

# Калькулятор вычисления данных поверхности Surface Calculator

Инструкции пользователя

**ООО «Лаборатория Приезжева»**

**Copyright 2016, ООО «Лаборатория Приезжева». Все права защищены.**

Этот документ содержит конфиденциальную и служебную коммерческую тайну ООО «Лаборатория Приезжева» и не могут быть скопированы или сохранены в информационно-поисковой системы, переданной, используемой, распространять, переводить или передавать в любой форме или любыми средствами, электронными или механическими, полностью или частично без письменного разрешения владельца авторских прав.

**Старт:**

**Surface attributes->**

**Surface Calculator**

**Правила работы калькулятора для вычисления данных поверхности:**

- 1. Определяемая процедура вычисляет каждый узел в результирующем свойстве поверхности.**
- 2. Имеется несколько предопределяемых переменных.**
  - **I, J** – индекс для вычисления текущего поверхностного узла.
  - **X, Y** – координаты узла для текущего поверхностного узла.
  - **Z** – глубина или время для текущего узла (индекс атрибута поверхности =0).
  - **nI, nJ** – размер скользящего окна со смежными узлами для вычислений вокруг текущего поверхностного узла.
  - **Prop[]** – массив всех атрибутов поверхности для текущего поверхностного узла.
  - **Surf** – массив со значениями в скользящем окне. Размер массива равен **Surf [nI,nJ][nK]**, где **nK** соответствует номеру атрибута поверхности.
  - Переменная с **"PropName"** позволяет получить значения этого атрибута для текущего узла.
- 3. Создается только один новый атрибут с результатом (последняя переменная присвоения).**
- 4. Имя нового атрибута с результатом будет определено переменной с '=' (**TokenKind.ASSIGN**). Например, если применяем строку процедуры: *surfResult=100;* это создает новое свойство с именем "surfResult" со значением константы =100.**
- 5. Если мы используем несколько строк для процедуры, то только последняя **ASSING** будет определять результат процедуры. Например, если мы применяем строку процедуры:**  
*d=(X-1000)\*(x-1000)+(y-3000)\*(y-3000)*  
*surfResult=sqrt(d);*

- в результате будет только одно новое свойство с именем "surfResult".
6. Для этой процедуры может использоваться несколько предопределяемых функций sin(), cos(), abs(), pow(,), atan2(,), sqrt(), exp(), log(), sign(), rand(), iif(,,).
  7. Дополнительно могут использоваться все функции подобные Math.Sqrt()...
  8. С# конструкции for(;;) {}, if() else, и все подобные другие могут быть использованы.
  9. В примере ниже приводится пример вычисления атрибута обнаружения разломов по данным поверхности:

```
//Fault detection attribute
double sum=0;
int n=0;
int k=0; //used property
for(int i=0; i< nI; i++)
    for(int j=0; j< nJ; j++)
    {
        int ii =nI/2+1;
        int jj =nJ/2+1;
        if(i==ii && j==jj) continue;
        var diff=abs(Surf[i,j][k]-Surf[ii,jj][k]);
        var norm=abs(Surf[i,j][k]+Surf[ii,jj][k]);
        if(norm==0)norm=1000000; // protect to division to
zero
        var raz=diff/norm;
        sum+=raz;
        n++;
    }
faultDetect=sum/n;
```

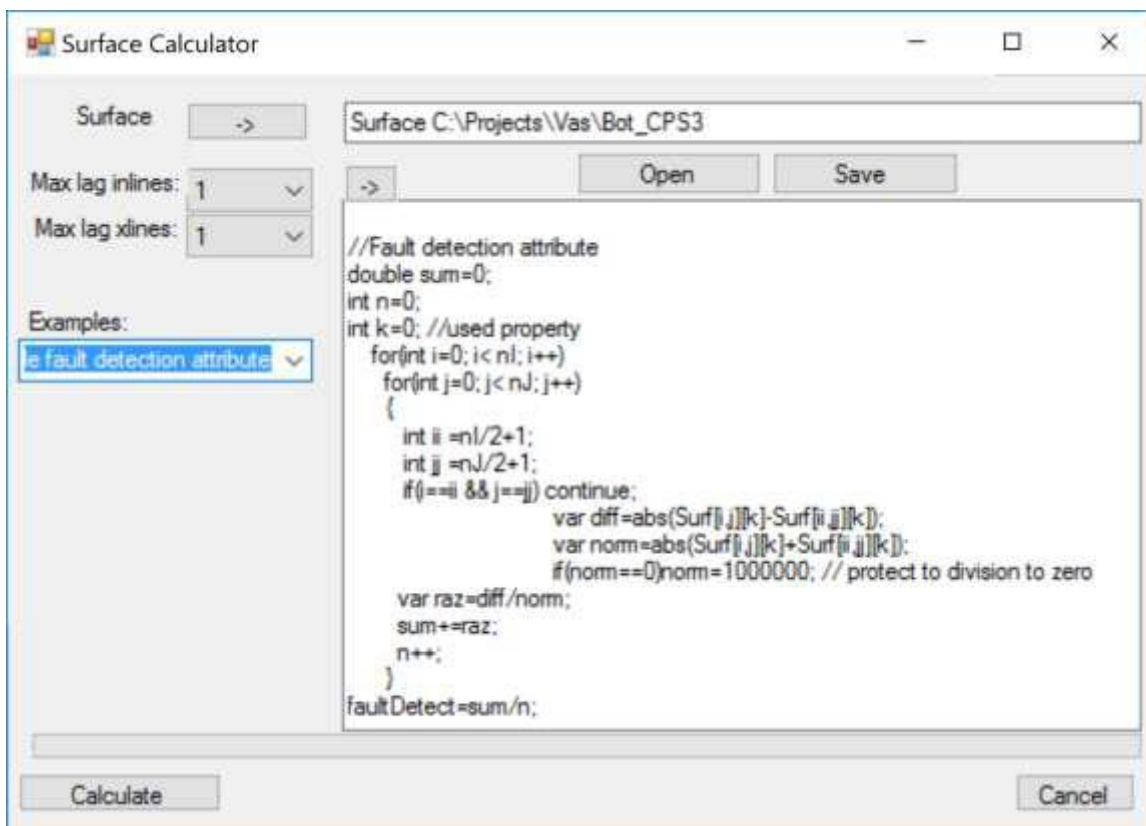


Рис. 1: Калькулятор вычисления данных поверхности