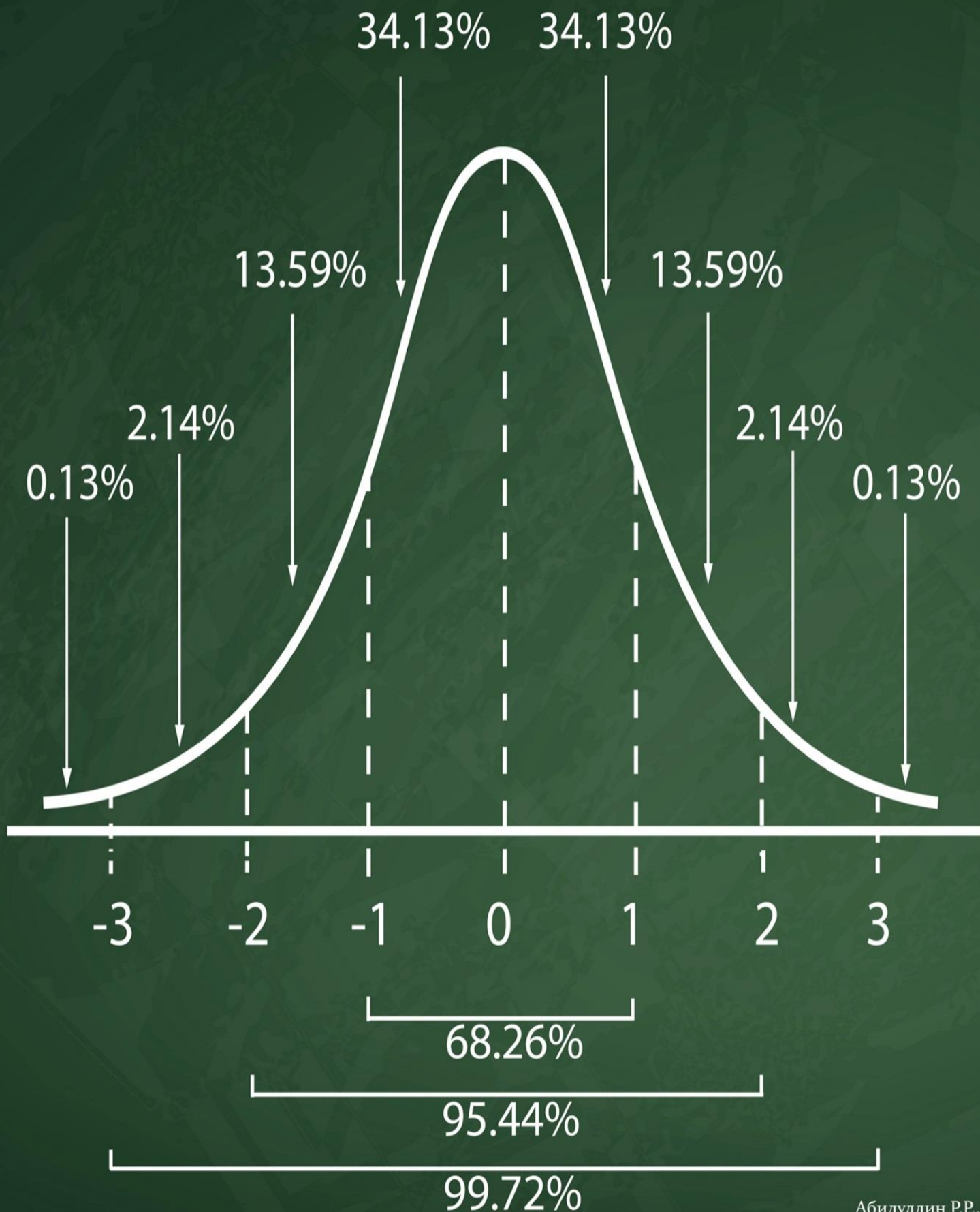


VariableExpressions

Руководство пользователя



VariableExpressions

Версия 2.3

Программа для вычисления выражений вероятностным методом. Позволяет вычислять любые выражения, в которых присутствуют величины с допусками (например, размерные цепи, определение допуска положения центра масс и т.д.).

Идея приложения заключается в том, чтобы многократно вычислять заданное выражение, каждый раз вычисляя конкретные значения величин с допусками в соответствии с нормальным законом их распределения. Имея в распоряжении массив вычисленных значений, можно определить для него математическое ожидание и среднеквадратическое отклонение.

Как пользоваться программой

Шаг 1. Подготовить текстовый файл с выражением.

Выражение записывается в виде формулы, результатом вычисления которой должно быть число. Например, такими формулами могут быть сумма размеров для вычисления замыкающего звена, отношение суммы произведений масс корпусов и расстояний до центров их масс к сумме масс корпусов для вычисления центра масс изделия и т.д.

Примеры выражений:

Замыкающее звено размерной цепи:

```
-gdu(1131, -1.05, 0) + gdu(6, 0, +0.03) - gdu(6.5, -0.2, +0.2) +  
gdu(328, -0.36, 0) + gdu(344, -1.4, 0) + gdu(398, -0.36, 0) -  
gdu(3, -0.3, +0.3) - gdu(1, -0.3, +0.3) + gdu(70, -0.5, +0.5)
```

Вышеприведенное выражение эквивалентно следующему:

$$\begin{aligned} & -1131_{-1,05} + 6^{+0,03} - 6,5^{+0,2}_{-0,2} + 328_{-0,36} + 344_{-1,4} + \\ & + 398_{-0,36} - 3^{+0,3}_{-0,3} - 1^{+0,3}_{-0,3} + 70^{+0,5}_{-0,5} \end{aligned}$$

```
(
  gpp(20,-4,+2) * gdu(520,-1.5,+1) +
  gpp(260,-4,+2) * gdu(2500,-2,+2) +
  gdu(1230,-17,+15) * gdu(5900,-30,+30) +
  gdu(7510,-70,+70) * gdu(9800,-50,+30)
)
/
(
  gpp(20,-4,+2) +
  gpp(260,-4,+2) +
  gdu(1230,-17,+15) +
  gdu(7510,-70,+70)
)
)
```

Вышеприведенное выражение эквивалентно следующему:

$$\frac{20_{-4\%}^{+2\%} \times 520_{-1.5}^{+1} + 260_{-4\%}^{+2\%} \times 2500_{-2}^{+2} + 1230_{-17}^{+15} \times 5900_{-30}^{+30} + 7510_{-70}^{+70} \times 9800_{-50}^{+30}}{20_{-4\%}^{+2\%} + 260_{-4\%}^{+2\%} + 1230_{-17}^{+15} + 7510_{-70}^{+70}}$$

Программа не учитывает символы пробелов, табуляций и переносов строк. Это позволяет отформатировать выражение в удобном для восприятия виде.

Допускается использовать следующие элементы в выражении:

Операторы

+	Сложение
-	Вычитание
*	Умножение
/	Деление
^	Возведение в степень
%	Остаток от деления

Константы

pi	$\pi = 3.14159 \dots$
e	$e = 2,71828 \dots$

Функции случайных распределений

<code>gauss(min, max)</code>	Возвращает случайное число в соответствии с нормальным законом распределения из диапазона [min , max]
<code>gauss_down_up(nominal, down, up)</code>	Возвращает случайное число в соответствии с нормальным законом распределения для номинального значения nominal с нижним отклонением down и верхним отклонением up , т.е.: $\text{nominal}^{\text{up}}_{\text{down}}$
<code>gauss_percents(nominal, downCent, upCent)</code>	Возвращает случайное число в соответствии с нормальным законом распределения для номинального значения nominal с нижним отклонением downCent , и верхним отклонением upCent , выраженных в процентном виде, т.е.: $\text{nominal}^{\text{upCent}\%}_{\text{downCent}\%}$
<code>gmm(min, max)</code>	Краткий вариант записи функции <code>gauss(min, max)</code>
<code>gdu(nominal, down, up)</code>	Краткий вариант записи функции <code>gauss_down_up(nominal, down, up)</code>
<code>gpp(nominal, downCent, upCent)</code>	Краткий вариант записи функции <code>gauss_percents(nominal, downCent, upCent)</code>

Функции математические

abs(x)	Возвращает абсолютное значение (модуль) числа x , математическая форма записи $ x $
acos(x)	Возвращает значение (в радианах) функции $\arccos(x)$ (арккосинус). Возвращаемое значение лежит в диапазоне $[0, \pi]$
asin(x)	Возвращает значение (в радианах) функции $\arcsin(x)$ (арксинус). Возвращаемое значение лежит в диапазоне $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$
atan(x)	Возвращает значение (в радианах) функции $\arctg(x)$ (арктангенс). Возвращаемое значение лежит в диапазоне $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$
ceil(x)	Возвращает наименьшее целое число, не меньшее x
cos(x)	Возвращает значение функции $\cos(x)$, x задается в радианах
cosh(x)	Возвращает значение функции $\operatorname{ch}(x)$ (гиперболический косинус)
deg2rad(x)	Конвертирует угол, заданный в градусах, в радианы
exp(x)	Возвращает значение функции e^x
floor(x)	Возвращает наибольшее целое число, не превосходящее x
ln(x)	Возвращает натуральный логарифм числа x
log(x)	Возвращает десятичный логарифм числа x
pow(x, y)	Возведение в степень: x^y
rad2deg (x)	Конвертирует угол, заданный в радианах, в градусы

$\sin(x)$	Возвращает значение функции $\sin(x)$, x задается в радианах
$\sinh(x)$	Возвращает значение функции $\operatorname{sh}(x)$ (гиперболический синус)
\sqrt{x}	Возвращает квадратный корень из числа x , т.е. \sqrt{x}
$\tan(x)$	Возвращает значение функции $\operatorname{tg}(x)$, x задается в радианах
$\tanh(x)$	Возвращает значение функции $\operatorname{th}(x)$ (гиперболический тангенс)

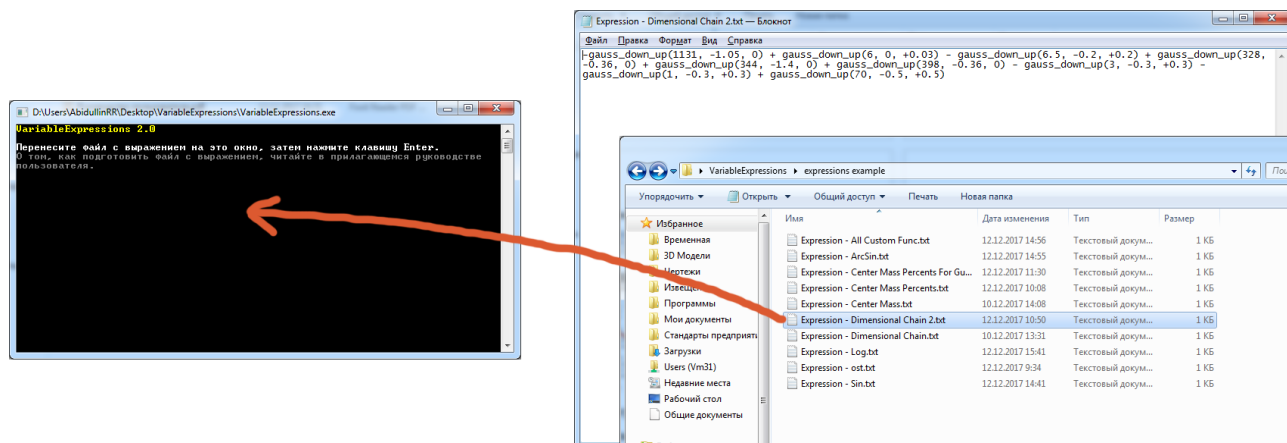
Комбинаторика

$\operatorname{fac}(x)$	Возвращает факториал (!) числа x ($x!$). Например, $5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$
$\operatorname{ncr}(n, k)$	Количество сочетаний из n по k (например, $\operatorname{ncr}(6, 2) = 15$)
$\operatorname{npr}(n, k)$	Количество размещений из n по k (например, $\operatorname{npr}(6, 2) = 30$)

Функции связывания переменных

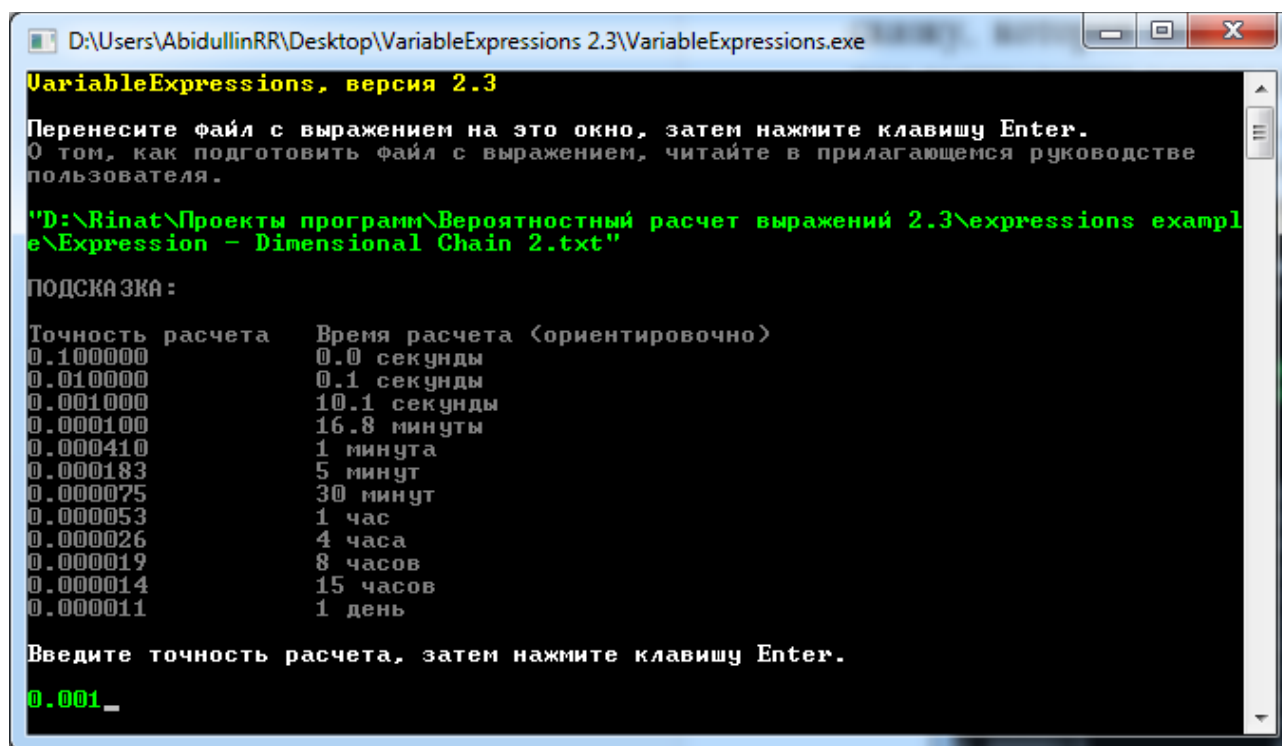
$\operatorname{link1}(x)$ (а также $\operatorname{link2}(x)$, $\operatorname{link3}(x) \dots$ $\operatorname{link100}(x)$)	Все одноименные функции link (например, $\operatorname{link75}(x)$) возвращают одно и то же значение, которое равно первому вычисленному значению x . Благодаря этому можно задать единое значение для переменной, входящей в состав сразу нескольких частей выражения (например, масса элемента при расчете центра масс в числителе и знаменателе).
--	--

Шаг 2. Открыть приложение и переместить файл с выражением.

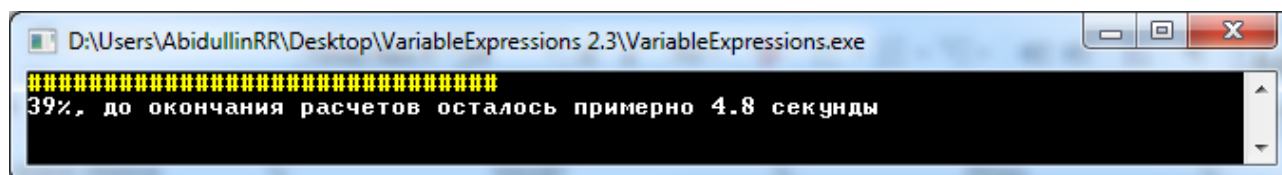


После перемещения файла на окно программы нажмите клавишу **Enter**.

Программа предложит ввести точность расчета выражения. Для многих инженерных задач достаточно точностей в диапазоне **0.1...0.001**. Вы увидите подсказку, которая показывает, сколько предположительно времени потребуется для вычисления результата в зависимости от точности расчета.

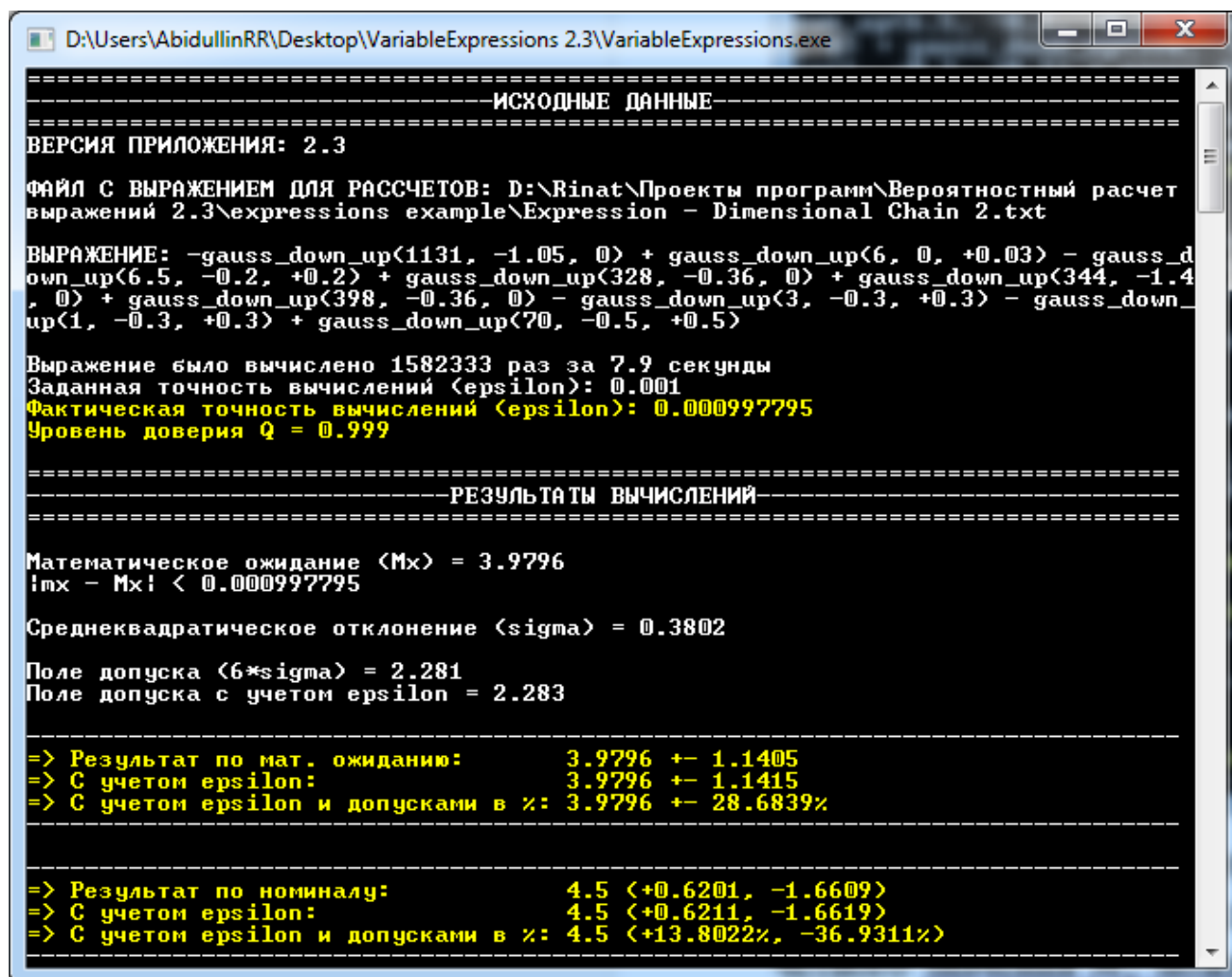


Введите точность расчета, которая вас устроит, и нажмите клавишу **Enter**.



Шаг 3. Дождаться результатов выполнения вычислений.

Пример вычислений приведен на рисунке ниже:



```
D:\Users\AbidullinRR\Desktop\VariableExpressions 2.3\VariableExpressions.exe

=====ИСХОДНЫЕ ДАННЫЕ=====
=====
ВЕРСИЯ ПРИЛОЖЕНИЯ: 2.3

ФАЙЛ С ВЫРАЖЕНИЕМ ДЛЯ РАССЧЕТОВ: D:\Rinat\Проекты программ\Вероятностный расчет
выражений 2.3\expressions example\Expression - Dimensional Chain 2.txt

ВЫРАЖЕНИЕ: -gauss_down_up<1131, -1.05, 0> + gauss_down_up<6, 0, +0.03> - gauss_d
own_up<6.5, -0.2, +0.2> + gauss_down_up<328, -0.36, 0> + gauss_down_up<344, -1.4
, 0> + gauss_down_up<398, -0.36, 0> - gauss_down_up<3, -0.3, +0.3> - gauss_down_
up<1, -0.3, +0.3> + gauss_down_up<70, -0.5, +0.5>

Выражение было вычислено 1582333 раз за 7.9 секунды
Заданная точность вычислений (epsilon): 0.001
Фактическая точность вычислений (epsilon): 0.000997795
Уровень доверия Q = 0.999

=====РЕЗУЛЬТАТЫ ВЫЧИСЛЕНИЙ=====
=====

Математическое ожидание <Mx> = 3.9796
|mx - Mx| < 0.000997795

Среднеквадратическое отклонение <sigma> = 0.3802
Поле допуска <6*sigma> = 2.281
Поле допуска с учетом epsilon = 2.283

=> Результат по мат. ожиданию: 3.9796 +- 1.1405
=> С учетом epsilon: 3.9796 +- 1.1415
=> С учетом epsilon и допусками в %: 3.9796 +- 28.6839%

=> Результат по номиналу: 4.5 (+0.6201, -1.6609)
=> С учетом epsilon: 4.5 (+0.6211, -1.6619)
=> С учетом epsilon и допусками в %: 4.5 (+13.8022%, -36.9311%)

=====
```

Результат работы программы – это два числа: математическое ожидание M_x и среднеквадратическое отклонение σ . За числовое значение поля допуска T расчетного значения M_x можно принять 6σ (вероятность выхода за пределы поля допуска составит 0,27%).

Математическое ожидание M_x определяется с заданной пользователем точностью ε , причем найденное программно значение M_x с уровнем доверия $Q = 0.9999$ отстоит от истинного значения m_x не более, чем на ε , т.е.:

$$|m_x - M_x| < \varepsilon$$

Это означает, что найденное поле допуска T необходимо расширить на величину 2ε :

$$T = 6\sigma + 2\varepsilon$$

На рисунке ниже результат расчетов, на который нужно ориентироваться, выделен:

```

D:\Users\AbidullinRR\Desktop\VariableExpressions 2.3\VariableExpressions.exe
=====ИСХОДНЫЕ ДАННЫЕ=====
ВЕРСИЯ ПРИЛОЖЕНИЯ: 2.3
ФАЙЛ С ВЫРАЖЕНИЕМ ДЛЯ РАСЧЕТОВ: D:\Rinat\Проекты программ\Вероятностный расчет
выражений 2.3\expressions example\Expression - Dimensional Chain 2.txt
ВЫРАЖЕНИЕ: -gauss_down_up<1131, -1.05, 0> + gauss_down_up<6, 0, +0.03> - gauss_d
own_up<6.5, -0.2, +0.2> + gauss_down_up<328, -0.36, 0> + gauss_down_up<344, -1.4
, 0> + gauss_down_up<398, -0.36, 0> - gauss_down_up<3, -0.3, +0.3> - gauss_down_
up<1, -0.3, +0.3> + gauss_down_up<70, -0.5, +0.5>
Выражение было вычислено 1582333 раз за 7.9 секунды
Заданная точность вычислений (epsilon): 0.001
Фактическая точность вычислений (epsilon): 0.000997795
Уровень доверия Q = 0.999
=====РЕЗУЛЬТАТЫ ВЫЧИСЛЕНИЙ=====
Математическое ожидание (Mx) = 3.9796
!mx - Mx! < 0.000997795
Среднеквадратическое отклонение (sigma) = 0.3802
Поле допуска (6*sigma) = 2.281
Поле допуска с учетом epsilon = 2.283

=> Результат по мат. ожиданию:      3.9796 +- 1.1405
=> С учетом epsilon:                3.9796 +- 1.1415
=> С учетом epsilon и допусками в %: 3.9796 +- 28.6839%

=> Результат по номиналу:           4.5 (<+0.6201, -1.6609>)
=> С учетом epsilon:                4.5 (<+0.6211, -1.6619>)
=> С учетом epsilon и допусками в %: 4.5 (<+13.8022%, -36.9311%)

```

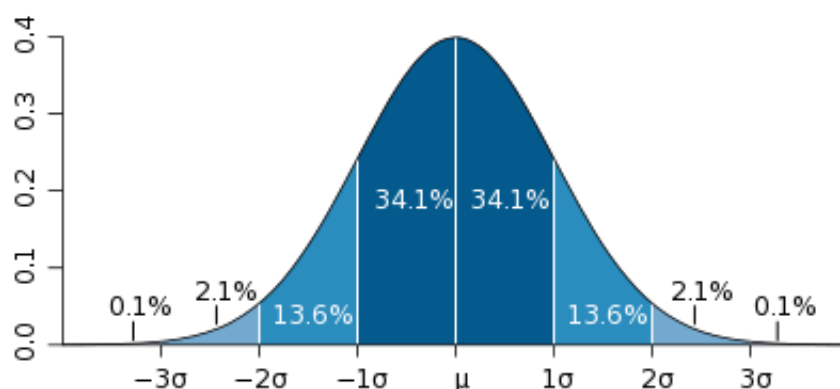


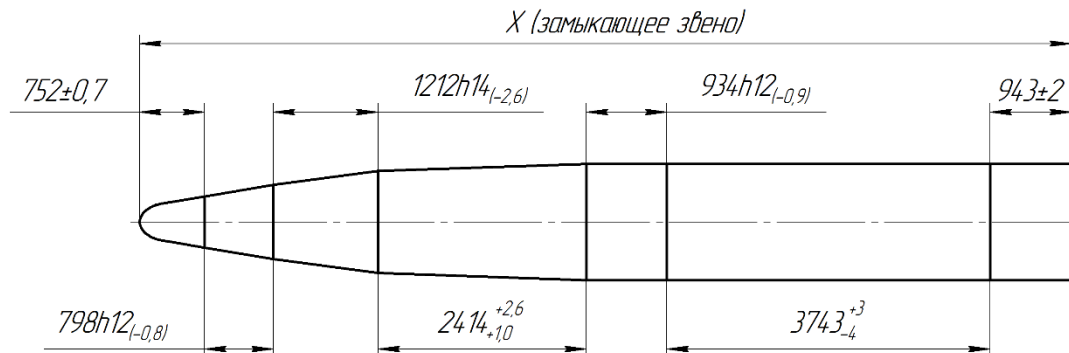
График плотности вероятности нормального распределения и процент попадания случайной величины на отрезки, равные среднеквадратическому отклонению.

Примеры решения задач

Задача 1. Размерная цепь

В программе результат расчета полностью соответствует вероятностному методу расчета размерных цепей.

Условие задачи: Необходимо вычислить суммарную длину изделия вероятностным методом.



Составим уравнение размерной цепи:

$$X = 752_{-0,7}^{+0,7} + 798_{-0,8} + 1212_{-2,6} + 2414_{-1}^{+2,6} + 934_{-0,9} + 3743_{-4}^{+3} + 943_{-2}^{+2}$$

Создадим текстовый файл «Расчет длины изделия.txt» и перепишем вышеприведенное уравнение в соответствии с правилами записи выражений для расчета в программе. Так как уравнение содержит величины с допусками, нам нужно использовать функции случайных распределений. В данном случае удобно воспользоваться функцией `gdu(номинал, нижнее_отклонение, верхнее_отклонение)`, которая возвращает случайное значение размера из заданного допуска в соответствии с нормальным законом распределения:

```
gdu(752, -0.7, +0.7) + gdu(798, -0.8, 0) + gdu(1212, -2.6, 0) +  
gdu(2414, -1, +2.6) + gdu(934, -0.9, 0) + gdu(3743, -4, +3) +  
gdu(943, -2, 2)
```

Откроем приложение и перенесем созданный файл с выражением на окно приложения. Введем точность расчета, которая нас устроит, например, 0.001. Результаты вычисления длины изделия приведены на рисунке ниже:

```

D:\Users\AbidullinRR\Desktop\VariableExpressions 3.0\VariableExpressions.exe
=====ИСХОДНЫЕ ДАННЫЕ=====
=====
ВЕРСИЯ ПРИЛОЖЕНИЯ: 2.3
ФАЙЛ С ВЫРАЖЕНИЕМ ДЛЯ РАССЧЕТОВ: D:\Users\AbidullinRR\Desktop\VariableExpressions 3.0\Примеры выражений\Расчет длины изделия.txt
ВЫРАЖЕНИЕ: gdu<752, -0.7, +0.7> + gdu<798, -0.8, 0> + gdu<1212, -2.6, 0> +
gdu<2414, -1, +2.6> + gdu<934, -0.9, 0> + gdu<3743, -4, +3> +
gdu<943, -2, 2>
Выражение было вычислено 27148641 раз за 1.6 минуты
Заданная точность вычислений (epsilon): 0.001
Фактическая точность вычислений (epsilon): 0.000991432
Уровень доверия Q = 0.999
=====РЕЗУЛЬТАТЫ ВЫЧИСЛЕНИЙ=====
=====
Математическое ожидание (Mx) = 10794.2
|mx - Mx| < 0.000991432
Среднеквадратическое отклонение (sigma) = 1.5647
Поле допуска (6*sigma) = 9.388
Поле допуска с учетом epsilon = 9.39
=====
=> Результат по мат. ожиданию:      10794.2 +- 4.694
=> С учетом epsilon:                10794.2 +- 4.695
=> С учетом epsilon и допусками в %: 10794.2 +- 0.0435%
=====
=> Результат по номиналу:           10796 (+2.8442, -6.5438)
=> С учетом epsilon:                10796 (+2.8452, -6.5448)
=> С учетом epsilon и допусками в %: 10796 (+0.0264%, -0.0606%)
=====

```

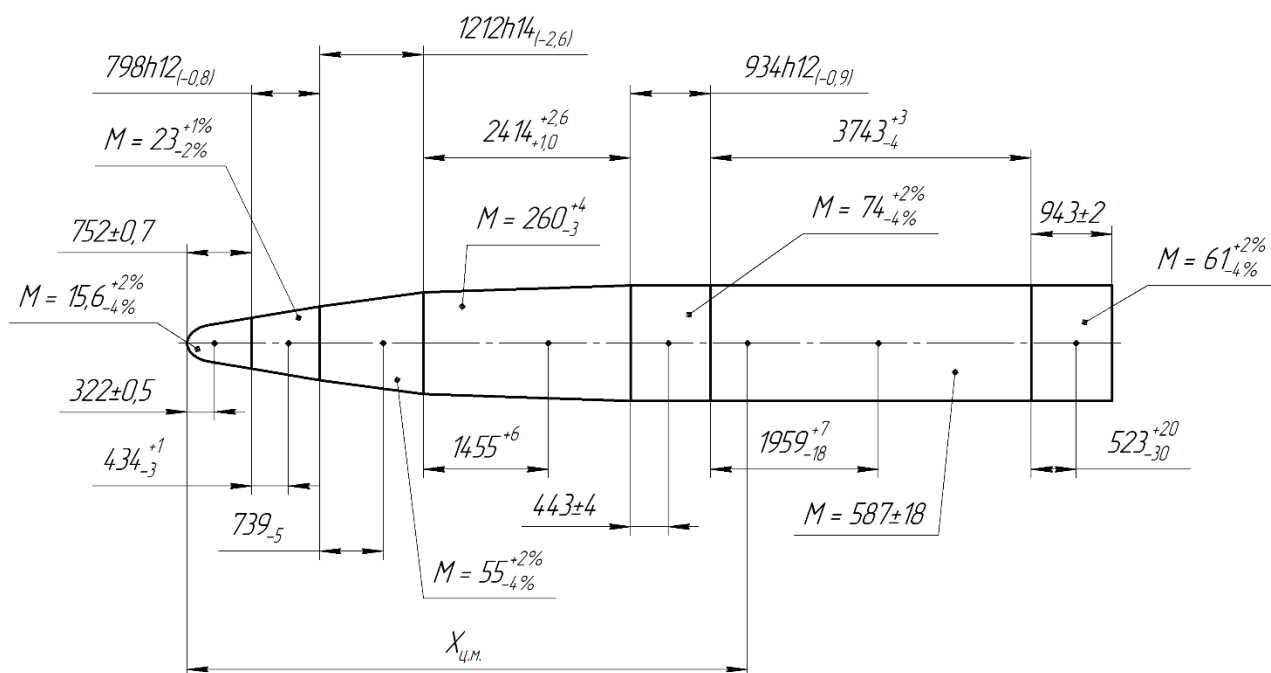
Нам необходимо выбрать решение с учетом точности вычислений (`epsilon`, которая для данной задачи принята `0.001`):

$$10796^{+2,825}_{-6,545} \text{ мм.}$$

Задача 2. Центр масс

Задача определения центра масс вероятностным способом особенно актуальна для использования этой программы, так как аналитического выражения для определения полей допусков для отношения двух вероятностей не существует.

Условие задачи: Необходимо вычислить положение центра масс изделия вероятностным методом.



Составим уравнение для определения центра масс изделия:

$$X_{ц.м.} = \frac{\sum M_i \times X_i}{\sum M_i}$$

$$X_{ц.м.} = \frac{15,6^{+2\%}_{-4\%} \times 322^{+0,5}_{-0,5} + 23^{+1\%}_{-2\%} \times (752^{+0,7}_{-0,7} + 434^{+1}_{-3}) + \dots}{15,6^{+2\%}_{-4\%} + 23^{+1\%}_{-2\%} + \dots}$$

Ввиду громоздкости получающегося уравнения для определения центра масс, перейдем сразу к составлению выражения для расчета в программе.

Создадим текстовый файл «Расчет центра масс.txt» и перепишем вышеприведенное уравнение в соответствии с правилами записи выражений для расчета в программе. В уравнении присутствуют величины с допусками, заданными как в процентном виде, так и в той же размерности, что и номинальный размер. Поэтому будем использовать функции `gpr(номинал, нижнее_отклонение_в_процентах, верхнее_отклонение_в_процентах)` и `gdu(номинал, ниж-`

нее_отклонение, верхнее_отклонение), которые возвращают случайные значения величин из заданных допусков в соответствии с нормальным законом распределения. Также мы использовали функции link (link1, link2 ...) для связывания одинаковых переменных (массы отсеков) в выражении:

```
(
  link1(gpp(15.6,-4,+2)) * gdu(322,-0.5,+0.5) +
  link2(gpp(23,-2,+1)) * (gdu(752,-0.7,+0.7) + gdu(434,-3,+1)) +
  link3(gpp(55,-4,+2)) * (gdu(752,-0.7,+0.7) + gdu(798,-0.8,0) +
    gdu(739,-5,0)) +
  link4(gdu(260,-3,+4)) * (gdu(752,-0.7,+0.7) + gdu(798,-0.8,0) +
    gdu(1212,-2.6,0) + gdu(1455,0,+6)) +
  link5(gpp(74,-4,+2)) * (gdu(752,-0.7,+0.7) + gdu(798,-0.8,0) +
    gdu(1212,-2.6,0) + gdu(2414,+1,+2.6) + gdu(443,-4,+4)) +
  link6(gdu(587,-18,+18)) * (gdu(752,-0.7,+0.7) + gdu(798,-0.8,0) +
    + gdu(1212,-2.6,0) + gdu(2414,+1,+2.6) + gdu(934,-0.9,0) +
    gdu(1959,-18,+7)) +
  link7(gpp(61,-4,+2)) * (gdu(752,-0.7,+0.7) + gdu(798,-0.8,0) +
    gdu(1212,-2.6,0) + gdu(2414,+1,+2.6) + gdu(934,-0.9,0) +
    gdu(3743,-4,+3) + gdu(523,-30,+20))
)
/
(
  link1(gpp(15.6,-4,+2)) +
  link2(gpp(23,-2,+1)) +
  link3(gpp(55,-4,+2)) +
  link4(gdu(260,-3,+4)) +
  link5(gpp(74,-4,+2)) +
  link6(gdu(587,-18,+18)) +
  link7(gpp(61,-4,+2))
)
```

Ограничимся точностью вычислений 0.01. Результаты вычисления центра масс изделия приведены на рисунке ниже:

```

D:\Users\AbidullinRR\Desktop\VariableExpressions 3.0\VariableExpressions.exe
=====ИСХОДНЫЕ ДАННЫЕ=====
ВЕРСИЯ ПРИЛОЖЕНИЯ: 2.3
ФАЙЛ С ВЫРАЖЕНИЕМ ДЛЯ РАССЧЕТОВ: D:\Users\AbidullinRR\Desktop\VariableExpressions 3.0\Примеры выражений\Расчет центра масс.txt
ВЫРАЖЕНИЕ: <
link1(gpp<15.6,-4,+2>) * gdu<322,-0.5,+0.5> +
link2(gpp<23,-2,+1>) * <gdu<752,-0.7,+0.7> + gdu<434,-3,+1>> +
link3(gpp<55,-4,+2>) * <gdu<752,-0.7,+0.7> + gdu<798,-0.8,0> +
gdu<739,-5,0>> +
link4(gdu<260,-3,+4>) * <gdu<752,-0.7,+0.7> + gdu<798,-0.8,0> +
gdu<1212,-2.6,0> + gdu<1455,0,+6>> +
link5(gpp<74,-4,+2>) * <gdu<752,-0.7,+0.7> + gdu<798,-0.8,0> +
gdu<1212,-2.6,0> + gdu<2414,+1,+2.6> + gdu<443,-4,+4>> +
link6(gdu<587,-18,+18>) * <gdu<752,-0.7,+0.7> + gdu<798,-0.8,0> +
gdu<1212,-2.6,0> + gdu<2414,+1,+2.6> + gdu<934,-0.9,0> +
gdu<1959,-18,+7>> +
link7(gpp<61,-4,+2>) * <gdu<752,-0.7,+0.7> + gdu<798,-0.8,0> +
gdu<1212,-2.6,0> + gdu<2414,+1,+2.6> + gdu<934,-0.9,0> +
gdu<3743,-4,+3> + gdu<523,-30,+20>>
>
/
<
link1(gpp<15.6,-4,+2>) +
link2(gpp<23,-2,+1>) +
link3(gpp<55,-4,+2>) +
link4(gdu<260,-3,+4>) +
link5(gpp<74,-4,+2>) +
link6(gdu<587,-18,+18>) +
link7(gpp<61,-4,+2>)
>

Выражение было вычислено 10798203 раз за 2.0 минуты
Заданная точность вычислений (epsilon): 0.01
Фактическая точность вычислений (epsilon): 0.00979809
Уровень доверия Q = 0.999

=====РЕЗУЛЬТАТЫ ВЫЧИСЛЕНИЙ=====

Математическое ожидание <Mx> = 6542.68
|mx - Mx| < 0.00979809

Среднеквадратическое отклонение <sigma> = 9.752
Поле допуска <6*sigma> = 58.513
Поле допуска с учетом epsilon = 58.533

=> Результат по мат. ожиданию: 6542.68 +- 29.257
=> С учетом epsilon: 6542.68 +- 29.266
=> С учетом epsilon и допусками в %: 6542.68 +- 0.447%

=====
=> Результат по номиналу: 6545.06 (+26.88, -31.634)
=> С учетом epsilon: 6545.06 (+26.889, -31.644)
=> С учетом epsilon и допусками в %: 6545.06 (+0.411%, -0.483%)
=====

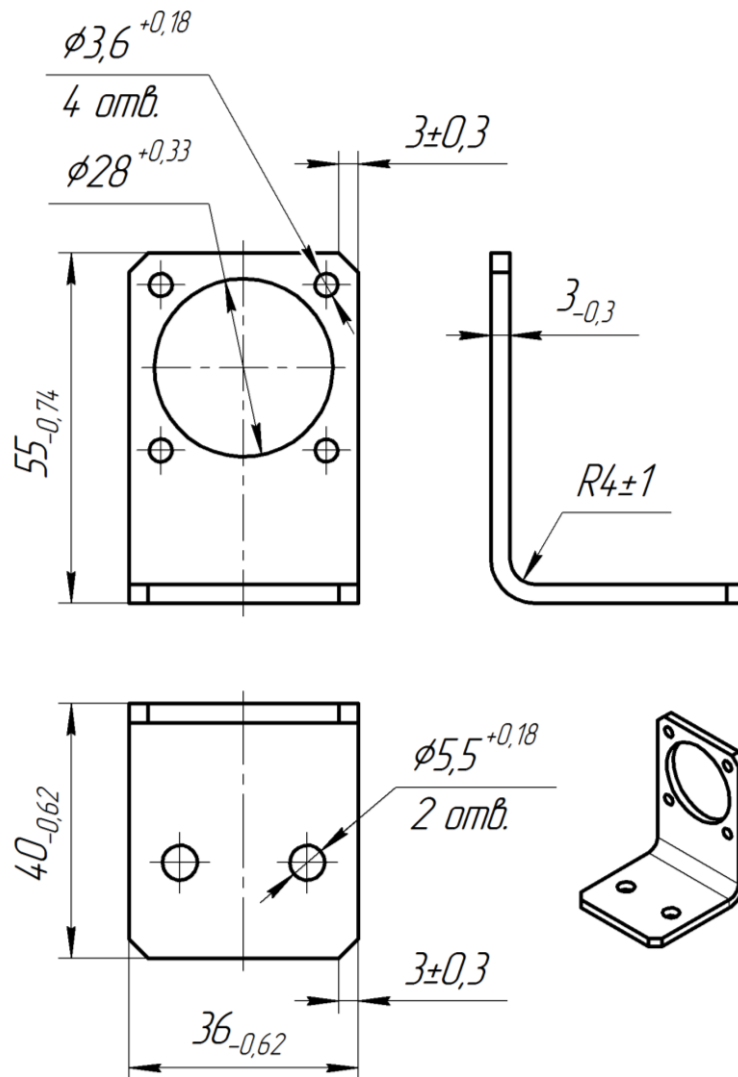
```

Нам необходимо выбрать решение с учетом точности вычислений (epsilon, которая для данной задачи принята 0.01):

$$6545,06^{+26,89}_{-31,64} \text{ мм.}$$

Задача 3. Масса кронштейна

Условие задачи: Необходимо вычислить отклонения по массе для кронштейна из стали с $\rho = 7800 \text{ кг/м}^3$.



Найдем длину развертки:

$$\begin{aligned}
 L_{\text{разв}} &= 55_{-0,74} + 40_{-0,62} - 2 \times (4_{-1}^{+1} + 3_{-0,3}) + \frac{\pi \left(4_{-1}^{+1} + \frac{3_{-0,3}}{2} \right)}{2} = \\
 &= 55_{-0,74} + 40_{-0,62} + \left(\frac{\pi}{2} - 2 \right) 4_{-1}^{+1} + \left(\frac{\pi}{4} - 2 \right) 3_{-0,3}
 \end{aligned}$$

Найдем площадь развертки:

$$S = L_{\text{разв}} \times 36_{-0,62} - (\text{сумма площадей фасок и отверстий})$$

Масса кронштейна может быть найдена как $M = \rho V = \rho S \times 3_{-0,3}$.

Не будем дальше расписывать формулу вычисления площади развертки, а сразу перейдем к составлению выражения для программы:

```
((
  gdu(55,-0.74,0) + gdu(40,-0.62,0) + (pi / 2 - 2) * gdu(4,-1,+1) +
  (pi / 4 - 2) * link1(gdu(3,-0.3,0))
) * gdu(36,-0.62,0)
-
(
  pi / 4 * gdu(5.5,0,+0.18)^2 + pi / 4 * gdu(5.5,0,+0.18)^2 +
  pi / 4 * gdu(3.6,0,+0.18)^2 + pi / 4 * gdu(3.6,0,+0.18)^2 +
  pi / 4 * gdu(3.6,0,+0.18)^2 + pi / 4 * gdu(3.6,0,+0.18)^2 +
  pi / 4 * gdu(28,0,+0.33)^2 +
  gdu(3,-0.3,+0.3) * gdu(3,-0.3,+0.3) / 2 +
  gdu(3,-0.3,+0.3) * gdu(3,-0.3,+0.3) / 2 +
  gdu(3,-0.3,+0.3) * gdu(3,-0.3,+0.3) / 2 +
  gdu(3,-0.3,+0.3) * gdu(3,-0.3,+0.3) / 2
))
*
link1(gdu(3,-0.3,0)) * 7800 / pow(1000,3)
```

Обратите внимание, что мы использовали функцию `link1(gdu(3,-0.3,0))` для связывания значений размера $Z_{-0,3}$, так как он входит одновременно как половина расстояния до нормальной линии гибки листа при расчете длины развертки и как толщина листа при определении объема.

Ограничимся точностью вычислений `0.0001`. Результаты вычисления массы кронштейна приведены на рисунке на следующей странице:

Нам необходимо выбрать решение с учетом точности вычислений (`epsilon`, которая для данной задачи принята `0.0001`):

$$0,0586_{-0,0071}^{-0,0013} = 0,0586_{-12,01\%}^{-2,23\%} \text{ кг.}$$

Проверим полученный результат. Для этого построим две модели в SolidWorks – одну по номинальным размерам, а вторую – по серединам полей допусков.

Результаты построений моделей и расчетов в программе сведены в таблицу:

	По номинальному размеру	По математическому ожиданию
VariableExpressions	$0,05862_{-0,00705}^{-0,00131} \text{ кг}$	$0,05444 \pm 0,00287 \text{ кг}$
SolidWorks	0,05862 кг	0,05443 кг

Результаты сходятся с большой точностью. Вычисления, выполненные в программе, верны.


```

D:\Users\AbidullinRR\Desktop\VariableExpressions 2.3\VariableExpressions.exe
=====ИСХОДНЫЕ ДАННЫЕ=====
ВЕРСИЯ ПРИЛОЖЕНИЯ: 2.3
ФАЙЛ С ВЫРАЖЕНИЕМ ДЛЯ РАССЧЕТОВ: D:\Rinat\Проекты программ\Вероятностный расчет
выражений 2.3\Примеры выражений\Расчет массы кронштейна.txt
ВЫРАЖЕНИЕ: <<
  gdu<55,-0.74,0> + gdu<40,-0.62,0> + (pi / 2 - 2) * gdu<4,-1,+1> +
  (pi / 4 - 2) * link1(gdu<3,-0.3,0>)
> * gdu<36,-0.62,0>
<
  pi / 4 * gdu<5.5,0,+0.18>^2 + pi / 4 * gdu<5.5,0,+0.18>^2 +
  pi / 4 * gdu<3.6,0,+0.18>^2 + pi / 4 * gdu<3.6,0,+0.18>^2 +
  pi / 4 * gdu<3.6,0,+0.18>^2 + pi / 4 * gdu<3.6,0,+0.18>^2 +
  pi / 4 * gdu<28,0,+0.33>^2 +
  gdu<3,-0.3,+0.3> * gdu<3,-0.3,+0.3> / 2 +
  gdu<3,-0.3,+0.3> * gdu<3,-0.3,+0.3> / 2 +
  gdu<3,-0.3,+0.3> * gdu<3,-0.3,+0.3> / 2 +
  gdu<3,-0.3,+0.3> * gdu<3,-0.3,+0.3> / 2
>>
*
link1(gdu<3,-0.3,0>) * 7800 / pow(1000,3)
Выражение было вычислено 5000 раз за 0.0 секунды
Заданная точность вычислений (epsilon): 0.0001
Фактическая точность вычислений (epsilon): 4.39832e-005
Уровень доверия Q = 0.999
=====РЕЗУЛЬТАТЫ ВЫЧИСЛЕНИЙ=====
Математическое ожидание <Mx> = 0.05444
|mx - Mx| < 4.39832e-005
Среднеквадратическое отклонение <sigma> = 0.00094
Поле допуска <6*sigma> = 0.00565
Поле допуска с учетом epsilon = 0.00574
=> Результат по мат. ожиданию: 0.05444 +- 0.00283
=> С учетом epsilon: 0.05444 +- 0.00287
=> С учетом epsilon и допусками в %: 0.05444 +- 5.27232%
=> Результат по номиналу: 0.05862 <-0.00136, -0.00701>
=> С учетом epsilon: 0.05862 <-0.00131, -0.00705>
=> С учетом epsilon и допусками в %: 0.05862 <-2.23482%, -12.0271%>

```