

### 3. Лабораторная работа №5

#### Задание №1

#### (СДАЁТСЯ ТОЛЬКО В ТЕЧЕНИЕ ЛАБОРАТОРНОЙ РАБОТЫ):

Создать метод, реализующий:

- Ввод переменной через форму с текстовым полем (как вариант – диалоговое окно InputBox).
- Перехват и обработку типовых ошибок ввода/конверсии данных (Type mismatch, Overflow).
- Вывод сообщения об ошибках с использованием формы MsgBox/MessageBox и обработать реакцию пользователя на ошибку (прекращение расчётов, возврат к вводу данных и т.п.).

Действия при выборе кнопок пользователем:

**Ignore** – задать значение по умолчанию (игнорировать ввод).

**Cancel/No/Abort** – Выход из программы

**Retry** – Повторить ввод.

**Примечание 1.** Если исключение не вызывается, а получается значение типа *double.NaN* или аналогичные, то сгенерировать исключение вручную.

**Примечание 2.** В Python можно использовать, например *tkinter.messagebox*

**Примечание 3.** Типы данных заданы для C#. Если в выбранном языке их нет заменить на ближайший похожий тип.

A	B	C	D
Тип переменной	Тип значка	Кнопки	Кнопка по умолчанию
1. Byte 2. Short 3. int 4. uint 5. long 6. ulong 7. float 8. double 9. decimal 10. bool	1. Нет 2. Информационный 3. Вопросительный 4. Восклицательный 5. Ошибка	1. Продолжать вычисления или остановиться (Ok/Cancel) 2. Повторить ввод, игнорировать ввод или остановиться (Abort/Retry/Ignore) 3. Повторить ввод или остановиться (Yes/No) 4. Повторить ввод, остановиться или задать значение по умолчанию (Yes/No/Cancel) 5. Повторить ввод или остановиться (Retry/Cancel)	1. Нет 2. Первая 3. Вторая

Вариант	Задание 1				Задание 2	Вариант	Задание 1				Задание 2
	A	B	C	D			A	B	C	D	
1.	8	1	2	3	8	41.	5	1	4	2	15
2.	1	4	3	2	38	42.	8	2	2	3	35
3.	6	2	5	1	23	43.	4	5	3	1	36
4.	9	5	1	2	7	44.	1	3	1	2	6
5.	3	3	4	1	6	45.	3	4	5	3	10
6.	4	2	5	3	19	46.	6	5	3	1	7
7.	2	4	4	1	34	47.	7	1	2	2	11
8.	10	5	2	3	1	48.	10	4	1	3	16
9.	7	1	1	2	33	49.	9	3	5	1	12
10.	5	3	3	1	4	50.	2	2	4	3	25
11.	3	2	5	2	11	51.	9	1	3	2	24
12.	7	1	3	3	25	52.	3	2	5	2	2
13.	10	5	2	1	31	53.	4	5	2	3	29
14.	5	4	1	3	29	54.	8	3	1	1	19
15.	4	3	4	2	9	55.	10	4	4	1	1

Вариант	Задание 1				Задание 2	Вариант	Задание 1				Задание 2
	A	B	C	D			A	B	C	D	
16.	8	5	1	2	12	56.	7	5	4	2	34
17.	9	1	5	3	32	57.	1	2	2	3	38
18.	2	3	3	1	21	58.	6	4	3	1	27
19.	6	2	4	2	35	59.	5	3	1	3	20
20.	1	4	2	3	26	60.	2	1	5	2	9
21.	2	3	3	1	13	61.	3	4	3	1	22
22.	9	2	4	2	36	62.	4	5	5	3	26
23.	4	1	1	3	5	63.	7	2	1	2	13
24.	3	5	5	1	3	64.	10	3	4	2	10
25.	10	4	2	1	24	65.	8	1	2	3	31
26.	5	3	4	2	39	66.	6	2	4	1	23
27.	6	2	5	3	27	67.	1	1	3	3	32
28.	7	4	1	2	2	68.	5	5	5	1	16
29.	1	5	2	3	3	69.	2	3	2	2	15
30.	8	1	3	1	22	70.	9	4	1	3	14
31.	9	3	5	3	18	71.	10	2	2	2	37
32.	7	2	2	1	30	72.	1	5	3	1	40
33.	5	5	3	2	37	73.	4	1	1	1	8
34.	10	4	4	3	4	74.	9	3	4	2	17
35.	4	1	1	1	28	75.	8	4	5	3	28
36.	1	2	5	2	20	76.					
37.	3	1	4	2	40	77.					
38.	8	4	1	1	5	78.					
39.	2	3	3	3	14	79.					
40.	6	5	2	1	17	80.					

## Задание №2

(Таблица вариантов функций к заданию приведена выше)

а – 2 балла, б – 2 балла, в – 3 балла

а) Требуется рассчитывать функцию от двух переменных  $G(x_i, y_j)$  при определённых пользователем вещественных значениях  $x_i$  и  $y_j$ .  $i=0..m$ ;  $j=0..n$

- Все заданные пользователем вещественные значения  $x_i$  и  $y_j$  будем называть наборами исходных данных.

Создать программу расчёта, где пользователем вводятся:

- Количество наборов исходных данных.
- Вычисление функции  $G(x, y)$  оформить в виде отдельной функции (метода) с обработкой вычислительных ошибок.

Произвести расчёты для 3-х или более наборов исходных данных.

**Внимание!** Задание исходных данных для расчётов каждого набора производится **единовременно** с формы или консоли, а не последовательно. Ввод данных через множество форм или InputBox-ов, а также вывод результатов через **MessageBox-ы запрещен!!!**

Пример возможного пользовательского интерфейса для одного из вариантов:

Номер набора

Заккрыть форму

Произвести расчет всех наборов

Это только один из вариантов исполнения

1 | 2 | 3 |

X0: -10 Y0: -5

Xk: 10 Yk: 100

Шаг X: 2,5 Шаг Y: 2,5

+ -

-10 -7.5 -5 -2.5 0 2.5 5 7.5 10

Расчет

Отмена

Вывести точки

Удалить набор

Добавить набор

Вывести все точки диапазона в таблицу (опционально)

**Примечание:** Значения функции не обязательно должны быть корректными в заданных интервалах.

### Функции G(x,y)

**Номер задания не совпадает с номером варианта!!!**

**Номер брать из таблицы выше**

Задание2	G(x,y)	Задание2	G(x,y)	Задание2	G(x,y)
1.	$\lg(\cos(x^2))/y$	15.	$y\sqrt{x^5-2}$	29.	$y/\arctan(x)$
2.	$\lg(x/y)$	16.	$x/(y-2)$	30.	$\exp(-y/ x )$
3.	$y/\sin(-x^2)$	17.	$y/ \log_2(x) $	31.	$\sin(\sqrt[3]{-2y/x^2})$
4.	$\sqrt{y}/\ln(x)$	18.	$\exp(-\sqrt{y/(1+x)})$	32.	$\sqrt{\frac{x^3-8}{y}}$
5.	$\sqrt[3]{y}\lg(x-5)$	19.	$y\cos(2/\sqrt{100-x^2})$	33.	$y\exp(\sqrt{x})$
6.	$\lg((x^5-7)/2y)$	20.	$y\cos(\sqrt[3]{2/x})$	34.	$y \times 10^{\ln(x)+1}$
7.	$\exp(y/\sin(x))$	21.	$\sqrt[3]{x^5-2}/y$	35.	$2^{1+\sqrt{xy}}$
8.	$\sqrt{x^3-2y}$	22.	$x/ \log_3(y) $	36.	$y/\cos(1/x^2)$
9.	$2^{x/\sqrt{y}}$	23.	$x/(1/y-5)$	37.	$y/\arctan(1/x)$
10.	$x/\arccos(1/y)$	24.	$x/\exp(y)$	38.	$(1/x)*y^2$
11.	$\sqrt{\cos(x+y)}$	25.	$\sqrt{\cos(x/y)}$	39.	$\ln(y/x)$
12.	$\exp(y/\ln(x))$	26.	$\ln(y)\sqrt{x}$	40.	$\text{Log}_2(y/x)$
13.	$y/\lg(x)$	27.	$\text{ch}(x)/\exp(y)$	41.	
14.	$\sqrt{x}/\ln(y)$	28.	$y \times 2^{\ln(x-1)}$	42.	

### Наборы исходных данных для вариантов

(вариант брать тот же, что и для номера функции):

X\Y	ФШу	ФТу	ФГу	Пу
ФШх	1.	2.	3.	4.
ФТх	5.	6.	7.	8.
ФГх	9.	10.	11.	12.
Пх	13.	14.	15.	16.
ФШх	17.	18.	19.	20.
ФГх	21.	22.	23.	24.
ФТх	25.	26.	27.	28.
Пх	29.	30.	31.	32.
ФШх	33.	34.	35.	36.
ФГх	37.	38.	39.	40.

- ФШ – Фиксированный, с заданным шагом (ФШх, ФШу).
- ФТ – Фиксированный, с заданным количеством точек (ФТх, ФТу)
- ФГ – Фиксированный с заданными границами (ФГх, ФГу)

- П – Произвольный (Пх, Пу)<sup>1</sup>

**Пример ФШх:** Задано начальное значение  $x_0$ , конечное  $x_k$  и шаг  $h_x$ .

**Пример ФТх:** Задано начальное значение  $x_0$ , количество точек  $N_x$  и шаг  $h_x$ .

**Пример ФГх:** Задано начальное значение  $x_0$ , конечное  $x_k$  и количество точек  $N_x$ .

**Пример Пх:** Заданы все значения  $x_i$  ( $i=0.. N_x-1$ ) в возрастающем порядке. (Данные задаются только пользователем(!), а не программно)

б) В ходе выполнения программы должен создаваться следующие файлы последовательного доступа:

- Текстовый файл **myProgram.log** в котором содержатся
  1. Название программы и номер варианта
  2. Дата и время начала выполнения расчёта
  3. Рассчитываемая функция  
(Примечание: символы типа корня заменить на соответствующие степени).
  4. Названия файлов, содержащих результаты расчётов по заданным наборам исходных данных (G0001.dat, G0002.dat и т.д.).
- Текстовый файл регистрации ошибок **myErrors.log** в котором собрать ошибки в формате:
  1. Имя файла данных (G0001.dat, G0002.dat и т.д.).
  2. Рассчитываемая функция (Примечание: символы типа корня заменить на соответствующие степени).
  3. Аргументы  $x, y$
  4. Тип ошибки (деление на 0, переполнение, и пр.)
- Текстовые файлы данных с именами по шаблону G####.dat (#### – номер набора исходных данных), под каждый набор исходных данных. Вывести **в каждый** такой файл следующую информацию:
  1. Рассчитываемая функция (Примечание: символы типа корня заменить на соответствующие степени).
  2. Количество точек для  $x$  и  $y$  для аргументов функции  $G(x, y)$ .
  3. Значения функции  $G(x, y)$  в табличном виде с выравниванием элементов для четных/нечетных вариантов:

Шаблон таблицы для  $G(x,y)$  для **нечётных** вариантов:

$x \backslash y$	$\langle y_0 \rangle,$	$\langle y_1 \rangle,$	$\langle y_2 \rangle \dots$	$\langle y_n \rangle$
$\langle x_0 \rangle,$	$G(x_0, y_0),$	$G(x_0, y_1),$	$G(x_0, y_2), \dots$	$G(x_0, y_n)$
$\langle x_1 \rangle,$	$G(x_1, y_0),$	$G(x_1, y_1),$	$G(x_1, y_2), \dots$	$G(x_1, y_n)$
....				
$\langle x_m \rangle,$	$G(x_m, y_0),$	$G(x_m, y_1),$	$G(x_m, y_2), \dots$	$G(x_m, y_n)$

Шаблон таблицы для  $G(x,y)$  для **чётных** вариантов:

$x \backslash y$	$\langle x_0 \rangle,$	$\langle x_1 \rangle,$	$\langle x_2 \rangle \dots$	$\langle x_m \rangle$
$\langle y_0 \rangle,$	$G(x_0, y_0),$	$G(x_1, y_0),$	$G(x_2, y_0), \dots$	$G(x_m, y_0)$
$\langle y_1 \rangle,$	$G(x_0, y_1),$	$G(x_1, y_1),$	$G(x_2, y_1), \dots$	$G(x_m, y_1)$
....				
$\langle y_n \rangle,$	$G(x_0, y_n),$	$G(x_1, y_n),$	$G(x_2, y_n), \dots$	$G(x_m, y_n)$

где  $\langle x_i \rangle$  и  $\langle y_j \rangle$  – конкретные числовые значения, а не подписи.

**Примечание 1:** В случае, когда для какого-то набора аргументов функция не определена, то в соответствующей позиции  $G(x, y)$  поместить текст «NaN», «Null» или схожий по смыслу.

**Примечание 2:** При выводе сохранять все значащие цифры аргументов и функций.

**Примечание 3:** Количество разрядов в имени файлов G####.dat задавать параметрически.

**Примечание 4:** Бонусная часть **+1 балл** за интерфейс ввода исходных данных.

в) Создать модуль, считывающий из dat-файлов, описанных в файле **myProgram.log**, всех данные в соответствующие внутренние массивы и/или табличные элементы управления.

- Предусмотреть возможность наличия ошибок при считывании (достижение конца файла, считывание некорректных данных)

<sup>1</sup> Произвольные точки задаются пользователем в консоли или с помощью элементов графического пользовательского интерфейса.

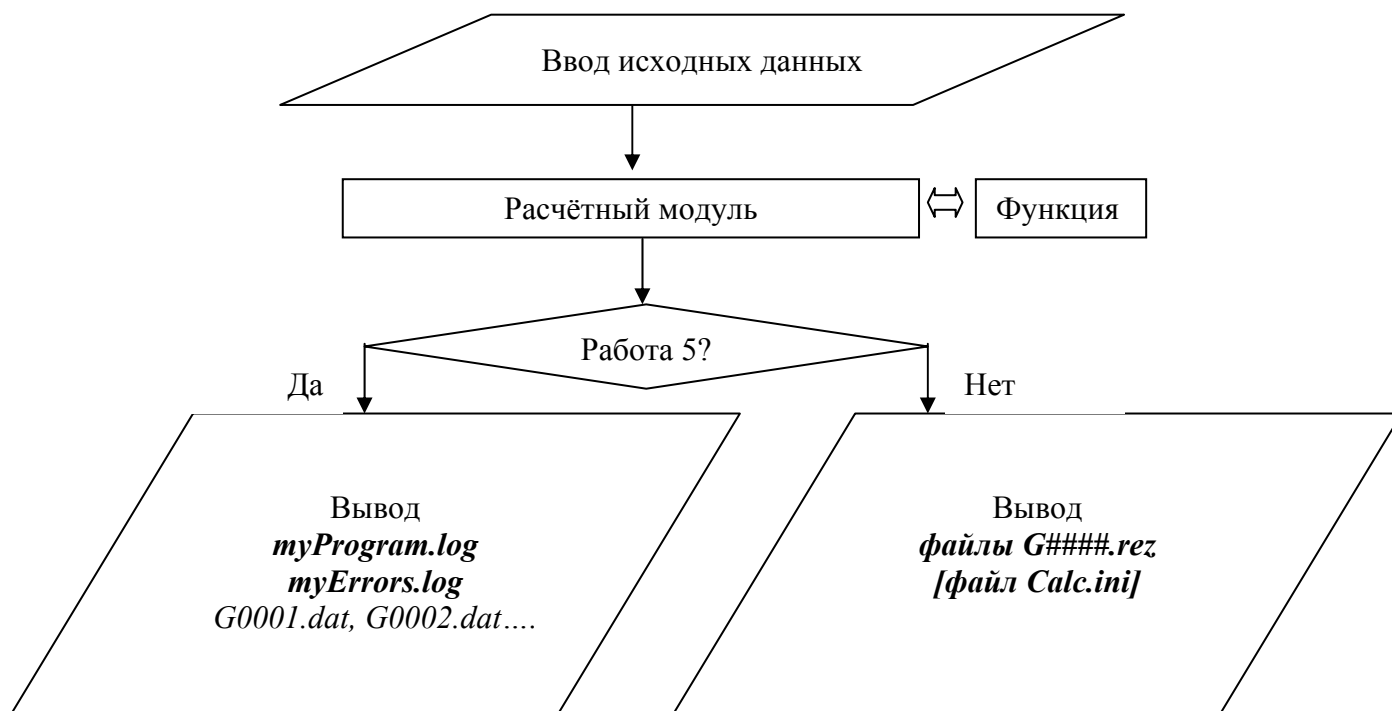
- Считанные данные должны быть доступными к использованию другими программными единицами через заголовок функций/методов в виде соответствующих массивов  $x$ ,  $y$  и матрицы  $G$ . Представить эти данные в виде класса или структуры.

**Примечание 1:** Блок считывания не должен быть связан с расчётным блоком. Т.е. все необходимые для своей работы данные он берет с интерфейса/консоли и из файлов. Сам интерфейс может объединять расчётный и считывающий блоки.

**Примечание 2:** Бонусная часть **+1 балл** за интерфейс вывода считанных данных из файлов. Пользовательский интерфейс должен позволять выбор набора(ов) и отображение считанных данных на форме.

## Принципиальная блок-схема работ 5 и 6

Расчётный блок:



Блок считывания (может быть изолированным от предыдущего):

