметры являются векторами (функция контроля типа `isvector`). В случае, если хотя бы один входной параметр не является вектором, функция `MaxMinVector` должна выдавать сообщение об ошибке (команда `error`) с указанием, какой из входных параметров не является вектором (см. `help` по команде `error` для создания форматированного сообщения об ошибке).

Задание 3

Написать функцию `MaxMinMatrix`, которая принимает произвольное количество (использовать `varargin`) матриц и возвращает два выходных параметра: максимальный и минимальный элементы среди элементов всех матриц. Вызвать функцию с разным количеством выходных параметров (без выходных параметров, с одним, двумя, более двух) и проанализировать результат. Протестировать работу функции для матриц, вектор-строк, вектор-столбцов, скаляров, других объектов.

Указание. Организовать проверку, что все входные параметры являются матрицами (функция контроля типа `ismatrix`). В случае, если хотя бы один входной параметр не является матрицей, функция `MaxMinMatrix` должна выдавать сообщение об ошибке (команда `error`) с указанием, какой из входных параметров не является матрицей (см. `help` по команде `error` для создания форматированного сообщения об ошибке).

Задание 4

Написать функцию `MinMaxAverageV`, которая принимает на вход вектор и возвращает от одного до трех выходных параметров (использовать `varargout`): первый параметр – минимальный элемент, второй параметр – максимальный элемент, третий параметр

– среднее значение всех элементов. Вызвать функцию с разным количеством выходных параметров (без выходных параметров, с одним, двумя, тремя, более трех) и проанализировать результат. Будет ли выводиться ошибка, если количество выходных параметров больше трех?

Указание. Функция должна проверять, что входной параметр является вектором.

Задание 5

Написать функцию `AnalyseV`, которая принимает на вход вектор и возвращает:

- среднее значение, если вызывается с одним выходным параметром или без указания выходных параметров;
- минимум и максимум, если вызывается с двумя выходными параметрами.

В случае, если функция вызывается с тремя и более выходными параметрами, выдавать сообщение об ошибке: «Слишком много выходных параметров». Проверить работу функции, организовав вызов без параметров, с одним выходным параметром, с двумя выходными параметрами и т.д.

Указание. Функция должна проверять, что входной параметр является вектором.