

Пакет "ГеоПоиск 8"

Калькулятор в примерах

Приложение к общей инструкции ГеоПоиск 8 (www.geopoisk.com)

Общие сведения	3
Загрузка и выполнение формулы в Калькуляторе	4
Настройка параметров выполнения формулы	4
Выполнение формулы	6
Пакетное выполнение формул	6
Создание новой формулы	7
Формула с кривыми	9
Инструкция условия	9
Формула с параметрами пропластков	10
Переменные	11
Печать, протокол	12
Несколько формул в одном файле (пакетный расчет)	13
Встроенные функции	14
Математические функции	14
Математические константы	15
Функция для получения текущей глубины	15
Функция для получения абсолютной глубины	15
Функции индексирования и работы с глубиной по кривой	15
Функции индексирования и работы с глубиной по пропласткам	16
Функции использования стратиграфических пластов	16
Однопроходные функции	18
Работа с палетками	19
Функции для работы с двумерными палетками	20
Функции для работы с трехмерными палетками	20
Функции для работы с регулярными палетками	21
Функция для работы с палетками, заданными узловыми точками	23
Определение компонент пласта по «треугольной палетке»:	23
Пример. Палетка определения коэффициента пористости	25
Доступ к параметрам скважины	26
Получение значения числового параметра	27
Получение значения текстового параметра	27
Получение значений числового интервала	27
Получение значений набора числовых интервалов	28
Получение разбор последовательности диапазонов глубин со значениями	29
Пример. Совместное использование палеток и параметров скважины	29
Доступ к данным керна и описаний интервалов скважины	30
Функции для работы с лабораторными исследованиями керна	30
Функции для работы с описаниями интервалов	31
Пример. Печать данных по керну и описаниям	31
Построение моделей	32
Пример. Модель по кривым	32
Пример. Модель по пропласткам с учетом групп пластов	34
Функции для решения систем уравнений	37
Приложение	40
Основные конструкции языка ГеоСи	40

Управление вычислением формулы из ГеоСи	41
Заготовки формул в Калькуляторе	43
Встроенные функции и процедуры (алфавитный перечень)	49

Общие сведения

Калькулятор (Int32, Интерпретатор формул ГеоСи) – это интегрированная среда программирования геофизических программ (формул) на языке ГеоСи. Поддерживает смешанные вычисления с использованием кривых и параметров пропластков (ГИС), потоковую обработку пакета формул на нескольких скважинах, или на нескольких скважинах в границах выбранного пласта. Интегрирован с Планшетом.

The screenshot shows the main interface of the 'Геофизический Каль' software. Several components are highlighted with red boxes and arrows:

- Окно редактирования формулы**: Points to the top-left text area containing input data like `тп = 45; // тем` and `Rф20 = 1.8; // УЭС раствора при 20 градусах Цельсия (Ом)`.
- Окно констант**: Points to the top-right table with columns 'константа', 'значение', and 'автор'.
- Параметры скважины**: Points to the central table with columns 'Код', 'Ключ', 'Значение', 'кривая', 'ЗК', 'ЗР', and 'количес'.
- Окно параметров выполнения формулы**: Points to the left sidebar with fields for 'месторождение', 'скважина/куст', 'интервал', 'кровля (ЗК)', and 'подошва (ЗР)'. A checkbox 'показывать константы' is also visible.
- Доступные методы ГИС и кривые в выбранной скважине**: Points to the 'Таблица ГИС' section in the central table.
- Листинг результата выполнения формулы**: Points to the right sidebar showing calculation steps and results, such as `Шаг 2: Палетка рис.73 (Шл)` and `tn = 45 e [40, 65]`.

Программы на ГеоСи можно выполнять не только из Калькулятора, но и непосредственно из Планшета (команда меню **Технология/Интерпретатор формул (ГеоСи)/Расчет формулы**):

The screenshot shows the 'Параметры для формулы' dialog box. Several components are highlighted with red boxes and arrows:

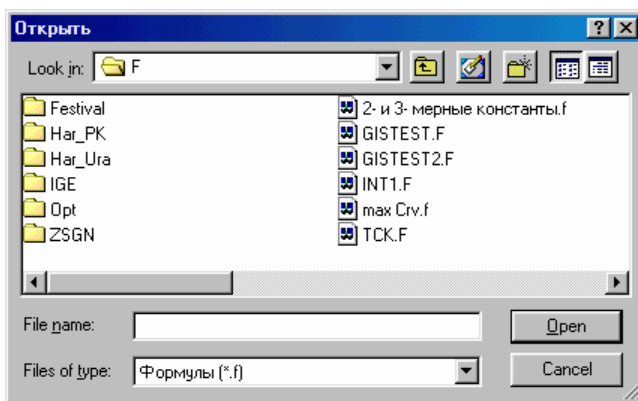
- Выбор скважины**: Points to the 'Скважина' dropdown menu.
- Диапазон глубин и шаг выполнения формулы**: Points to the 'Интервал глубин обработки' section, including 'Кровля (м)', 'Подошва (м)', and 'Шаг каротажа (м)'.
- Генератор простой вычислительной формулы**: Points to the 'Преобразовать' button.
- Листинг результата выполнения формулы**: Points to the 'Формула' text area showing the generated code: `formula Planshet((&'DGK,'GK) { 'DGK = ((GK)-1.2) / (12.4-1.2); }`.
- Окно редактирования формулы**: Points to the bottom-right text area showing compilation and execution details, such as `==== компиляция ====` and `==== выполнение ====`.

ЗАМЕЧАНИЕ: исправление формулы, сделанное в Планшете, не сохраняется

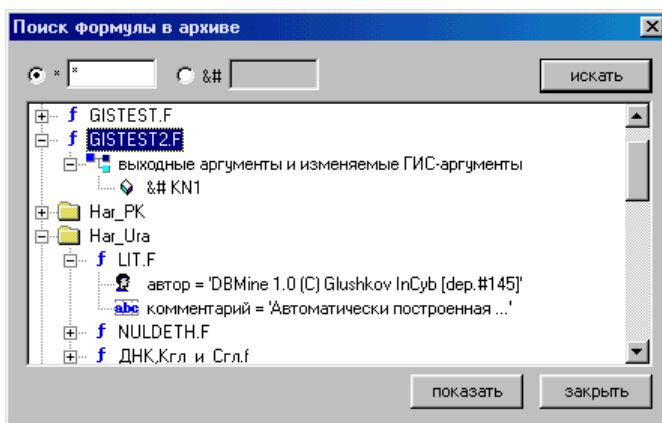
Команды меню **Технология/Интерпретатор формул (ГеоСи)** позволяют генерировать простые виды формул (вычисления кривых по другим кривым, вычисления параметров пропластков по другим параметрам ГИС, вычисления с условиями, формулы по палеткам, формулы снятия отсчетов). Кроме этого меню в ПЛАНШЕТЕ есть много других команд, создающих и позволяющих выполнить формулы на ГеоСи. Подробности в общей инструкции ГеоПоиск 8.

Загрузка и выполнение формулы в Калькуляторе

1-й способ. Команда **Файл/Открыть** для выбора файла с формулой (*.f) с помощью стандартного диалога «Открыть файл»

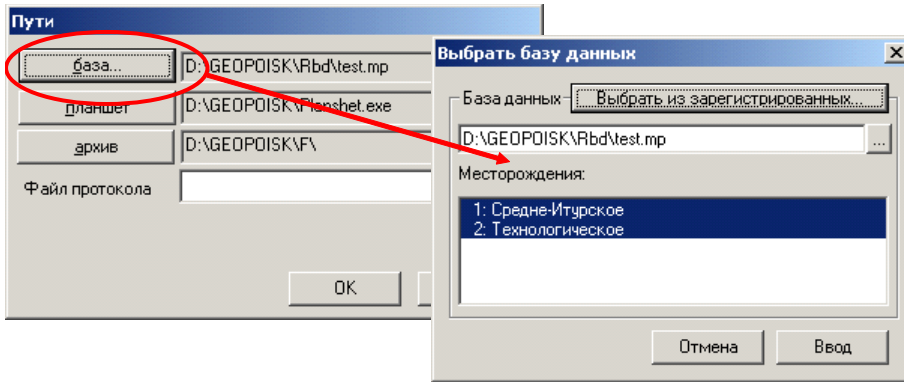


2-й способ. Команда **Файл/Выбор формулы из архива** для выбора формулы с использованием внутренней информации (например, входным или выходным параметрам). Архив традиционно расположен в каталоге *GeoPoisk/F*.

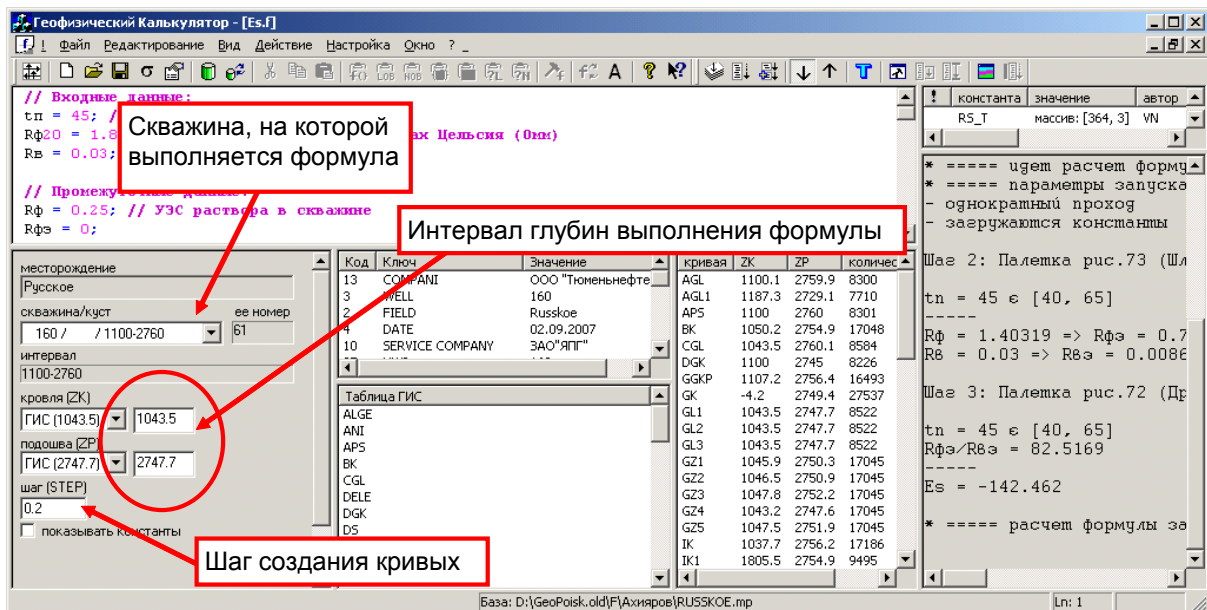


Настройка параметров выполнения формулы

Шаг 1. Выбор рабочей базы данных. Команда **Настройка/Пути**. В появившемся диалоге по кнопке «База» выберите рабочую базу данных.



Шаг 2. Выбор скважины из списка скважин рабочей базы



Шаг 3. Установка интервала и шага для создания кривых.

Кровлю и подошву интервала выполнения формулы можно «взять» из гис (списки слева от интервала формулы) или из кривой (выберите кривую в списке кривых скважины; далее, контекстное меню по правой кнопке мышки, команда «Установить параметры расчета по данной кривой»).

Дополнительно.

Автоматический запуск программы Планшет. Установка режима **Действие/Запустить ли Планшет** позволяет автоматически запускать планшет настроенный на базу и скважину формулы с кривыми и методами гис, участвующими в формуле.

Выбор шрифта окна редактирования формулы. Команда **Настройка/Шрифт/редактора программы**.

ЗАМЕЧАНИЕ: При копировании текста из/в окно редактирования формулы правильно установите кодировку (меню **Настройка/Шрифт/кодировка**)

Выполнение формулы

Команда **Действие/Запустить формулу**

Листинг выполнения формулы. Результаты выполнения формулы см. В листинге. В случае успешного выполнения формулу текст листинга будет приблизительно следующим:

```
-- обрабатывается БД 'C:\GEOPOISK\Rbd\TEST.MP'
- читается метод ГИС 'KC1' (101 пропластков)
* ===== идет расчет формулы 'f' =====
* ===== параметры запуска: Кровля(ZK) =2616.4, Подошва(ZP) =2975, Шаг
(STEP) =0.2 =====
- расчет по кривой
* ===== расчет формулы завершен =====
- записывается кривая 'KC1' (1794 точек)
```

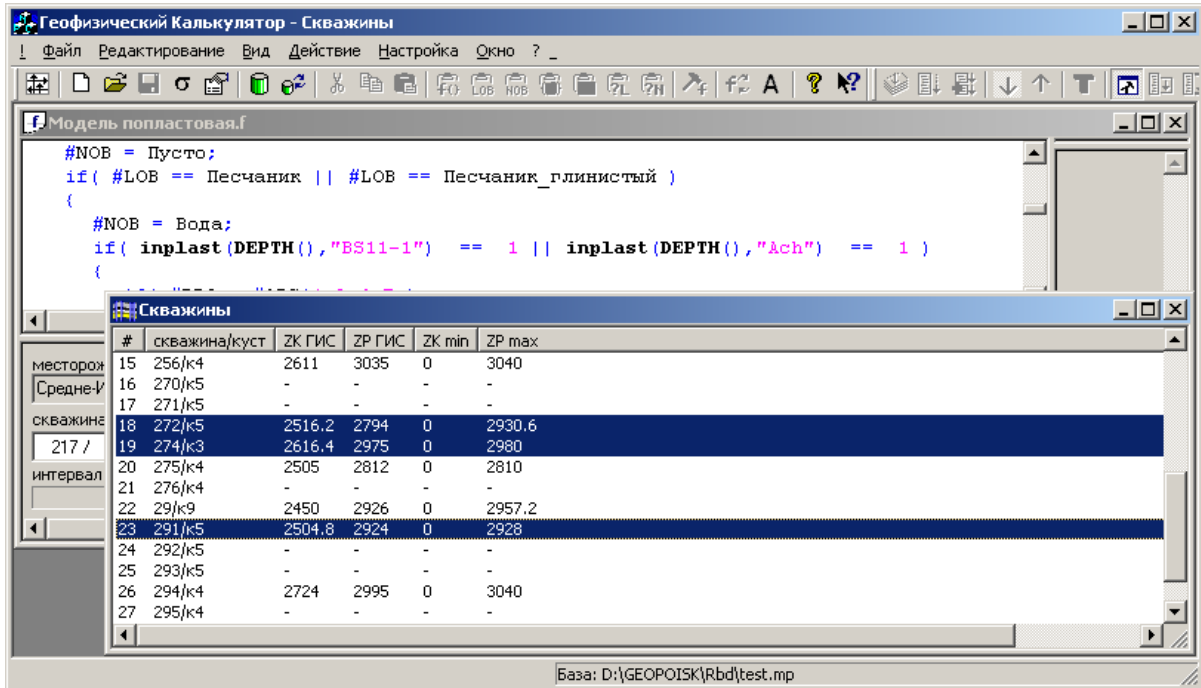
Красным цветом выделена информация об успешном выполнении формулы. Если какой-то входной аргумент (кривая или метод гис) отсутствует в скважине, то текст в листинге будет приблизительно следующим:

```
-- обрабатывается БД 'C:\GEOPOISK\Rbd\TEST.MP'
отсутствует ГИС-метод для аргумента 'KC1'
```

Ошибки выполнения. Если в синтаксисе формулы допущена ошибка, то компилятор сформирует список ошибок в окне «Сообщения». Двойное нажатие левой кнопкой мышки на строку с ошибкой автоматически покажет место ошибки в окне редактора формулы.

Пакетное выполнение формул

Пакет по скважинам. Для выполнения формулы в пакете для списка скважин вызовите окно «скважины» (команда **Вид/Скважины**) и выберите из списка мышкой при нажатых кнопках «**Shift**» или «**Ctrl**» скважины, для которых хотите выполнить формулу.



Далее, выберите окно с формулой и выполните команду из меню **Действие/Групповая обработка скважин**. Команда **расчет В СВОИХ ГРАНИЦАХ** автоматически устанавливает интервал расчета каждой скважины по кровле, подошве ГИС. Команда **расчет В ЗАДАННОМ ИНТЕРВАДЛЕ** всегда использует интервал установленный в окне параметров формулы.

Пакет по скважинам с выбором пластов. В окне «Пласты» (команда **Вид/Пласты**) выберите из списка нужные пласты. Выберите окно с формулой. Команда **Действие/Групповая обработка скважин/расчет по ПЛАСТАМ** позволяет выполнить формулу для всех скважин месторождения, но только в интервале глубин выбранных пластов.

Другой способ выполнения формулы только по выбранным пластам либо использование разных алгоритмов расчета в зависимости от кода пласта подразумевает программирование условий на ГеоСи. Он описан дальше.

Создание новой формулы

Команда **Файл/Создать формулу**

Создание шаблона формулы – команда **Редактирование/Вставить заготовку/Пустая формула**

```
// так выглядит шаблон формулы
formula New( ) ( ) // заголовок формулы
{
/*Тело формулы*/
/*
инструкция1;
инструкция2;
...
инструкцияN;
```

```
*/
}
```

Синим цветом выделяются ключевые слова. Регистр букв (большие, маленькие) имеет значение.

ЗАМЕЧАНИЕ: В тексте программы на ГеоСи можно использовать комментарии. Комментарии до конца строки отделяются символами «//», внутри строки выделяются символами «/*» – в начале и «*/» – в конце комментария.

Формула состоит из заголовка и тела формулы, внутри фигурных скобок. Структура заголовка следующая:

formula – обязательное ключевое слово

New – имя формулы. Любое название русскими или английскими буквами и цифрами, без пробелов (можно использовать знак подчеркивания «_»).

() – первые скобки содержат список выходных аргументов (кривые, создаваемых при выполнении формулы)

() – вторые скобки содержат список входных аргументов (кривые и методы гис, использующиеся в расчете)

Перед именем кривых в ГеоСи ставится символ «*», перед именами параметров пропластков (методов или колонок таблицы интерпретации ГИС) – «#».

{ ... } – тело формулы с инструкциями. После каждой инструкции обязательно ставится точка с запятой «;»

ЗАМЕЧАНИЕ: Принцип выполнения формул на ГеоСи следующий:

1. Если выходные или изменяемые аргументы – кривые, то формула выполняется как цикл от кровли до подошвы интервала расчета (или от подошвы до кровли в зависимости от направления расчета) с шагом, заданным в параметрах расчета. На каждой итерации последовательно выполняются команды внутри тела формулы. Значения входных кривых и параметров ГИС пропластков берутся на соответствующей глубине.

2. Если изменяемые параметры – ГИС пропластков, то формула выполняется для всех пропластков внутри интервала расчета, пропласток рассматривается как единое целое, в качестве значений кривых используются автоматически снятые отсчеты в текущем пропластке.

В ГеоСи используется «свободная запись программ». Это означает, что смысл программы зависит только от порядка слов и знаков и не зависит от форматирования текста программы. Например, следующие программы языка ГеоСи

```
formula ЭкспортКривыхВLasФайл ( ) ( *BK, *GK )
{
    SaveAsLAS (*BK [], *GK []);
}
```

```
formula
    ЭкспортКривыхВLasФайл
        (
            (*BK, *GK)
```

```

{
    SaveAsLAS(*BK[], *GK[]);
}

```

```
formula ЭкспортКривыхВЛасФайл() (*BK, *GK) {SaveAsLAS(*BK[], *GK[]);}
```

работают совершенно одинаково.

Формула с кривыми

```
formula f1(*RPI) (*CILD) // пересчет мСм -> Омм
{
    *RPI=1168.18/pow(*CILD,1.06);
}

```

***CILD** – входная кривая

***RPI** – расчетная кривая

Инструкция «***RPI=1168.18/pow(*CILD,1.06);**» значит **RPI = 1168.18/CILD^{1.06}**

pow – встроенная функция степени, поддерживается также запись **(*CILD)^1.06**. Список встроенных функций см. в приложении

```
formula f2(*RGR, *RVP) (*APS, &*KVO, &*EPHI, &*RPO) // Кв и Кнг
{
    if (*EPHI<=0)*KPO=3.0;
    if (*RPO<=0.0) *RPO=0.1;
    *RVP=pow(10, (4.24-2.1164*lg(*EPHI)))*0.07;
    *RGR=10.8-(5.36*(*APS));
}

```

Входных и выходных аргументов может быть несколько, тогда они отделяются запятыми. Символ «&» перед именем кривой значит, что эта кривая может меняться в формуле (например, **&*EPHI, &*RPO**). Отличие выходной кривой (описанной в первых скобках) от изменяемой (описанной во вторых скобках с значком «&») – в том, что в первом случае если кривая с таким именем уже есть в скважине, она будет удалена и заменена на новую, а во втором случае, в кривой изменятся только значения устанавливаемые в формуле, например, кривая останется неизменной за пределами интервала расчета.

НОВОЕ: Для вставки заготовки формулы можно использовать меню **Редактирование/Вставить заготовку/Встроенные функции:** formula есть среди «встроенных конструкций языка»

Инструкция условия

```

if(<условие>) // если
{
    // условие выполняется
    инструкция1;
    инструкция2;
    ...
}
else // иначе
{
    // условие не выполняется
    инструкция1;
    инструкция2;
    ...
}

```

```

}

//ПРИМЕР////////////////////////////////////
if(*APS<0.3)
{
    *RGR=11.4;
    *LIT=1;
}
else
{
    *RGR=10.8-(5.36*( *APS));
    *LIT=2;
}

```

Если внутри фигурных скобок – только одна инструкция, то фигурные скобки можно опустить

```

if(<условие>) инструкция1;
else инструкция2;

//ПРИМЕР////////////////////////////////////
if(*APS<0.3) *RGR=11.4;
else *RGR=10.8-(5.36*( *APS));

```

В **<условии>** могут использоваться отношения сравнения:

<	Меньше	if(*A<1) *B=0;
>	Больше	if(*A>1) *B=0;
>=	Больше или равно	if(*A>=1) *B=0;
<=	Меньше или равно	if(*A<=1) *B=0;
==	Равно	if(*A==1) *B=0;
!=	Не равно	if(*A!=1) *B=0;

В **<условии>** одновременно могут использоваться несколько логически связанных и выделенных скобками условий:

&&	И	If(*A<1 && *A>0) *B=0;
	ИЛИ	If(*A<1 *A>10) *B=0;
!	НЕ	if(!(*A==1)) *B=0;

НОВОЕ: Для вставки заготовки формулы можно использовать меню **Редактирование/Вставить заготовку/Встроенные функции**: if есть среди «встроенных конструкций языка». Могут пригодиться и другие команды меню **Редактирование/Вставить заготовку: Формула по группам пластов, Условия для литологии и Условия для насыщения**.

Формула с параметрами пропластков

Параметры ГИС пропластков описываются *только во вторых* скобках заголовка формулы. Перед именем параметра пропластка ставится значок «#». (В Планшете и Проводнике БД именам параметров пропластков предшествует префикс «гис», поэтому нужно учитывать, что «гис.имя» в языке ГеоСи соответствует «#имя».)

```

formula Подготовка_RIK() (&#RIK, *RIK, #NAS)
{
    if(#NAS>0) #RIK = *RIK;
    else #RIK = 0;
}

```

ЗАМЕЧАНИЕ: В одной формуле нельзя одновременно изменять кривые и параметры пропластков. Но среди входных аргументов формулы могут быть одновременно и параметры пропластков, и кривые, независимо от выходных аргументов формулы.

Переменные

GeoСи позволяет хранить числа в переменных. Переменные объявляются в момент первого присвоения им значений:

```
formula New() (*APS)
{
    Aps_min=100; // переменная
    Aps_max=0; // переменная
    if( Aps_min>*APS ) Aps_min=*APS;
    if( Aps_max<*APS ) Aps_max=*APS;
}
```

Если нужно, чтобы значение переменной сохранялись при выходе из цикла (формулы), переменную нужно описать перед первой формулой:

```
Aps_min=100; // переменная
Aps_max=0; // переменная
formula New() (*APS)
{
    if( Aps_min>*APS ) Aps_min=*APS;
    if( Aps_max<*APS ) Aps_max=*APS;
}
```

Такие «внешние» переменные называются глобальными, а переменные, описанные внутри формулы – локальными. Глобальные переменные можно использовать внутри формул так же, как и локальные: присваивать им значения, подставлять в условия, передавать в функции и т.п.

Можно сделать так, чтобы пользователь сам вводил значение переменной при каждом выполнении формулы. Для этого используется встроенная функция

input(<описание>, <Переменная>, <значение по умолчанию>)

```
// *ZKA - кривая абсолютной глубины
formula New(*KVO) (*ZKA,*EPHI,*APS)
{
    input( "порог абсолютной глубины", Абс_глубина, 2500);
    if( *ZPA<Абс_глубина )
    {
        if((*EPHI<12.5)&&(*APS<0.3)) *KVO=10^((-0.2156*(*APS))+1.94);
        else *KVO=sqrt((-209.44*(*EPHI))+4576.92
            -((-396.28*(*EPHI))+8722.5)*lg(*APS));
    }
    if( *ZKA>=Абс_глубина )
    {
        if ((*EPHI<12.5)&&(*APS<0.3)) *KVO=10^((-0.2156*(*APS))+1.94);
        else *KVO=sqrt((-325.21*(*EPHI))+6628.13
            -((155.4*(*EPHI))+2209.04)*lg(*APS));
    }
}
```

Перед выполнением формулы **New** появится диалог ввода значения переменной **Абс_глубина**

Если в формуле используются коды литологии или насыщения, связанные со словарем базы, то соответствующие переменные можно создать автоматически с помощью команд **Редактирование/Вставить заготовку/Литология** и **Редактирование/Вставить заготовку/Насыщение**

```
// ===== ЛОВ: обобщенный литиндекс =====
Песчаник = 3;
Песчаник_глинистый = 4;
Песчаник_уплотненный = 5;
// =====
formula f2(*RPO) (#RT,*RPI,#ЛОВ) {
    *RPO=#RT;
    if (#ЛОВ==Песчаник
        || #ЛОВ==Песчаник_глинистый
        || #ЛОВ==Песчаник_уплотненный)
        *RPO=*RPI;
}
```

НОВОЕ: Для вставки значений кодов любых данных, описанных в словаре базы, можно использовать меню **Редактирование/Вставить заготовку/Из словаря:** появится диалог, в котором можно выбрать раздел словаря и отметить нужные параметры. Раздел «ЛОВ» содержит коды литотипов, а раздел «NOB» – коды типов насыщения.

Печать, протокол

В процессе выполнения формулы в окно листинга или в отдельный файл можно вывести любую текстовую информацию о данных, использующихся в формуле.

```
bTitle = 0;
formula New () (#APS,#RT)
{
    if(bTitle==0)
    {
        // первая строка в файле protocol.txt
        protocol("WELL,ZK,ZP,APS,RT"); bTitle=1;
    }
    // на каждой итерации добавляется строка в файл protocol.txt
    protocol(WELL(),",",",",#ZK,",",",#ZP,",",",#APS,",",",#RT);
}
```

Результат - в файле protocol.txt:

```
WELL,ZK,ZP,APS,RT
274,2616.4,2653.2,0.081,5.5
274,2653.2,2654.2,0.531,4.4
274,2654.2,2661.8,0.093,6.1
274,2661.8,2667.2,0.849,4.4
274,2667.2,2669,0.332,11.4
274,2669,2670.8,0.721,4.3
274,2670.8,2672.2,0.165,3.6
274,2672.2,2673.2,0.506,5.4
274,2673.2,2674,0.192,5.3
274,2674,2675.4,0.564,3.6
274,2675.4,2683.2,0.034,3.3
```

По умолчанию файл протокола сохраняется в той папке, где лежит файл базы (mp или gr8). Другой путь можно задать в диалоге, который появляется по команде **Настройка/Пути**.

НОВОЕ: Имя файла протокола (с путем) можно задать прямо в тексте формулы, используя встроенную функцию **protocol_path**. Эта функция вместе с функцией очистки файла протокола **empty_protocol** и функцией проверки, пустой ли протокол **is_protocol_empty**, помогает использованию ГеоСи в качестве простого средства формирования текстовых заключений.

Печать в листинг (в правой части окна Калькулятора):

```
formula Объемная_влажность () (&#W, #RP)
{
  if( #RP <= 0 )
  {
    #W = NOVAL();
    print(#ZK, "-", #ZP, " :#RP=", #RP);
  }
  else
  {
    #W = 10^((3.2-log10(#RP))/2.17)/100;
    if(#W<0) #W=0;
    if(#W>1) #W=1;
  }
}
```

Результат в листинге:

```
* БД: 'C:\GEOPOISK\Rbd\TEST.MP'
- читается метод ГИС 'W' (101 пропластков)
- читается метод ГИС 'RP' (101 пропластков)
* ===== идет расчет формулы 'Объемная_влажность' =====
* ===== параметры запуска: Кровля(ZK) =2616.4, Подошва(ZP) =2975, Шаг
(STEP) =0.2, Направление =ВНИЗ (по умолчанию) =====
- расчет по DBM
- отображение кривой в метод ГИС - АЛГОРИТМ
2715.8-2720 :#RP=-0.055
2742.8-2749 :#RP=-0.1
2754.6-2764.8 :#RP=-0.086
2766.4-2768.6 :#RP=-0.045
2886.8-2911.2 :#RP=-0.148
2924-2925.8 :#RP=-0.035
2928.4-2935.4 :#RP=-0.049
2944.4-2947.4 :#RP=-0.081
2949.2-2965.4 :#RP=-0.038
* ===== расчет формулы завершен =====
- записывается метод ГИС 'W' (101 пропластков)
```

ЗАМЕЧАНИЕ: Полезно использовать функцию **print** для отладочной печати

Несколько формул в одном файле (пакетный расчет)

Программа на ГеоСи может содержать несколько формул в одном файле. В таком случае формулы выполняются последовательно.

```

// расчет АльфаПС по ПС и линии глин (AGL)
Stp1=0;
Stp2=0;
formula Шаг( ) ( *PS, *AGL ) *
{
    if(Stp1<*PS-*AGL) Stp1=*PS-*AGL;
    if(Stp2<-*PS+*AGL) Stp2=-*PS+*AGL;
}
formula Линия_песков( *LPS ) ( *PS, *AGL )
{
    *LPS = *AGL-Stp2+Stp1;
}
formula АльфаПС( *APS ) ( *LPS, *PS, *AGL )
{
    *APS = (*AGL-*PS)/(*AGL-*LPS);
    if(*APS<0) *APS = 0;
    if(*APS>1) *APS = 1;
}

```

Встроенные функции

В ГеоСи встроено множество функций для удобства вычислений, использования других типов данных и т.п. Полный список встроенных функций см. в приложении.

НОВОЕ: Для поиска встроенных функций, получения справки об их параметрах и вставки функций в текст формулы можно использовать меню **Редактирование/Вставить заготовку/Встроенные функции:** появится диалог, в котором функции сгруппированы в разделы.

Ниже предложено описание некоторых часто используемых функций

Математические функции

pow (основание, показатель) основание^показатель	*B= pow (*A, 2) *B=*A^2	Степень
lg (значение) log10 (значение)	*B= lg (*A) *B= log10 (*A)	Десятичный логарифм
ln (значение) log (значение)	*B= ln (*A) *B= log (*A)	Натуральный логарифм
sqrt (значение)	*B= sqrt (*A)	Корень
round (значение)	*B= round (*A)	Округление до целого
exp (значение)	*B= exp (*A)	Экспонента
maxA (массив) minA (массив)	formula MIN () (*BK) { M = minA (*BK[]); }	Возвращает максимально (минимальное) значение кривой или метода гис
NOVAL ()	if(*A== NOVAL ()) *B=0; if(*A!= 0) *B=100/(*A) else *B = NOVAL ();	Возвращает "пустое" значение

Математические константы

Хотя константы по существу отличаются от функций, они часто используются в вычислениях вперемешку.

NULL	"Пустое" значение, то же, что NOVAL()
PI	"Пи" (π) – отношение длины окружности к ее диаметру
E	"Е" (e) – основание натурального логарифма

Синтаксически встроенную константу от встроенной функции, отличает отсутствие круглых скобок в конце.

Функция для получения текущей глубины

DEPTH()	Возвращает текущее значение глубины
----------------	-------------------------------------

```
// пример использования текущей глубины «приведенной» кривой
formula f4(*KPS) (*APS) // пористость по Апс
{
  curdepth = DEPTH(); //значение глубины
  if(*APS==0) *KPS=NOVAL();
  else
  {
    if(curdepth<3072.0) *KPS=1.0/(0.054-0.043*lg(*APS));
    if(curdepth>=3072.0) *KPS=1.0/(0.059-0.0665*lg(*APS));
  }
}
```

Функция для получения абсолютной глубины

ABS_DEPTH (Глубина)	Преобразует глубину (параметр 'Глубина') в значение абсолютной глубины по инклинометрии, а при ее отсутствии – по альтитуде (для вертикальных скважин)
----------------------------	--

```
formula Zabs (*ZA) () // расчет кривой абсолютных глубин
{
  * ZA = ABS_DEPTH( DEPTH() );
}
```

Функции индексирования и работы с глубиной по кривой

CRV_DEPTH (Номер)	Преобразует номер точки кривой ('Номер') в значение глубины
CUR_CRV_POINT ()	Возвращает номер текущей точки кривой
ITH_CRV_POINT (Глубина)	
NUM_CRV_POINTS ()	Возвращает количество точек приведенной кривой текущей скважины

```
// пример переворота кривой
formula TCK (*X) (*KC1)
{
  numpnt = NUM_CRV_POINTS(); // количество точек приведенной кривой
  curpnt = CUR_CRV_POINT(); // номер текущей точки кривой
  *X[numpnt-curpnt-1]=*KC1;
}
```

```

// пример использования текущей точки в цикле
dh = 10;
formula Сглаживание_кривой(*APSSr) (*APS)
{
    *APSSr=0;
    num=0;    valnum=0;
    while(num<dh) // цикл пока выполняется условие в скобках
    {
        ii = CUR_CRV_POINT()-dh/2+num; //номер текущей точки кривой
        num=num+1;
        if( ii >= 0 && ii < NUM_CRV_POINTS() ) //кол-во точек кривой
        {
            *APSSr = *APSSr+*APS[ii];
            valnum=valnum+1;
        }
    }
    if( valnum > 0 )
        *APSSr = *APSSr/valnum;
}

```

```

// заполнение литологии пропластков (гис.LOB) по дискретной кривой LOB
formula Снятие_LOB () (&#LOB,*LOB)
{
    H=(#ZK+#ZP)/2; // #ZK и #ZP - кровля и подошва текущего пропластка
    ind = ITH_CRV_POINT (H); // преобразует значение по глубине
    // в номер точки кривой
    #LOB = round(*LOB[ind]); // округление до целого
}

```

Функции индексирования и работы с глубиной по пропласткам

CUR_GIS_POINT()	Возвращает номер текущего пропластка
CUR_GIS_ZK()	Возвращает кровлю текущего пропластка
CUR_GIS_ZP()	Возвращает подошву текущего
ITH_GIS_POINT(Глубина)	Преобразует глубину (параметр 'Глубина') в номер пропластка
ITH_GIS_ZK(Номер)	Возвращает кровлю пропластка (параметр 'Номер' - это номер пропластка)
ITH_GIS_ZP(Номер)	Возвращает подошву пропластка (параметр 'Номер' - это номер пропластка)
NUM_GIS_POINTS()	Возвращает количество пропластков текущей скважины

Функции использования стратиграфических пластов

LAYER («Имя»)	Возвращает ненулевое значение, если текущая глубина обработки соответствует указанному пласту (Имя)
Inplast («Имя1», «Имя2», ..., Глубина)	Возвращает ненулевое значение, если глубина (параметр 'Глубина') обработки попадает между кровлей верхнего и подошвой нижнего из указанных пластов (параметры 'Имя1', 'Имя2', ...)
CUR_LAYER_ZK ()	Возвращает кровлю текущего пласта
CUR_LAYER_ZP ()	Возвращает подошву текущего пласта
GetLayerCode ()	Возвращает код пласта на текущей глубине расчета
GetLayerName ()	Возвращает строковую константу - имя пласта на

	текущей глубине расчета
GetLayerCodeOfType (Тип)	Возвращает код пласта указанного типа на текущей глубине расчета. Тип задается числом в соответствии с разделом #ТипПласта словаря базы, например, 0 - пласт, 2 - толща.
GetLayerNameOfType (Тип)	Возвращает строковую константу - имя пласта указанного типа на текущей глубине расчета

```
// присвоить индекс пласта параметру пропластков «KINS»
formula Planshet() (&#KINS)
{
    #KINS = GetLayerCodeOfType(0);
    if(GetLayerCode() != NOVAL())
    {
        print(GetLayerNameOfType(0), " (" , #KINS, ") ", ":", #ZK, "-", #ZP);
    }
}
```

```
// использование разных функций расчета пористости
// для разных групп пластов
formula f4(#EPHI) (#APS) { // пористость по Апс
    if( LAYER("BC8") != 0
        || LAYER("BC8-1") != 0
        || LAYER("BC8-2") != 0 )
    {
        #EPHI=(19.23*(#APS));
    }
    if( LAYER("BC10") != 0
        || LAYER("BC10-1") != 0
        || LAYER("BC10-2") != 0
        || LAYER("BC11") != 0 )
    {
        #EPHI=(13.2*(#APS));
    }
}
```

ЗАМЕЧАНИЕ: При расчете кривых функция **LAYER** проверяет, находится ли в пласте текущая точка кривой, а при расчете методов ГИС функция возвращает НЕ-НОЛЬ (1), когда текущий пропласток *полностью* находится внутри пласта.

```
// проверка на попадание середины пропластка в пласт
formula f4_2(#EPHI) (#APS) // пористость по Апс
{
    H=(CUR_GIS_ZK()+CUR_GIS_ZP())/2;
    if( inplast("BC8",H) != 0
        || inplast("BC8-1",H) != 0
        || inplast("BC8-2",H) != 0 )
    {
        #EPHI=(19.23*(#APS));
    }
    else #EPHI=NOVAL();
}
```

ЗАМЕЧАНИЕ: При расчете кривых функция **inplast** с одним пластом проверяет, находится ли текущая точка внутри

пласта. Если нужно аналогичные расчеты проводить между пластами, их нужно перечислить в одной функции **inplast**.

```
// проверка на попадание середины пропластка в диапазон пластов
formula f4_3(#ЕРНІ) (#АПС) // пористость по АПС
{
    Н=DEPTH();
    if( inplast("BC8","BC8-1","BC8-2",Н) != 0 )
    {
        #ЕРНІ=(19.23*(#АПС));
    }
    else #ЕРНІ=NOVAL();
}
```

Однопроходные функции

Эти функции не нуждаются в цикле.

SaveAsLas (*Кривая1[], *Кривая2[], ...)	Сохраняет перечисленные кривые в LAS-файл с именем файла совпадающим с номером скважины, расположенный в той же папке, где находится файл базы
MakeGIS (*Кривая[])	Создает разбиение на пропластки по ступенчатой кривой
Inclinometry ()	Рассчитывает абсолютные отметки пластов и пропластков. Если инклинометрия не была рассчитана, предварительно рассчитывает траекторию скважины
DeleteLog (*Кривая1[], *Кривая2[], ...)	Удаляет перечисленные кривые

```
// Экспорт кривых в LAS-файл
formula ЭкспортКривыхВLasФайл() (*ВК, *ГК)
{
    SaveAsLAS(*ВК[], *ГК[]);
}

// Создание таблицы пропластков по кривой
formula Разбивка() (*ЛОВ)
{
    MakeGIS(*ЛОВ[]);
}

// расчет инклинометрии по скважине
formula Инклинометрия() (*ЛОВ)
{
    Inclinometry();
}

// удаление кривых, например, временных, использующихся только в формуле
formula Подчистка () (*W000,*W001)
{
    DeleteLog(*W000[],*W001[]);
}
```

ЗАМЕЧАНИЕ: Эти функции не рекомендуется объединять с другими инструкциями в одну формулу.

Работа с палетками

ЗАМЕЧАНИЕ: Формулу по палетке форматах **curve**, **chart** и **con** можно создать в КАЛЬКУЛЯТОРЕ (команда **Файл/Создать формулу по палетке**) и в ПЛАНШЕТЕ (команда **Технология/Интерпретатор формул/Создание формулы по палетке**). Подробности в общей инструкции к ГеоПоиск 8 www.geopoisk.com .

Для представления палеток в ГеоСи используется специальная конструкция, «многомерная константа». Физически каждой константе соответствует файл с расширением **con**, который должен лежать в той же папке, где формула.

Палетку можно создать (меню Калькулятора **Файл | Создать константу**) и заполнить значениями во встроенном в Калькулятор редакторе палеток.

Для использования константы в формуле ее надо описать сразу после заголовка формулы и перед ее телом. В описании типичной двумерной константы (таблицы) количество строк – первое число в квадратных скобках, количество столбцов – второе:

```
formula New () ()
const Константа1[91,3];
const Константа2[123,2];
{
}
```

Палетку можно загрузить из текстового файла с разделителями, созданного, например, в Excel. Для этого создается константа соответствующего размера. В редакторе палеток в нее загружаются данные из этого файла.

Если палетка задана в виде графического файла, ее можно оцифровать в программе **Оцифровка карт, палеток и каротажа** (ImageDigitizer) и экспортировать в файл палетки. Файл палетки – это текстовый файл с расширением **curve** для двумерных палеток-графиков и **chart** для трехмерных палеток (с индексами-значениями палеточных кривых).

Хотя ГеоСи поддерживает константы разной размерности, для палеток, как правило, используется формат констант, согласованный с Оцифровщиком. Они представляют собой таблицы (двумерные константы) с двумя, тремя или четырьмя столбцами (двумерные, трехмерные и четырехмерные палетки). Первый столбец соответствует X, второй – Y, третий – значение палеточной кривой (ветви палетки), четвертый – дополнительное значение в узле палеточной кривой.

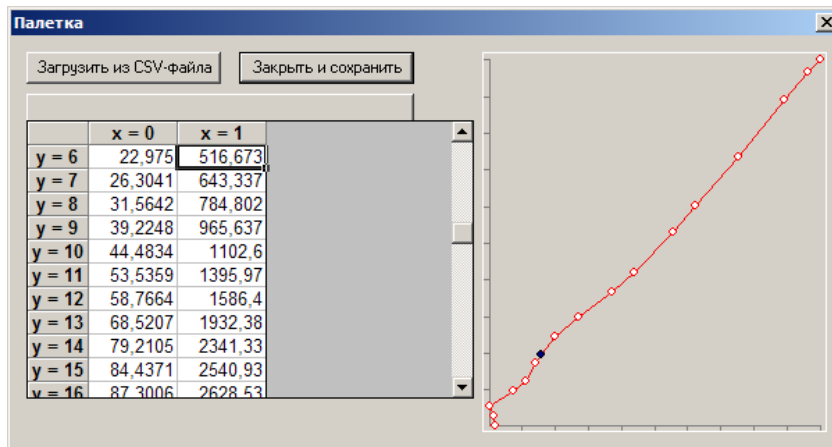
Готовый файл палетки в формате **con**, **curve**, или **chart** можно автоматически преобразовать в формулу, используя команду **Файл/Создать формулу по палетке**. Создание формулы включает преобразование единиц измерения (для поддержки американских палеток).

Интегрированный с кроссплотом генератор формул по палетке есть в Планшете.

Функции для работы с двумерными палетками

graph2y (двумерная константа, X)	Возвращает значение Y по X
graph2x (двумерная константа, Y)	Возвращает значение X по Y

Пример палетки с двумя столбцами (графика) – термоградиент Барсуковского месторождения:

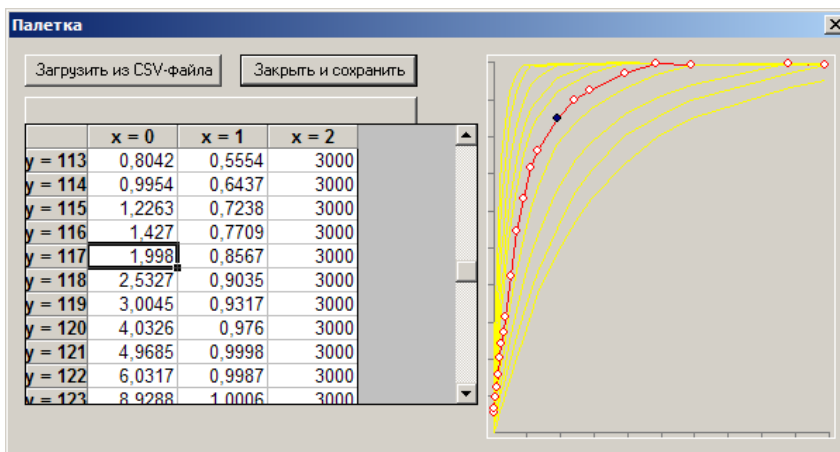


```
//пример использования двумерной палетки для оценки температуры по глубине
formula Барсуковское () (&#T, #ZKA, #ZPA)
const Barsuk[17,2];
{
  H = ABS_DEPTH(DEPTH());
  #T = graph2x(Barsuk, H);
}
```

Функции для работы с трехмерными палетками

grid2x_i (двумерная константа, X, индекс палеточной кривой)	Возвращает значение X по Y и значению ветви палетки
grid2y_i (двумерная константа, Y, индекс палеточной кривой)	Возвращает значение Y по X и значению ветви палетки
gridof_i (двумерная константа, X, Y)	Возвращает интерполированный индекс ветви палетки по X и Y

Пример палетки с тремя столбцами (трехмерной) – поправка РК за инерционность:



```

/* ПОПРАВКА RK ЗА ИНЕРЦИОННОСТЬ */
/* #RK - входной */
/* #RKISP - выходной */
VTAU = 700; // УСТАНОВЛИВАЕТСЯ ВРУЧНУЮ!
/* шифр кривых (V * Tau) - константа, специфическая для прибора и
спускоподъема */
/* значения по палетке от 0 до 10000 */
formula Поправка_RK_за_инерционность (&#RKISP, #RK) #
const Nu[213,3]; // оцифрованные палетки
{
  // инициализация
  #RKISP = #RK;

  H=#ZP-#ZK;
  if(H<0.1)H=0.1;
  if(H>10)H=10;
  _Nu = grid2y_i(Nu,H,VTAU);
  i = ITH_GIS_POINT((#ZK+#ZP)/2)+1;
  if( i<NUM_GIS_POINTS() && i >= 0 )
  {
    RKvm = #RK[i];
    #RKISP = (#RK-RKvm)/_Nu+RKvm;
  }
}
}

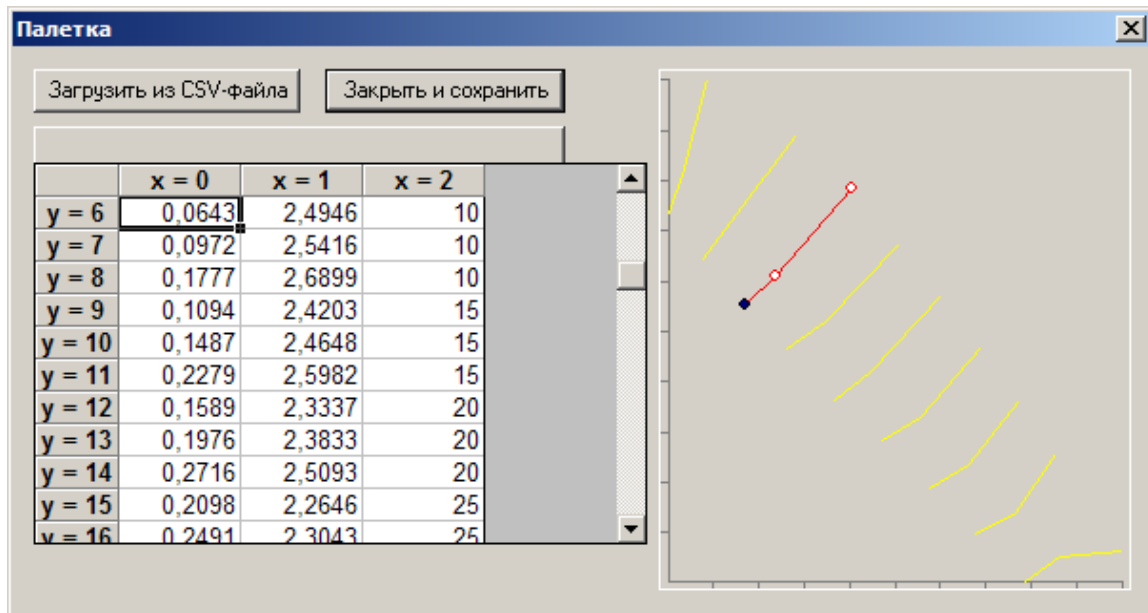
```

Функции для работы с регулярными палетками

chartof (двумерная константа, X, Y)	Возвращает интерполированный индекс ветви палетки по X и Y
chart2n (двумерная константа, X, Y)	Возвращает интерполированный номер узла или значение 4-й колонки

Особое место занимают «регулярные» палетки, т.е. такие трех- или четырехмерные палетки, в которых все ветви имеют совпадающее число и последовательность узлов, а интерполяция выполняется не просто между кривыми, а в ячейках. Пример:





```
// ПАЛЕТКА - пористость и литология (доломит, известняк, песчаник)
// по водородосодержанию (NNK) , и ГГК-п
// Выходные параметры: KP, LITI - известняк, LITD - доломит,
// LITP - песчаник (доли 0-1)
formula Karbonat (*KP) (&*LITI, &*LITD, &*LITP, *GGK, *NNK )
const Karbonat[27,3];
{
  *KP = NOVAL(); *LITP = 0; *LITI = 0; *LITD = 0;
  code=chart2n(Karbonat,*NNK,*GGK);
  if(code!=NOVAL())
  {
    if(code>=0 && code<=1)
    {
      *LITP=code; *LITI=1-code; *LITD=0;
    }
    else if(code>=1 && code<=2)
    {
      *LITP=0; *LITI=code-1; *LITD=2-code;
    }
    *KP = chartof(Karbonat,*NNK,*GGK);
  }
  else
  {
    code=chart2n(Karbonat,*NNK+0.05,*GGK+0.06);
    if(code!=NOVAL())
    {
      *LITP=1; *LITI=0; *LITD=0;
      *KP = chart2n(Karbonat,*NNK+0.05,*GGK+0.06);
    }
    else
    {
      code=chart2n(Karbonat,*NNK-0.05,*GGK-0.08);
      if(code!=NOVAL())
      {
        *LITP=0; *LITI=0; *LITD=1;
        *KP = chartof(Karbonat,*NNK-0.05,*GGK-0.08);
      }
    }
  }
}
}
```

Функция для работы с палетками, заданными узловыми точками

`interp_2lines (x,y, v1, x11,y11, x12,y12, v2, x21,y21,x22,y22)`

Возвращает интерполированное значение в точке (x,y) между двумя отрезками: $(x11,y11 - x12,y12)$ со значением $v1$ и $(x21,y21 - x22,y22)$ со значением $v2$

Кроме палеток, представленных набором линий, поддерживаются палетки, заданные узловыми точками. Пример:

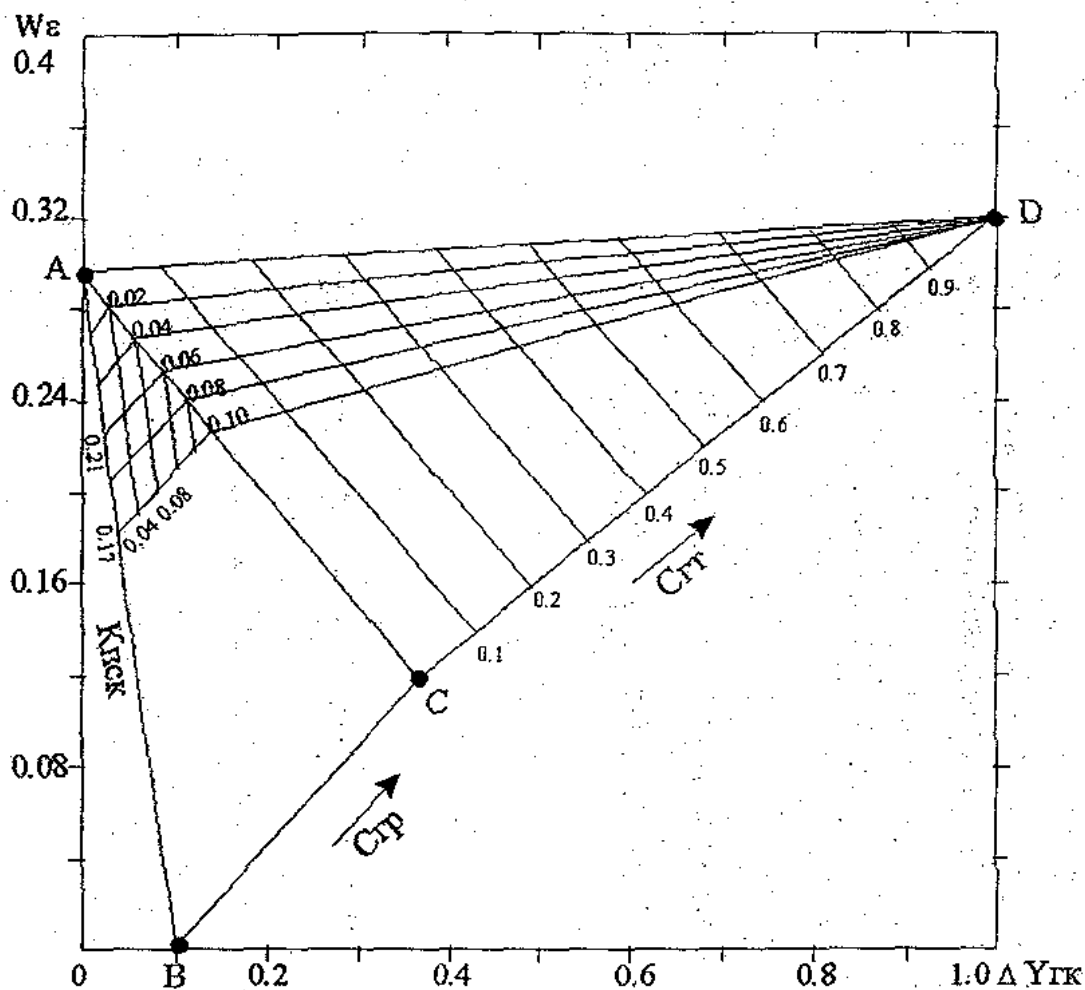


Рис.2 ... Схема "Треугольная палетка" для определения пористости по данным РК на примере пластов БС 10-11 (W_{Σ} - водородосодержание, $\Delta Y_{ГК}$ - двойной разностный параметр по ГК).

Определение компонент пласта по «треугольной палетке»:

Область палетки	Заданное условие	Определяемые параметры
Выше AD	$Стр=0$	$Кпск, Стт$
ACD	$Кпск=Кпск \text{ модальное}$	$Стр, Стт$
ABC	$Стт=0$	$Стр, Кпск$
Левее AB	$Стт=0; Стр=0$	$Кпск=W \text{ водородосод.}$

//=====

```

// ОПИСАНИЕ ПЕРЕМЕННЫХ

Кпск_мод = 0.3;
Кпск_мах = 0.4;

// БС 10-11:
DGKa=0; Wa=0.3; DGKb=0.01; Wb=0; DGKc=0.37; Wc=0.12; DGKd=1; Wd=0.32;

// БС 16-20:
//DGKa=0; Wa=0.247; DGKb=0.05; Wb=0; DGKc=0.285; Wc=0.099; DGKd=1;
Wd=0.251;

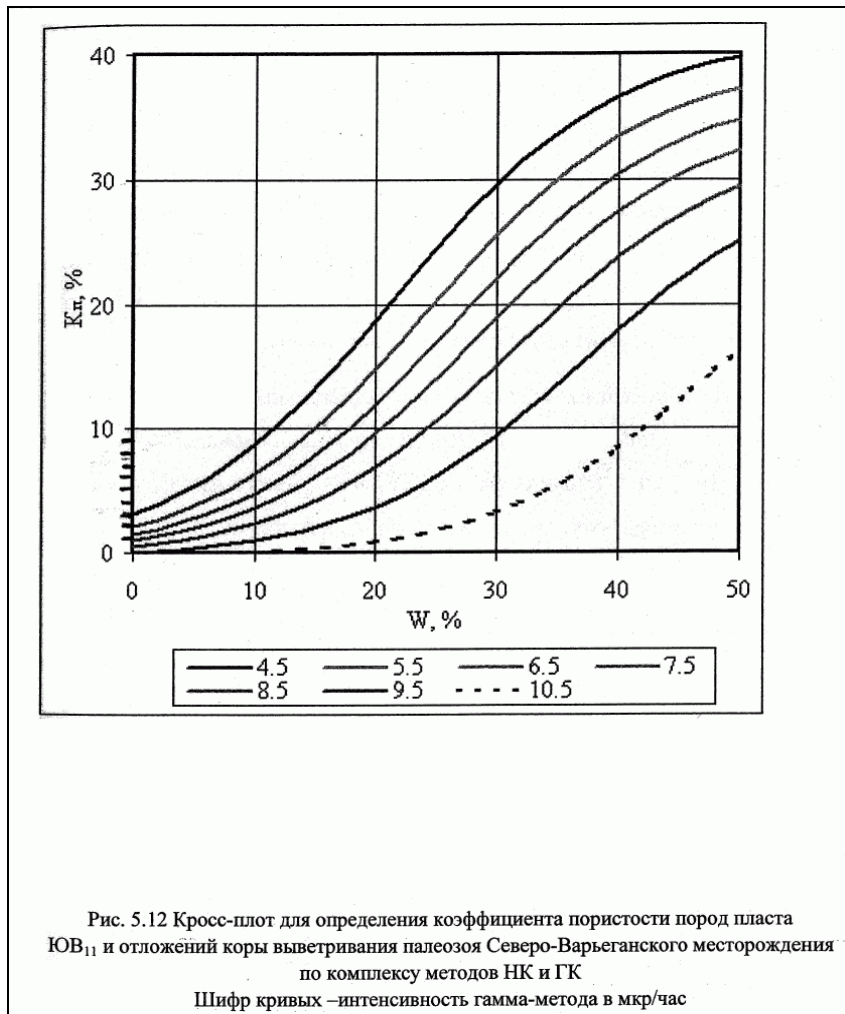
// Ю 1:
//DGKa=0; Wa=0.273; DGKb=0.289; Wb=0; DGKc=0.483; Wc=0.109; DGKd=1;
Wd=0.264;

//=====
// ФОРМУЛА

formula Пористость() (#DGK,#WNKT,&#CGR,&#CGT,&#КРСК,&#КРРК)
{
  #CGR=0; #CGT=0; #КРСК=0; #КРРК=0;
  //ACD
  #CGT = interpol_2lines(#DGK,#WNKT,0,DGKa,Wa,DGKc,Wc,1,DGKd,Wd,DGKd,Wd);
  if(#CGT!=NOVAL())// попали в ACD
  {
    #КРСК = Кпск_мод;
    #CGR = interpol_2lines(#DGK,#WNKT,0,DGKa,Wa,DGKd,Wd,Кпск_мод,DGKc,Wc,
      DGKd,Wd);
    #КРРК = (#КРСК-#CGR)*(1-#CGT);
  }
  else
  {
    #КРСК = interpol_2lines(#DGK,#WNKT,0,DGKb,Wb,DGKc,Wc,Кпск_мод,DGKa,Wa,
      DGKa,Wa);
    if(#КРСК!=NOVAL())// попали в ABC
    {
      #CGR = interpol_2lines(#DGK,#WNKT,0,DGKa,Wa,DGKb,Wb,
        Кпск_мод,DGKc,Wc,DGKc,Wc);
      #CGT = 0;
      #КРРК = (#КРСК-#CGR)*(1-#CGT);
    }
    else
    {
      DGKa1=(DGKa-DGKb)*Кпск_мах/Кпск_мод+DGKb;
      Wa1=(Wa-Wb)*Кпск_мах/Кпск_мод+Wb;
      #CGT = interpol_2lines(#DGK,#WNKT,0,DGKa,Wa,DGKa1,Wa1,
        1,DGKd,Wd,DGKd,Wd);
      if(#CGT!=NOVAL())// попали над AD
      {
        #CGR = 0;
        #КРСК = interpol_2lines(#DGK,#WNKT,Кпск_мод,DGKa,Wa,DGKd,Wd,
          Кпск_мах,DGKa1,Wa1,DGKd,Wd);
        #КРРК = (#КРСК-#CGR)*(1-#CGT);
      }
    }
  }
}
}
}

```

Пример. Палетка определения коэффициента пористости



Оцифрованная палетка загруженная в константу

Константа

Загрузить из CSV-файла Закреть и сохранить

	x = 0	x = 1	x = 2
y = 0	0	3.14	4.5
y = 1	3.4	4.43	4.5
y = 2	7.39	6.68	4.5
y = 3	10.3	8.77	4.5
y = 4	13.37	11.43	4.5
y = 5	16.69	14.73	4.5
y = 6	20.1	18.59	4.5
y = 7	23	21.87	4.5
y = 8	25.91	25.27	4.5
y = 9	28.32	27.93	4.5
y = 10	30.23	29.62	4.5

Интенсивность гамма метода

W Kп

Формула, использующая оцифрованную палетку как двумерную константу

```
formula KP_plasta_JuV11(*KP) (*W,*GK)
const KP_JuV11[133,3]; // описание константы, использующейся в формуле
{
    *KP=NOVAL();
    if( *W != NOVAL() && *GK != NOVAL() )
    {
        *KP=grid2y_i(KP_JuV11, *W, *GK);
    }
}
```

Доступ к параметрам скважины

В Калькуляторе над таблицей параметров пропластков-ГИС, есть окно для просмотра и редактирования таблицы параметров выбранной скважины:

Код	Ключ	Значение
3	СКВАЖИНА	534
14	КУСТ	37
2	ПЛОЩАДЬ	UJNO-HARAMPURSKOE
4	ДАТА	09.08.2006
10	ПОДРЯДЧИК	ОАО ПУРГЕОФИЗИКА
27	СКВАЖИНЫ, УНИК.ИДЕНТИФИКАТОР	281_534
26	Bottom Hole Temp. (Max Temp) , DEGC	80
25	No description , DEGC	90
555	Интервал обработки расходомерии, м	2886.2-2892.2
121	Диаметр колонны, мм	168
122	Глубина спуска колонны, м	2937.5
141	Тип колонны	Эксплуатационная
163	Внутренний диаметр НКТ, мм	140
164	Глубина спуска НКТ, м	2888.3
100	Забой, м	2946.9
154	Интервал текущей перфорации (факт), м	2897.0-2899.0,2899.9-2902.0,2908.5-2913.0,2915.5-2919.6,2924.9-2929.0
362	Интервал текущей перфорации (факт) - пр	2930.4-2933.5

Для ввода и редактирования параметров непосредственно в этом поле: выбрать строку и в контекстном меню по правой кнопки мыши – команду **Редактировать** или команду **Создать**.

Каждый параметр идентифицирует его уникальный цифровой код. Полный список доступных кодов прописан в разделе #PARAM словаря базы. Параметры загружаются в базу из LAS файлов, из файлов описания скважины в специальном формате SKV, или заполняются вручную при редактировании конструкции скважины в Планшете, правке свойств скважины в Проводнике БД и т.п.

При написании формул для многократного использования, особенно с учетом возможностей групповой обработки, рекомендуется использовать данные из таблицы параметров скважины, а не менять каждый раз значения переменных в тексте формулы. Для доступа к данным параметров скважин используется встроенный объект **params**.

Используйте команду **Редактирование/Вставить заготовку/Код параметра скважины** для извлечения данных из параметров

Получение значения числового параметра

Если в параметре с кодом Код записано число, для его вычисления используйте

```
params[Код].value
```

Пример – печать диаметра колонны (см. рис. в начале раздела):

```
formula f() ()
{
  val= params [121].value;
  print ("Диаметр колонны: ", val);
}
```

После выполнения этой формулы в окно протокола будет выдано:

Диаметр колонны: 168

Получение значения текстового параметра

Если в параметре с кодом Код записана строка, для его вычисления используйте

```
params[Код].text
```

Пример – печать диаметра колонны (см. рис. в начале раздела):

```
formula f() ()
{
  print ("Тип колонны: ", params [141].text);
}
```

После выполнения этой формулы в окно протокола будет выдано

Тип колонны: Эксплуатационная

Получение значений числового интервала

Если в параметре с кодом Код записан интервал (два числа через тире, например, «1234.5 – 1235.6» или «1234.5-1235.6»), для получения первого из них используйте

```
params[Код].value.lo
```

а для получения второго –

```
params[Код].value.hi
```

Пример: печать интервала обработки расходомерии (см. рис. в начале раздела):

```
formula f() ()
{
  v1= params [555].value.lo; v2= params [555].value.hi;
  print ("Кровля интервала: ", v1, ", подошва интервала: ", v2);
}
```

}

После выполнения этой формулы в окно протокола будет выдано:

Кровля интервала: 2886.2, подошва интервала: 2892.2

Получение значений набора числовых интервалов

Если в параметре с кодом Код записан набор интервалов через запятую, для получения их количества используйте

```
params[Код].items.count
```

а для получения значений начала и конца *i*-ого интервала - соответственно

```
params[Код].items[i].lo
```

и

```
params[Код].items[i].hi
```

Пример: печать фактических интервалов текущей перфорации (см. рис. в начале раздела):

```
formula f() ()
{
  i=0; n=1;
  while(i<params[154].items.count)
  {
    z1=params[154].items[i].lo;
    z2=params[154].items[i].hi;
    print("Интервал перфорации №",n,": от ",z1," до ",z2);
    i=i+1; n=n+1;
  }
  i=0;
  while(i<params[362].items.count)
  {
    z1=params[362].items[i].lo;
    z2=params[362].items[i].hi;
    print("Интервал перфорации №",n,": от ",z1," до ",z2);
    i=i+1; n=n+1;
  }
}
```

После выполнения этой формулы в окно протокола будет выдано:

Интервал перфорации №1: от 2897 до 2899
Интервал перфорации №2: от 2899.9 до 2902
Интервал перфорации №3: от 2908.5 до 2913
Интервал перфорации №4: от 2915.5 до 2919.6
Интервал перфорации №5: от 2924.9 до 2929
Интервал перфорации №6: от 2930.4 до 2933.5

Получение разбор последовательности диапазонов глубин со значениями

Если в параметре интервал глубин дополнен значением (два числа через тире, затем через двоеточие – третье, например, «2879–2882.3:178»), для получения диапазона глубин используйте

```
params[Код].value.lo или params[Код].items[0].lo
```

```
params[Код].value.hi или params[Код].items[0].hi
```

а для получения третьего числа (значения):

```
params[Код].items[0].value
```

Аналогично, если интервалов несколько, для получения значения привязанного к i-ому интервалу используйте

```
params[Код].items[i].value
```

НОВОЕ: Для поиска параметра по его текстовому описанию и вставки в текст формулы конструкции для доступа к его значению или значениям можно использовать меню **Редактирование/Вставить заготовку/Код параметра скважины.**

Пример. Совместное использование палеток и параметров скважины

```
// Входные данные:
тп = 45; // температура в скважине
Rφ20 = 1.8; // УЭС раствора при 20 градусах Цельсия (Омм)
Рв = 0.03; // УЭС воды в скважине

// Промежуточные данные:
Rφ = 0.25; // УЭС раствора в скважине
Rφэ = 0;
Рвэ = 0;

// Выходные данные:
Es = 0;

formula Запрос_исходных_данных () ()
{
  t=params[26].value;
  if(t<=0)
  {
    t=тп;
    input( "Температура в скважине на глубине обработки:", t);
  }
  тп=t;

  r=params[43].value;
  if(r<=0)
  {
    r=Rφ20;
    input( "УЭС раствора при 20 градусах Цельсия (Омм):", r);
  }
  Rφ20=r;
```

```

}

formula УЭС_раствора () ()
const RS_T[364,3];
{
  print("");
  print("Шаг 1: Палетка рис.4а (МИНХ и ГП): Rφ(tп)");
  print("");

  print("Pp(20) = ",Rφ20," , tп = ",tп);
  print("----");

  Rφ = gridof_i(RS_T,tп,log10(Rφ20))/100;

  print("Rφ = ",Rφ);
  print("");
}

formula Приведенная_ПС () ()
const Rve_Rv_t[91,3];
const Rv_Es_t[24,3];
{
  print("");
  print("Шаг 2: Палетка рис.73 (Шлюмберже): Rвэ(Rв,tп)");
  print("");
  Rφэ=10^grid2x_i(Rve_Rv_t,lg(Rφ),tп);
  print("Rφ = ",Rφ," => Rφэ = ",Rφэ);

  Rвэ=10^grid2x_i(Rve_Rv_t,lg(Rв),tп);
  print("Rв = ",Rв," => Rвэ = ",Rвэ);

  print("");
  print("Шаг 3: Палетка рис.72 (Дрессер Атлас): Es(tп,Rвэ)");
  print("");
  print("Rφэ/Rвэ = ",Rφэ/Rвэ);
  Es=grid2x_i(Rv_Es_t,lg(Rφэ/Rвэ),tп);
  print("Es = ",Es);

  print("");
}

```

Доступ к данным керна и описаний интервалов скважины

В версии ГеоПоиск 7.6 появились встроенные функции для доступа к данным керна: лабораторным исследованиям и первичным или увязанным литологическим описаниям керна по интервалам.

Функции для работы с лабораторными исследованиями керна

CUR_CORE ("Параметр") или CUR_CORE_USE_0 ("Параметр")	возвращает среднее значение параметра керна в интервале текущего пропластка или точки кривой (USE_0 – учитывать нули)
CUR_CORE_ERR ("Параметр") или CUR_CORE_ERR_USE_0 ("Параметр")	возвращает среднеквадратичное отклонение параметра керна в интервале текущего пропластка или точки кривой (USE_0 – учитывать 0)
CUR_CORE_MAX ("Параметр")	возвращает максимальное значение

или CUR_CORE_MAX_USE_0 ("Параметр")	параметра керна в интервале текущего пропластка или точки кривой (USE_0 – учитывать нули)
CUR_CORE_MIN ("Параметр") или CUR_CORE_MIN_USE_0 ("Параметр")	возвращает минимальное значение параметра керна в интервале текущего пропластка или точки кривой (USE_0 – учитывать нули)
CUR_CORE_NUM ("Параметр") или CUR_CORE_NUM_USE_0 ("Параметр")	возвращает количество замеров параметра керна в интервале текущего пропластка или точки кривой (USE_0 – учитывать нули)
CUR_CORE_ZK()	возвращает кровлю текущего интервала долбления керна
CUR_CORE_ZP()	возвращает подошву текущего интервала долбления керна

Функции для работы с описаниями интервалов

CUR_TEXT (Тип описания)	возвращает текст текущего интервала описания указанного типа
CUR_TEXT_CODE (Тип описания)	возвращает код (литологию) текущего интервала описания указанного типа
CUR_TEXT_ZK (Тип описания)	возвращает кровлю текущего интервала описания указанного типа
CUR_TEXT_ZP (Тип описания)	возвращает подошву текущего интервала описания указанного типа

Помимо литологических описаний керна описания по интервалам используются для представления в текстовом виде результатов испытаний, комментариев и т.п. Разные типы описаний идентифицируются текстовыми номерами, описанными в словарики базы.

Пример. Печать данных по керну и описаниям

```
formula New () (#ЛОБ,#КР) #
{
  if(CUR_GIS_POINT ()==0)
  {
    print("FIELD: ",FIELD()," WELL: ",WELL());
  }
  print("Глубина пропластка: ",#ZK," - ",#ZP,
    ", а.о.: ",ABS_DEPTH(#ZK)," - ",ABS_DEPTH(#ZP));
  if(CUR_CORE_ZK()!=NOVAL())
    print(" Интервал долбления: ",CUR_CORE_ZK()," - ",CUR_CORE_ZP());
  if(CUR_CORE_NUM("КР")>0)
  {
    print(" Кп по керну=",CUR_CORE("КР")," (" ,CUR_CORE_MIN("КР"),
      " - ",CUR_CORE_MAX("КР"),"), Кп по ГИС=",#КР*100);
    print(" Измерений Кп керна: ",CUR_CORE_NUM("КР"),
      ", среднеквадратическая ошибка: ",CUR_CORE_ERR("КР"));
  }
  if(CUR_TEXT_CODE(1)!=NOVAL())
  {
    print(" Литологический интервал по керну: ",CUR_TEXT_ZK(1),
      " - ",CUR_TEXT_ZP(1));
    print(" Литология по ГИС: ",#ЛОБ," по керну :",CUR_TEXT_CODE(1),
      ", текст описания: ",CUR_TEXT(1));
  }
  if(CUR_TEXT_CODE(9)!=NOVAL())
  {
```

```

print(" Интервал ПГИ: ", CUR_ТЕХТ_ZK(9), " - ", CUR_ТЕХТ_ZP(9));
print(" Результаты испытания: ", CUR_ТЕХТ(9));
}
}

```

Построение моделей

Для упрощения создания моделей используйте команды меню **Редактирование/Вставить заготовку: Литология; Насыщение; Формула по группам пластов; Пласты текущей скважины; Условия для литологии и Условия для насыщения.**

Пример. Модель по кривым

```

// Расчет объемной модели
//
//Входные данные:
// АльфаПС (*APS), УЭС пласта (*RP),
// литология (#LOB), насыщение (#NOB)
//
//Выходные данные:
// пористость (*KP), глинистость (*KGL),
// эфф. пористость (*KPE), проницаемость (*KPR),
// алевролитность (*ALEV), карбонатность (*CARB),
// песчаннистость (*KPS),
// Кв (*KV), Кнг (*KNG), Кн (*KN), Кг (*KG)
//
//Все коэффициенты - в долях единицы

Rпв = 0.15; // УЭС пластовой воды, Омм

// ===== LOB: обобщенный литотип =====
Песчанник = 1;
Песч_гл_ = 2;
Глина = 3;
Аргиллит = 4;
Плотный = 5;
Уголь = 6;

// ===== NOB: обобщенное насыщение =====
Газ = 1;
Конденсат = 2;
Нефть = 3;
Продукт = 4;
Газ_Вода = 5;
Вода = 6;
Нефть_Вода = 7;
Возм_продукт = 8;
Газ_недонас_зона = 9;
Продукт_Вода = 10;
Неясно = 11;

formula КП () (&*KP,*RP,*APS,#LOB)
{
    ЛИТ = #LOB;
    if(*RP>0 && ЛИТ==Плотный)
        КП = (Rпв / *RP)^0.56;
    else
    {
        КП = 0.01*(20.15* *APS +18.25);
    }
}

```

```

        if(КП>0.39) КП=0.39;
    }
    if(КП<0.01) КП = 0.01;
    if(КП>1) КП = 1;
    *КР = КП;
}

formula Карбонатность () (&*CARB,*КР)
{
    Карб = 51.65-18.2*ln(*КР*100);
    if(Карб<0) Карб=0;
    if(Карб>100) Карб=100;
    Карб = Карб*0.01*(1-*КР);
    *CARB = Карб;
}

formula КГЛ () (&*KGL,*КР,*АПС,*САРВ)
{
    КГЛ = 0.01*(38-143*(*КР));
    if(КГЛ<0) КГЛ = 0;
    if(КГЛ>1) КГЛ = 1;
    if(*АПС<=0.2)
        КГЛ = 0.01*(74.8-45*(*АПС));
    else
        КГЛ = КГЛ*(1-*КР)*(1-*САРВ);
    if(КГЛ<0) КГЛ = 0;
    if(КГЛ>1) КГЛ = 1;
    *KGL = КГЛ;
}

formula Кп_эф () (&*КРЕ,*КР)
{
    КПэф = *КР*1.93-0.374;
    if(КПэф<0) КПэф = 0;
    if(КПэф>1) КПэф = 1;
    *КРЕ = КПэф;
}

formula Алевролит () (&*ALEV,*АПС)
{
    Алевролит = 0.01*(95.79*0.096^(*АПС^2));
    if(Алевролит<0) Алевролит = 0;
    if(Алевролит>1) Алевролит = 1;
    *ALEV = Алевролит;
}

formula Песчанник () (&*KPS,*КР,*KGL,*ALEV,*САРВ)
{
    *KPS = 1-*КР-*KGL-*ALEV-*САРВ;
    if(*KPS<0) *KPS=0;
    if(*KPS>1) *KPS=1;
}

formula Кпр () (&*KPR,*КРЕ,#NOB)
{
    if(#NOB==Газ) *KPR = 10^(0.132*(*КРЕ*100-0.75));
}

formula Кнг () (&*KV,&*KNG,&*KN,&*KG,*RP,*КР,#NOB)
{

```

```

КП = *КР;
if(КП<0.01) КП=0.01;
if(#NOB!=Вода)
{
    КВ = 10^(1.47-0.43*log10(*RP));
    КВ = КВ/КП*0.01;
    if(КВ>1) КВ=1;
}
else
{
    КВ=1;
}
*KV = КВ;
*KNG = 1-КВ;
if(#NOB==Газ)
    *KG = 1-КВ;
if(#NOB==Нефть)
    *KN = 1-КВ;
}

```

Пример. Модель по пропласткам с учетом групп пластов

```

// Группы пластов
// ВНИМАНИЕ! Проверьте правильность объединения пластов в группы (формулу
Kins)
Группа_АВ=1;
Группа_ВВ=2;
Группа_ВВ8=3;
Группа_Ачим=4;
Группа_ЮВ=5;

// ===== ЛОВ: обобщенный литотип =====
Песчаник = 1;
Глина = 2;
Гл_песч_ = 3;
С_г_песч_ = 4;
Плотный = 35;
Уголь = 45;
Битумин_ = 52;

// ===== NOB: обобщенное насыщение =====
Нефть = 1;
Газ = 2;
Обводнение = 3;
Обвод_пресн_водой = 4;
Вода = 5;
Газ__вода_пласт_ = 6;
Нефть_Вода = 7;
Вода_минерализован_ = 9;
Пониж_нефтенасыщ_ = 10;
Не_ясен_в_коллект_ = 12;
Неколлектор = 13;

// Объединение пластов в группы
// ВНИМАНИЕ! при добавлении пластов в группы следите за точностью ввода
кодов!
formula Kins(*KINS)()
{
    *KINS = 0;

    if( LAYER("АВ")!=0

```

```

        || LAYER("AB1")!=0
        || LAYER("AB1(1)")!=0 || LAYER("AB1(2)")!=0 ||
LAYER("AB1(3)")!=0
        || LAYER("AB2")!=0 || LAYER("AB2(1)")!=0 || LAYER("AB2(2)")!=0
        || LAYER("AB3")!=0
        || LAYER("AB4")!=0
        || LAYER("AB5")!=0
        || LAYER("AB6")!=0
        || LAYER("AB7")!=0
        || LAYER("AB8")!=0 )
        *KINS = Группа_AB;

if( LAYER("BB")!=0
    || LAYER("BB0")!=0
    || LAYER("BB1")!=0
    || LAYER("BB2")!=0
    || LAYER("BB3")!=0
    || LAYER("BB4")!=0
    || LAYER("BB5")!=0
    || LAYER("BB6")!=0
    || LAYER("BB7")!=0 )
    *KINS = Группа_BB;

if( LAYER("BB8")!=0
    || LAYER("BB8(0)")!=0 || LAYER("BB8(1)")!=0 ||
LAYER("BB8(2)")!=0 )
    *KINS = Группа_BB8;

if( LAYER("BB10")!=0 || LAYER("BB10(1)")!=0 || LAYER("BB10(2)")!=0
    || LAYER("BB16")!=0
    || LAYER("BB17")!=0
    || LAYER("BB18")!=0
    || LAYER("BB19")!=0
    || LAYER("BB20")!=0
    || LAYER("BB21")!=0
    || LAYER("BB22")!=0 )
    *KINS = Группа_Ачим;

if( LAYER("ЮБ")!=0
    || LAYER("Ю0")!=0
    || LAYER("Ю1(1)")!=0 || LAYER("Ю3(3)")!=0 || LAYER("Ю")!=0
    || LAYER("Ю2(2)")!=0 || LAYER("Ю4")!=0 )
    *KINS = Группа_ЮБ;
}

// формула для расчета параметров за пределами пластов (в "дырках")
// прокомментируйте ее, если нужен расчет только внутри пластов
formula Kins2()(&*KINS)
{
    if( *KINS == 0 )
    {
        inum = NUM_CRV_POINTS() ;
        icur = CUR_CRV_POINT();
        i = icur+1;
        while( i<inum && *KINS == 0 )
        {
            if( *KINS[i] > 0 )
            {
                *KINS = *KINS[i];
            }
            i = i+1;
        }
    }
}

```

```

    }
}

// ПЕТРОФИЗИКА
/*
#PHIE - Кп(ПС) коэфф. пористости
#PERM - Кпр(ПС) проницаемость
#SW - Кв - коэфф. водонасыщенности доли.
#SO - Кн - коэфф. нефтенасыщенности доли.
#FRF - Рп относительное сопротивление
#WP - Рв
#RWP - Рвп - сопротивление водоносного пласта
#PN - Пн -
#RT - Рп
*/
formula Petrophizica() (&#NOB, &#PHIE, &#PERM, &#SW, &#SO, &#FRF, &#WP,
&#PN, &#RWP, #APS, #RT, #LOB, *KINS )
{
    H = (#ZK+#ZP)/2;
    ind = ITH_CRV_POINT (H);
    groupPlast = round(*KINS[ind]);

    #NOB = 0;
    #PHIE = NOVAL();
    #PERM = NOVAL();
    #SW = NOVAL();
    #SO = NOVAL();
    #FRF = NOVAL();
    #WP = NOVAL();
    #PN = NOVAL();
    #RWP = NOVAL();

    if( groupPlast == Группа_БВ )
    {
        #PHIE=9.351* #APS + 13.29;
        #PERM=10 ^ (0.357* #APS - 6.28);
        // проницаемый
        if( #LOB == Песчаник || #LOB == С__г_песч_ || #LOB == Гл_песч_
)
        {
            if( #RT < 4 ) #NOB = Вода;
            if( #RT >= 4 && #RT < 4.7 ) #NOB = Нефть_Вода;
            if( #RT >= 4.7 ) #NOB = Нефть;
        }
        if( #PHIE == 0 || #RT <= 0 ) #SW = NOVAL();
        else #SW = (10^(0.241* (lg(#RT)^2)-
0.986*lg(#RT)+1.616))/#PHIE;
        if( #SW < 0 ) #SW = 0;
        if( #SW > 0.98 ) #SW = 0.98;

        #SO = 1 - #SW;
    }
    if( groupPlast == Группа_ЮВ )
    {
        if( #LOB == Песчаник || #LOB == С__г_песч_
|| #LOB == Гл_песч_ || #LOB == Глина )
        {
            #PHIE=12.1* #APS + 7.2;
            #PERM=10 ^ (0.36* #PHIE - 5.0);
        }
    }
}

```

```

// параметр пористости
if( #PHIE <= 0 ) #FRF = NOVAL();
else #FRF = 10 ^ (3.4032 - 1.6129*(lg(#PHIE) ) );
}
if( #LOB == Песчаник || #LOB == С__г_песч_ || #LOB == Гл_песч_
)
{
// проницаемый
if( #RT < 4.8 ) #NOB = Вода;
if( #RT >= 4.8 && #RT < 5.5 ) #NOB = Нефть_Вода;
if( #RT >= 5.5 ) #NOB = Нефть;

// сопротивление водоносного пласта
#RWP = 0.01;
if( #FRF != NOVAL() )
#RWP = #FRF*0.08;

// параметр насыщения
#PN = #RT/#RWP;

if( #PN <= 0 ) #SW=NOVAL();
else
{
if( #PERM < 1 ) #SW=10^(2.0169-0.8275*lg(#PN));
if( #PERM >= 1 && #PERM < 10 )
#SW=10^(2.0244-0.813*lg(#PN));
if( #PERM >= 10 ) #SW=10^(2.01-0.6711*lg(#PN));
}
#SW = #SW/100;
if( #SW < 0 ) #SW = 0;
if( #SW > 1 ) #SW = 1;

#SO = 1 - #SW;
}
}
}

```

Функции для решения систем уравнений

Кроме палеток и статистических зависимостей модели можно описывать системами уравнений. ГеоСи включает средства для решения систем обыкновенных линейных уравнений и определения компонентного состава породы.

Gauss (x1, x2,..., a11, a12,... b1, a21, a22,..., b2,...)	решает систему линейных уравнений (в начале идут переменные, потом коэффициенты уравнений по строкам). Если все коэффициенты неотрицательны и среди уравнений есть уравнение, все коэффициенты которого равны 1, эта функция ищет решение с неотрицательными значениями переменных. (Такие системы уравнений соответствуют задаче определения компонентного состава.) В остальных случаях система решается как обычно без ограничения знака переменных
components (n, C1, C2,..., Cn, P11,P12,..., P1n,V1, P21,P22,..., P2n,V2, ...)	определяет компонентный состав породы или флюида в точке по показаниям нескольких методов и средним значениям показаний этих методов в «чистой» породе или флюиде каждого типа. Первый параметр - число компонентов, за ним идут выходные переменные, которым будут присвоены значения

	<p>доли каждой компоненте (0..1), потом показания первого метода в «чистых» компонентах и фактическое значение этого метода в смеси, потом показания второго метода в «чистых» компонентах и фактическое значение второго метода в смеси и т.д.</p> <p>Фактически эта функция тоже решает систему линейных уравнений, но с учётом того, что доля компонента не может быть отрицательной, а сумма долей всех компонентов составляет 1.</p> <p>Для этой функции не требуется совпадения числа компонентов и количества использованных методов.</p> <p>Соотношение между ними может быть любым</p>
<p>auto_components (n, C1, C2,..., Cn, P11,P12,..., P1n,V1, P21,P22,..., P2n,V2, ..., max_err)</p>	<p>определяет компонентный состав породы или флюида в точке по показаниям нескольких методов и средним значениям показаний этих методов в «чистой» породе или флюиде каждого типа.</p> <p>Первый параметр – число компонентов, за ним идут выходные переменные, которым будут присвоены значения доли каждой компоненте (0..1), потом показания первого метода в «чистых» компонентах и фактическое значение этого метода в смеси, потом показания второго метода в «чистых» компонентах и фактическое значение второго метода в смеси и т.д. Последний параметр – оценка допустимой погрешности подбора, оценивается исходя из значения погрешности измерений использованных методов (0.05–0.1).</p> <p>Эта функция аналогична функции components, но надёжнее работает при большом числе возможных компонент, т.к. отбирает небольшую группу значимых компонент и оптимизирует состав только по ним.</p> <p>Фактически эта функция тоже решает систему линейных уравнений, с учётом того, что доля компоненты не может быть отрицательной, а сумма долей всех компонент составляет 1, но не сразу, а по шагам.</p> <p>Сначала подбирает самую подходящую компоненту, потом – лучшую пару компонент, потом – лучшую тройку и т.д.</p> <p>Прекращает работу тогда, когда добавление новой компоненты не приводит к существенному улучшению точности подбора. Работает медленнее, чем components.</p> <p>Для этой функции также не требуется совпадения числа компонент и количества использованных методов.</p> <p>Соотношение между ними может быть любым.</p>
<p>НОВОЕ: components_lim (n, C1, C2,..., Cn, min1,min2..., minn, max1,max2...,maxn, P11,P12,..., P1n,V1, P21,P22,..., P2n,V2, ...)</p>	<p>определяет компонентный состав породы или флюида в точке по показаниям нескольких методов и средним значениям показаний этих методов в «чистой» породе или флюиде каждого типа <u>с учетом ограничений</u>.</p> <p>Первый параметр – число компонентов, за ним идут выходные переменные, которым будут присвоены значения доли каждой компоненте (0..1), за ними – минимальные значения доли каждого компонента, потом – максимальные значения доли каждого компонента, затем показания первого метода в «чистых» компонентах и фактическое значение этого метода в смеси, потом показания второго метода в «чистых» компонентах и фактическое значение второго метода в смеси и т.д.</p> <p>Фактически эта функция тоже решает систему линейных уравнений, но с учётом того, что доли компонентов не выходят из заданных ограничений, а сумма долей всех компонентов составляет 1.</p> <p>Для этой функции не требуется совпадения числа компонентов и количества использованных методов.</p>

Соотношение между ними может быть любым

<p>ЗАМЕЧАНИЕ: эти встроенные функции можно вызывать только из пользовательских функций, в которые x1, x2,... или C1, C2,... переданы в качестве параметров.</p>
--

Кроме палеток и статистических зависимостей модели можно описывать системами уравнений. ГеоСи включает средства для решения систем обыкновенных линейных уравнений.

```

//пластовый флюид
dT_f=530; //время пробега в пластовом флюиде
Ro_f=1.1; //плотность пластового флюида
W_f=1; //водородосодержание пластового флюида
//известняк
dT_i=155; //время пробега в известняке
Ro_i=2.71; //плотность известняка
W_i=0; //водородосодержание известняка
//доломит
dT_d=138; //время пробега в доломите
Ro_d=2.87; //плотность доломита
W_d=0.03; //водородосодержание доломита
//ангидрит
dT_a=164; //время пробега в ангидрите
Ro_a=2.98; //плотность ангидрита
W_a=0.007; //водородосодержание ангидрита

function вариант_1 (&Кп, &Изв, &Дол, &Анг, АК, GGK, W)
{
    Кп=NOVAL (); Изв=NOVAL (); Дол=NOVAL (); Анг=NOVAL ();
    //Уравнения:
    //1) АК = Кп*dT_f + Изв*dT_i + Дол*dT_d + Анг*dT_a
    //2) GGK = Кп*Ro_f + Изв*Ro_i + Дол*Ro_d + Анг*Ro_a
    //3) W = Кп*W_f + Изв*W_i + Дол*W_d + Анг*W_a
    //4) Кп + Изв + Дол + Анг = 1
    if( Gauss (Кп, Изв, Дол, Анг,
        dT_f, dT_i, dT_d, dT_a, АК,
        Ro_f, Ro_i, Ro_d, Ro_a, GGK,
        W_f, W_i, W_d, W_a, W,
        1., 1., 1., 1., 1.) > 0 ) return 0;
    return NOVAL ();
}
function вариант_2 (&Кп, &Изв, &Дол, &Анг, АК, GGK, W)
{
    Кп=0; Изв=0; Дол=0; Анг=0;
    return components (4, Кп, Изв, Дол, Анг,
        dT_f, dT_i, dT_d, dT_a, АК,
        Ro_f, Ro_i, Ro_d, Ro_a, GGK,
        W_f, W_i, W_d, W_a, W);
}
function вариант_3 (&Кп, &Изв, &Дол, &Анг, АК, GGK, W)
{
    Кп=0; Изв=0; Дол=0; Анг=0;
    return auto_components (4, Кп, Изв, Дол, Анг,
        dT_f, dT_i, dT_d, dT_a, АК,
        Ro_f, Ro_i, Ro_d, Ro_a, GGK,
        W_f, W_i, W_d, W_a, W, 0.05);
}
function вариант_4 (&Кп, &Изв, &Дол, &Анг, АК, GGK, W)
{
    Кп=0; Изв=0; Дол=0; Анг=0;
    return components_lim (4,

```

```

Кп, Изв, Дол, Анг,
0, 0, 0, 0,
0.4, 1, 1, 0.5,
dT_f, dT_i, dT_d, dT_a, AK,
Ro_f, Ro_i, Ro_d, Ro_a, GGK,
W_f, W_i, W_d, W_a, W);
}

formula Карбонаты () (&*KP, &*IZV, &*DOL, &*ANG, *DT, *GGK, *WNKT)
{
  if (вариант_4 (*KP, *IZV, *DOL, *ANG, *DT, *GGK, *WNKT) == NOVAL ())
  {
    *KP=NOVAL (); *IZV=NOVAL (); *DOL=NOVAL (); *ANG=NOVAL ();
  }
}

```

В этом примере формула «Карбонаты» может вызвать любой из четырёх вариантов функции оценки состава: «вариант_1», «вариант_2», «вариант_3», или «вариант_4». Результаты будут несколько различаться. Рекомендуется использовать **components_lim** - «вариант_4».

ЗАМЕЧАНИЕ: Для упрощения создания формул по системе уравнений в программу ПЛАНШЕТ встроена специальная команда Технология/Интерпретатор формул/Состав породы (Система уравнений). Подробности в общей инструкции к ГеоПоиск 8 на сайте www.geopoisk.com

Приложение

Основные конструкции языка ГеоСи

```

// шаблон формулы
formula New( ) ( ) // заголовок формулы
{
  /*Тело формулы*/
  /*
  инструкция1;
  инструкция2;
  ...
  инструкцияN;
  */
}

////////////////////////////////////
// функция
function New(<переменная1>, <переменная2>, ...)
{
  return <значение>;
}

////////////////////////////////////
// выбор по условию
if(<условие>) // если
{
  // условие выполняется
  /*
  инструкция1;
  инструкция2;
  ...

```

```

    */
}
else // иначе
{
    // условие выполняется
    /*
    инструкция1;
    инструкция2;
    ...
    */
}

////////////////////////////////////
// цикл на каждом шаге выполнения формулы
while(<условие>) // цикл пока выполняется условие в скобках
{
    // условие выполняется
    /*
    инструкция1;
    инструкция2;
    ...
    */
}

do // делать
{
    // условие выполняется
    /*
    инструкция1;
    инструкция2;
    ...
    */
}
while(<условие>); // цикл пока выполняется условие в скобках

////////////////////////////////////
// текст в файл protocol.txt
protocol(<функция|переменная|кривая|метод гис|строка>, <...>, <...>, ...);
// текст в листинге
print(<функция|переменная|кривая|метод гис|строка>, <...>, <...>, ...);

```

Отношения в <условии>

<	Меньше	if(*A<1) *B=0;
>	Больше	if(*A>1) *B=0;
>=	Больше или равно	if(*A>=1) *B=0;
<=	Меньше или равно	if(*A<=1) *B=0;
==	Равно	if(*A==1) *B=0;
!=	Не равно	if(*A!=1) *B=0;

Объединение нескольких условий в <условии>

&&	И	If(*A<1 && *A>0) *B=0;
	ИЛИ	If(*A<1 *A>10) *B=0;
!	НЕ	If(!(*A==1)) *B=0;

Управление вычислением формулы из ГеоСи

Калькулятор обычно сам разбирает, какой именно цикл подразумевает каждая формула. Направление расчета формулы можно задать командой **Вверх** или **Вниз** меню

Действие/Направление расчета. Однако, дополнительные пометки в конце заголовка формулы (перед const и фигурной скобкой "{") позволяют управлять «стратегией вычислений» в самом тексте программы. Например:

```
formula F1 () (*K,#P) *
{
    print ("K=",*K," P=",#P);
}
```

указатель стратегии
«по кривой»

делает расчет в заданном диапазоне глубин с постоянным шагом. Эта стратегия итерации – «по кривой». Другой пример:

```
formula F2 () (*K,#P) #
{
    print ("K=",*K," P=",#P);
}
```

указатель стратегии
«по пропласткам»

Здесь, наоборот, явно указано, что нужно делать цикл заданном диапазоне глубин по таблице пропластков. При этом с кривой *K будет автоматически сниматься и печататься отсчет по текущему пропластку. Эти отсчеты формируются в памяти Калькулятора, но в базу данных не сохраняются.

Чтобы отсчеты сохранились, преобразование должно быть задано явно, например:

```
formula F3 () (*K,&#P) #
{
    #P = *K;
}
```

В этом случае признак стратегии итерации можно убрать, т.к. здесь есть изменяемый параметр (отсчеты по пропласткам #P).

```
formula F3 () (*K,&#P) #
{
    #P = *K;
}
```

Эти формулы
эквивалентны!

```
formula F3 () (*K,&#P)
{
    #P = *K;
}
```

Иногда требуется выполнить однократное действие, например, посчитать максимум кривой. В этом случае требуется третья стратегия – *однократный проход*, и тогда опускается указатель стратегии – символ * или # между списком входных параметров и телом формулы. Например,

```
formula F4 () (*K)
{
    print("максимум K равен ",
    maxA (*K []));
}
```

Нет звездочки или решетки,
значит – стратегия
«однократный проход»

Однократный проход автоматически выполняется только если в списке аргументов нет изменяемых.

Направление расчета (сверху-вниз или снизу-вверх)

Еще один аспект итерации – направление обхода (перебора) точек. Если для формулы это направление существенно, его можно указать в ее описании: up (снизу-вверх) или down (сверху-вниз).

```
formula F5 () (*K) * up
{
    print (*DEPTH(), " K=", *K);
}
```

указатель направления
«снизу-вверх»

Выбор аргументов во время выполнения

Бывает, что одну и ту же операцию необходимо применять к входным кривым с разными именами. Для этого случая GeoSi поддерживает уточнение имен кривых в процессе выполнения:

```
formula F1 () ("Входная кривая: "* __K__ )
{
    print ("макс=%g", maxA(* __K__ []));
}
```

Признак «выбираемого
аргумента» – текст
его описания

Если в скважине нет кривой __K__ (а такой скорее всего нет), программа выдаст диалог с предложением сопоставить одну из существующих кривых, кривой __K__, или присвоить всем ее точкам признак отсутствия значений NOVAL().

Заготовки формул в Калькуляторе

Команды меню **Редактирование/Вставить заготовку**.

Команда **Редактирование/Вставить заготовку/Пустая формула**:

```
formula New () ()
{
}
```

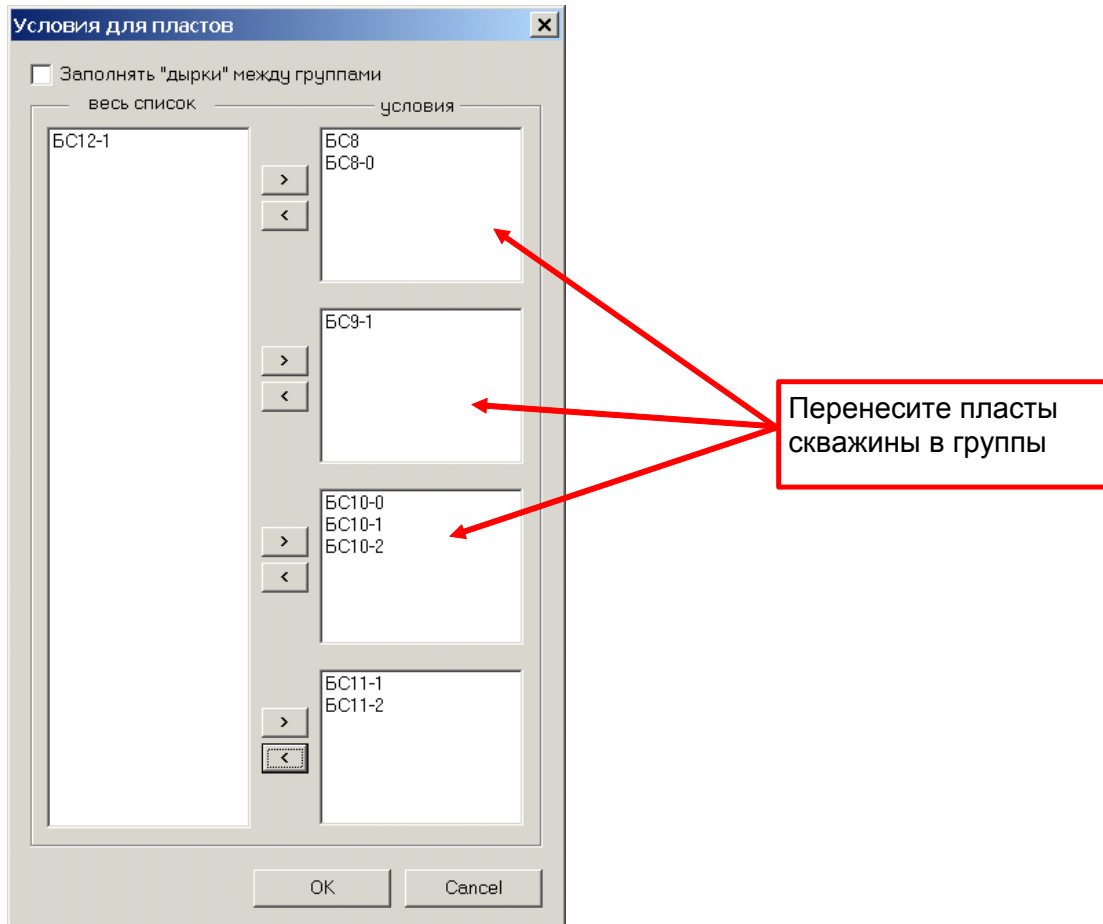
Команды **Редактирование/Вставить заготовку/Литология** и **Редактирование/Вставить заготовку/Насыщение** вставляют в формулу переменные, соответствующие кодам литологии (насыщения) по текущему словарю базы. Например, для литологии базы «...\GeoPoisk\RBD\Test.mp»:

```
// ===== LOB: обобщенный литотип =====
нет_данных = 0;
Глина = 1;
Известняк = 6;
Песчаник = 7;
Песч_глинистый = 8;
Песч_сильно_глин_ = 9;
Песч_уплотненный = 10;
Аргиллит = 2;
Алевролит = 3;
Доломит = 5;
Мергель = 12;
Уголь = 4;
```

```
Неколлектор = 11;
```

Эти переменные нужны, чтобы в теле формул использовать строки **Глина**, **Известняк** и т.д. вместо значений кодов **1**, **6** и др.

Команда **Редактирование/Вставить заготовку/формула по группам пластов** автоматически создает формулу, содержащую шаблоны условий для нескольких (до 4-х) групп пластов. Используется функция **inplast()**. Например,



Результат

```
formula По_группам_пластов() ()
{
  if (inplast(DEPTH(), "BC8", "BC8-0") == 1)
  {
    ;// текст
  }
  if (inplast(DEPTH(), "BC9-1") == 1)
  {
    ;// текст
  }
  if (inplast(DEPTH(), "BC10-0", "BC10-1", "BC10-2") == 1)
  {
    ;// текст
  }
  if (inplast(DEPTH(), "BC11-1", "BC11-2") == 1)
  {
```

```

    ;// текст
  }
}

```

В таком шаблоне, например, первое условие выполняется для пластов БС8 и БС8-0 и для интервала глубин между этими пластами. А для интервала глубин между группами (например, между БС9-1 и БС10-0) не выполняется ни какое условие.

Чтобы условия выполнялись для групп пластов и для интервалов между группами, надо в диалоге **Условия для Пластов** отметить **Заполнять дырки между пластами**. Тогда шаблон формулы примет вид:

```

formula Группы_пластов (*ZONEGRP) ()
{
  *ZONEGRP = 0;
  if (inplast(DEPTH(), "БС8", "БС8-0") != 0)
  {
    *ZONEGRP = 1;
  }
  if (inplast(DEPTH(), "БС9-1") != 0)
  {
    *ZONEGRP = 2;
  }
  if (inplast(DEPTH(), "БС10-0", "БС10-1", "БС10-2") != 0)
  {
    *ZONEGRP = 3;
  }
  if (inplast(DEPTH(), "БС11-1", "БС11-2", "БС12-1") != 0)
  {
    *ZONEGRP = 4;
  }
}
// в кривой *ZONEGRP заполняются "дырки" между группами
formula Заполнить_дырки() (&*ZONEGRP )
{
  if( *ZONEGRP == 0 )
  {
    icur = CUR_CRV_POINT();
    i = icur-1;
    while( i>=0 && *ZONEGRP == 0 )
    {
      if( *ZONEGRP [i] > 0 )
      {
        *ZONEGRP = *ZONEGRP [i];
      }
      i = i-1;
    }
  }
}

formula По_группам_пластов() (*ZONEGRP)
{
  if( *ZONEGRP == 1 ) // "БС8", "БС8-0" и до кровли "БС9-1"
  {
    ;// Текст
  }
  if( *ZONEGRP == 2 ) // "БС9-1" и до кровли "БС10-0"
  {
    ;// Текст
  }
}

```

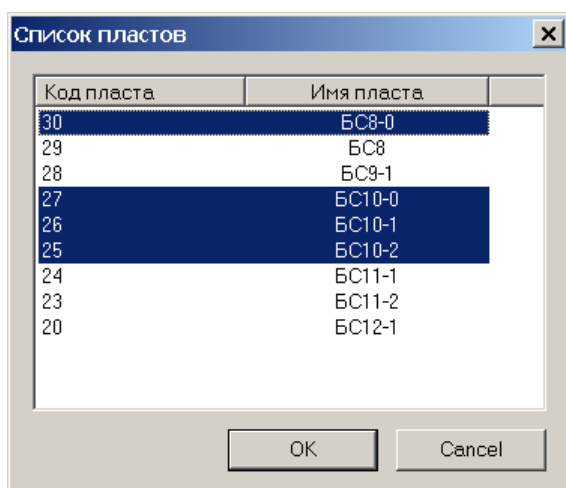
```

}
if( *ZONEGRP == 3 ) // "BC10-0", "BC10-1", "BC10-2" и до кровли "BC11-1"
{
    ;// Текст
}
if( *ZONEGRP == 4 ) // "BC11-1", "BC11-2", "BC12-1" и до конца
{
    ;// Текст
}
}
}

```

Т.е. если, например, между нижним пластом группы **BC10** и верхним пластом следующей группы **BC11** будет интервал глубин не относящийся ни к одному пласту, то для этого интервала будет выполняться условие группы пластов **BC10**

Команда **Редактирование/Вставить заготовку/Пласты текущей скважины**

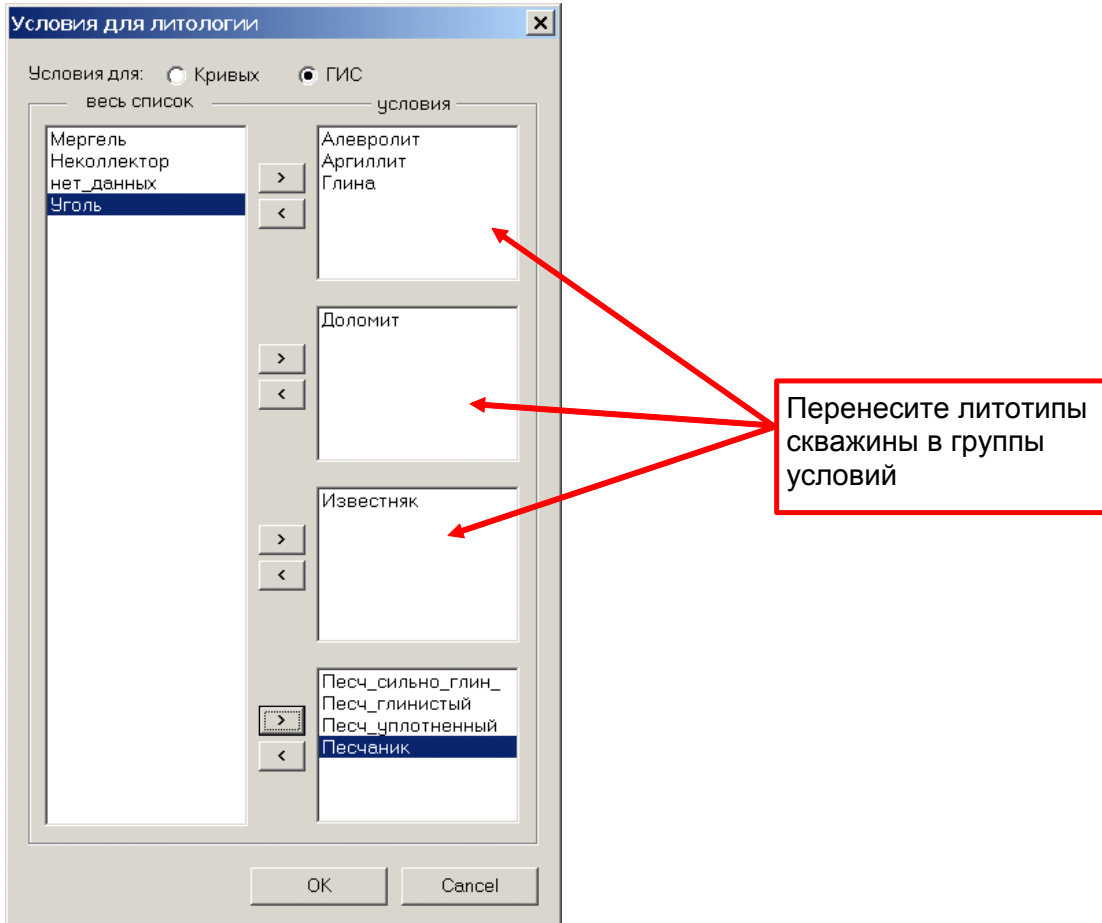


Команда используется для заполнения списка пластов в функции **inplast()**

Результат:

```
"BC8-0", "BC10-0", "BC10-1", "BC10-2"
```

Команды **Редактирование/Вставить заготовку/Условия для литологии** и **Редактирование/Вставить заготовку/Условия для насыщения**



Результат

```

if ( #LOB == Алевролит || #LOB == Аргиллит || #LOB == Глина )
{
  ;// текст
}

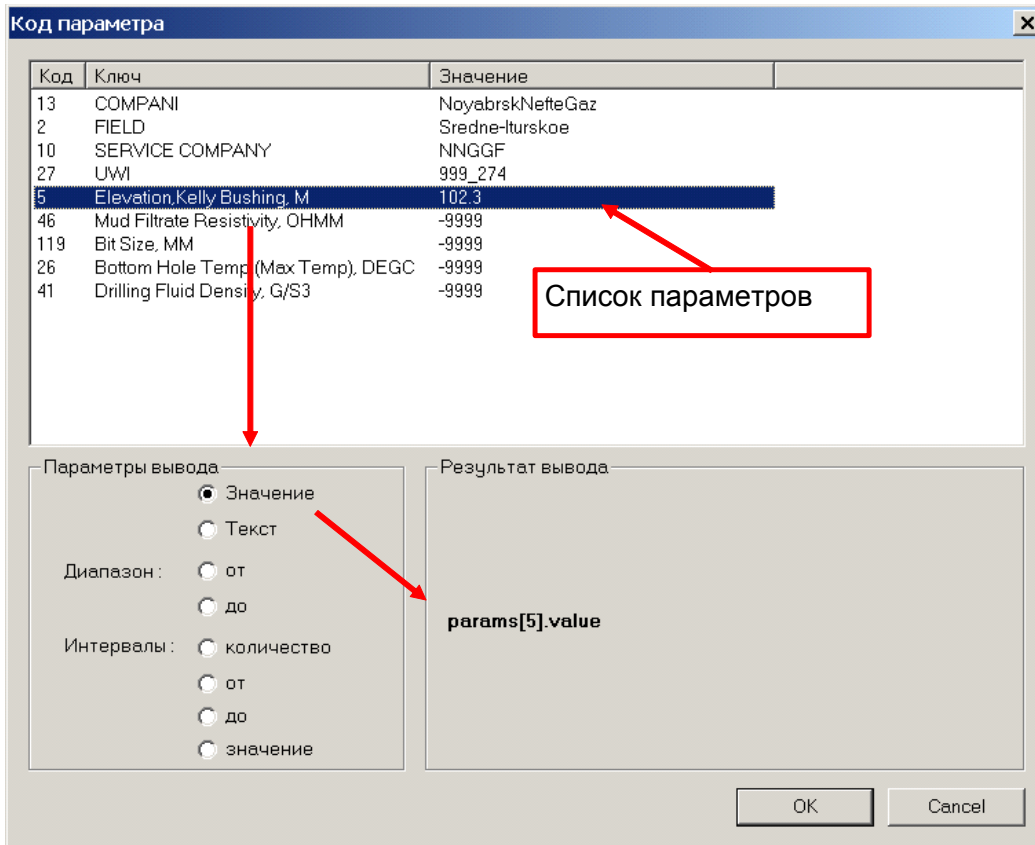
if ( #LOB == Доломит )
{
  ;// текст
}

if ( #LOB == Известняк )
{
  ;// текст
}

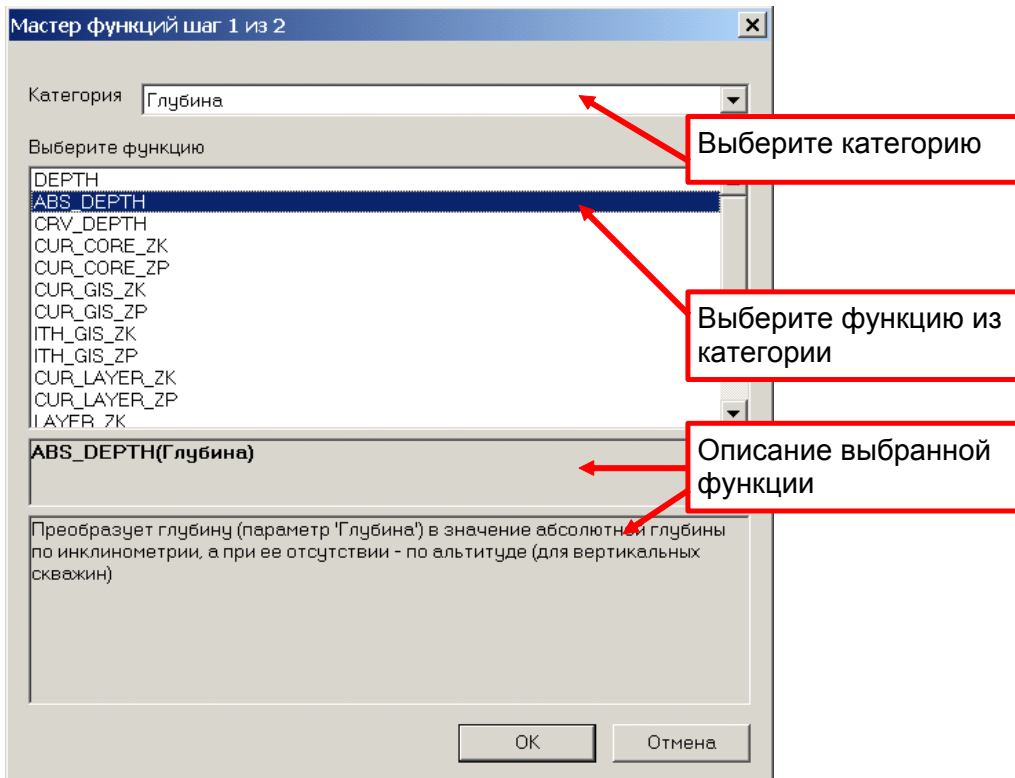
if ( #LOB == Песч_сильно_глин_
  || #LOB == Песч_глинистый
  || #LOB == Песч_уплотненный
  || #LOB == Песчаник )
{
  ;// текст
}

```

Команда **Редактирование/Вставить заготовку/Код параметра скважины**. В формулах, использующих данные, хранящиеся в параметрах скважины, для упрощения работы со специфическим синтаксисом параметров в ГеоСи.



Команда **Редактирование/Вставить заготовку/Встроенные функции** вызывает мастер вставки функции в формулу



На следующем шаге Вам будет предложено ввести значения параметров выбранной функции (если функция имеет параметры). Например, для **ABS_DEPTH**:

Аргументы функции

ABS_DEPTH

(Глубина) 2000

Возвращает число

OK Cancel

Результат:

ABS_DEPTH (2000)

Встроенные функции и процедуры (алфавитный перечень)

Имя функции	Аргументы	Назначение	Возвращаемое значение
_KAV НОВОЕ!	()	формирование текста с кавычками	символ двойной кавычки "
_NL НОВОЕ!	()	формирование многострочного текста	символ перехода на новую строку
_TAB НОВОЕ!	()	формирование текста с табуляциями	символ табуляции
abs	(число)	вычисляет модуль (абсолютное значение) числа	неотрицательное число
ABS_DEPTH	(число)	преобразует значение глубины в значение абсолютной глубины по инклинометрии или альтитуде вертикальной скважины	число
acos	(число от 0 до 1)	вычисляет арккосинус числа в радианах	число от 0 до ПИ
arccos	(число от 0 до 1)	вычисляет арккосинус числа в радианах	число от 0 до ПИ
arcsin	(число от 0 до 1)	вычисляет арксинус числа в радианах	число от - ПИ/2 до ПИ/2
arctg	(число)	вычисляет арктангенс числа в радианах	число от - ПИ/2 до ПИ/3
asin	(число от 0 до 1)	вычисляет арксинус числа в радианах	число от - ПИ/2 до ПИ/2
atan	(число)	вычисляет арктангенс числа в радианах	число от - ПИ/2 до ПИ/3
atan2	(число X, число Y)	вычисляет арктангенс от Y/X числа в радианах	число от 0 до 2*ПИ
atof	("строка")	преобразует строку, в которой записано число, в это число	число
atoi	("строка")	преобразует строку, в которой записано целое число, в это число	целое число

auto_components	(n, C1, C2,..., Cn, P11, P12, ..., P1n, V1, P21, P22, ..., P2n, V2, ..., max_err)	определяет компонентный состав породы или флюида в точке по показаниям нескольких методов и средним значениям показаний этих методов в «чистой» породе или флюиде каждого типа. Рекомендуется для большого числа компонент	неотрицательное число (относительную ошибку) или NOVAL()
chart2n	(двумерная константа, число X, число Y)	интерполирует номер узла или значение 4-й колонки регулярной палетки из 3(4) колонок	число или NOVAL()
chartof	(двумерная константа, число X, число Y)	интерполирует значение ветви регулярной палетки из 3(4) колонок	число или NOVAL()
components	(n, C1, C2,..., Cn, P11, P12, ..., P1n, V1, P21, P22, ..., P2n, V2, ...)	определяет компонентный состав породы или флюида в точке по показаниям нескольких методов и средним значениям показаний этих методов в «чистой» породе или флюиде каждого типа. Рекомендуется для небольшого числа компонент	неотрицательное число (относительную ошибку) или NOVAL()
components_lim НОВОЕ!	(n, C1, C2,..., Cn, min1, min2..., minn, max1, max2..., maxn, P11, P12, ..., P1n, V1, P21, P22, ..., P2n, V2, ...)	определяет компонентный состав породы или флюида в точке по показаниям нескольких методов и средним значениям показаний этих методов в «чистой» породе или флюиде каждого типа с учетом ограничений. Рекомендуется всегда.	неотрицательное число (относительную ошибку) или NOVAL()
convert_units НОВОЕ!	(число, "старая ед.измерения", "новая ед.измерения ")	преобразует единицы измерения из «старых» в «новые»	число
ConvertLas	()	вызывает программу ConvertLas.exe (только после SaveAsLas)	ничего
cos	(угол в радианах)	вычисляет косинус угла	число от 0 до 1
CRV_DEPTH	(целый номер точки)	преобразует номер точки кривой (от кровли расчета) в значение глубины	Число
CUR_CORE или CUR_CORE_USE_0	("название параметра керна")	возвращает среднее значение параметра керна в интервале текущего пропластка или точки кривой (USE_0 - учитывать нули)	число или NOVAL()
CUR_CORE_ERR или CUR_CORE_ERR_USE_0	("название параметра керна")	возвращает среднеквадратичное отклонение параметра керна в интервале текущего пропластка или точки кривой (USE_0 - учитывать нули)	число или NOVAL()

CUR_CORE_MAX или CUR_CORE_MAX_USE_0	("название параметра керна")	возвращает максимальное значение параметра керна в интервале текущего пропластка или точки кривой (USE_0 - учитывать нули)	число или NOVAL ()
CUR_CORE_MIN или CUR_CORE_MIN_USE_0	("название параметра керна")	возвращает минимальное значение параметра керна в интервале текущего пропластка или точки кривой (USE_0 - учитывать нули)	число или NOVAL ()
CUR_CORE_NUM или CUR_CORE_NUM_USE_0	("название параметра керна")	возвращает количество замеров параметра керна в интервале текущего пропластка или точки кривой (USE_0 - учитывать нули)	число или NOVAL ()
CUR_CORE_ZK	()	возвращает кровлю текущего интервала долбления керна	число или NOVAL ()
CUR_CORE_ZP	()	возвращает подошву текущего интервала долбления керна	число или NOVAL ()
CUR_CRV_POINT	()	возвращает номер текущей точки кривой (от кровли расчета)	целое число
CUR_GIS_POINT	()	возвращает номер текущего пропластка (от кровли расчета)	целое число
CUR_GIS_ZK	()	возвращает кровлю текущего пропластка	число
CUR_GIS_ZP	()	возвращает подошву текущего пропластка	число
CUR_LAYER_ZK	()	возвращает кровлю текущего стратиграфического пласта	число
CUR_LAYER_ZP	()	возвращает подошву текущего стратиграфического пласта	число
CUR_TEXT	(целый тип описания)	возвращает текст текущего интервала описания указанного типа	строка
CUR_TEXT_CODE	(целый тип описания)	возвращает код (литологию) текущего интервала описания указанного типа	целое число или NOVAL ()
CUR_TEXT_ZK	(целый тип описания)	возвращает кровлю текущего интервала описания указанного типа	число или NOVAL ()
CUR_TEXT_ZP	(целый тип описания)	возвращает подошву текущего интервала описания указанного типа	число или NOVAL ()
DeleteLog	(*Кривая1 [], *Кривая2 [], ...)	удаляет перечисленные кривые	ничего (subroutine)
DEPTH	()	возвращает текущую глубину	число
E		это не функция, а встроенная константа (без скобок), равна e	2.7...
empty_protocol НОВОЕ!	()	очищает протокол (удаляет файл протокола)	1 при успехе, иначе - 0
Execute НОВОЕ!	("запрос к БД")	выполняет запрос на изменение данных БД	ничего
exp	(число)	вычисляет экспоненту числа	положительное число
FIELD	()	возвращает текущее месторождение	строка

Gauss	(x1, x2,..., a11, a12,... b1, a21, a22,..., b2,...)	решает систему линейных уравнений (в начале идут переменные, потом коэффициенты уравнений по строкам). Эту функцию можно вызывать только из пользовательских функций, в которые x1, x2,... переданы в качестве параметров	1 при успехе, иначе - 0
GetLayerCode	()	возвращает код текущего стратиграфического пласта	целое число или NOVAL()
GetLayerCodeOfType	(целый тип пласта)	возвращает код текущего стратиграфического пласта заданного типа	целое число или NOVAL()
GetLayerName	()	возвращает имя текущего стратиграфического пласта	строка или ""
GetLayerNameOfType	(целый тип пласта)	возвращает имя текущего стратиграфического пласта заданного типа	строка или ""
GIS_k Plastam НОВОЕ!	(целый тип пласта, max сдвиг по глубине для отождествления)	выравнивает границы пропластков к границам пластов	ничего
GIS_ZK	()	возвращает кровлю интервала разбиения на пропластки	число или NOVAL()
GIS_ZP	()	возвращает подошву интервала разбиения на пропластки	число или NOVAL()
GNK НОВОЕ!	("имя/код пласта")	возвращает глубину ГНК пласта вдоль ствола	число или NOVAL()
graph2x	(двумерная константа, число Y)	интерполирует X по Y на палетке из двух колонок	число или NOVAL()
graph2y	(двумерная константа, число X)	интерполирует Y по X на палетке из двух колонок	число или NOVAL()
grid2x	(двумерная константа, число Y, число K)	интерполирует X по Y и ближайшей к K ветви палетки из 3x колонок	число или NOVAL()
grid2x_i	(двумерная константа, число Y, число K)	интерполирует X по Y и двум ближайшим к K ветвям палетки из 3x колонок	число или NOVAL()
grid2y	(двумерная константа, число X, число K)	интерполирует Y по X и ближайшей к K ветви палетки из 3x колонок	число или NOVAL()
grid2y_i	(двумерная константа, число X, число K)	интерполирует Y по X и двум ближайшим к K ветвям палетки из 3x колонок	число или NOVAL()
gridof	(двумерная константа, число X, число Y)	возвращает значение ближайшего к (X, Y) узла палетки из 3x колонок	число или NOVAL()
gridof_i	(двумерная константа, число X, число Y)	интерполирует значение палетки из 3x колонок	число или NOVAL()
GVK НОВОЕ!	("имя/код пласта")	возвращает глубину ГVK пласта вдоль ствола	число или NOVAL()
Inclinometry	()	рассчитывает абсолютные отметки и, если не рассчитана, инклинометрию	ничего (subroutine)
inplast	("строка1", "строка2", ..., Глубина)	проверяет, попадает ли Глубина между кровлей верхнего и подошвой нижнего из перечисленных пластов	1 если попадает, иначе - 0

insidepolygon	(x, y, x1, y1, x2, y2, x3, y3, ...)	проверяет, попадает ли точка (x,y) в многоугольник с углами (x1,y1), (x2,y2), (x3,y3), ...	1 если попадает, иначе - 0
interpol_2lines	(x,y, v1, x11,y11, x12,y12, v2, x21,y21,x22,y22)	интерполирует значение в точке (x,y) между двумя отрезками: (x11,y11 - x12,y12) со значением v1 и (x21,y21 - x22,y22) со значением v2	число или NOVAL()
is_protocol_empty НОВОЕ!	()	проверяет, пуст ли файл протокола	1 если пуст, иначе - 0
ITH_CRV_POINT	(глубина в м)	возвращает номер точки кривой на указанной глубине	целое число или NOVAL()
ITH_GIS_POINT	(глубина в м)	возвращает номер пропластка на указанной глубине	целое число или NOVAL()
ITH_GIS_ZK	(глубина в м)	возвращает кровлю пропластка на указанной глубине	число или NOVAL()
ITH_GIS_ZP	(глубина в м)	возвращает подошву пропластка на указанной глубине	число или NOVAL()
LAYER	("имя/код пласта")	проверяет, попадает ли текущая точка между кровлей подошвой указанного пласта	1 если попадает, иначе - 0
LAYER_ZK	("имя/код пласта")	возвращает кровлю указанного пласта	число или NOVAL()
LAYER_ZP	("имя/код пласта")	возвращает подошву указанного пласта	число или NOVAL()
lg	(положительное число)	вычисляет десятичный логарифм	число или NOVAL()
ln	(положительное число)	вычисляет натуральный логарифм	число или NOVAL()
log	(положительное число)	вычисляет натуральный логарифм	число или NOVAL()
log10	(положительное число)	вычисляет десятичный логарифм	число или NOVAL()
MakeExtremums	(*Кривая1[], *Кривая2[])	преобразует Кривую1 в Кривую2, удаляя все точки, кроме локальных минимумов и максимумов	целое число экстремумов
MakeGIS	(*СтупенчатаяКривая[])	создает разбиение на пропластки по ступенчатой кривой	целое число пропластков или NOVAL()
max	(число, число)	вычисляет максимум двух чисел	число
max2	(число, число, результат)	вычисляет максимум двух чисел	ничего (subroutine)
maxA	(*Кривая[])	вычисляет максимум кривой	число или NOVAL()
min	(число, число)	вычисляет минимум двух чисел	число
min2	(число, число, результат)	вычисляет минимум двух чисел	ничего (subroutine)
minA	(*Кривая[])	вычисляет минимум кривой	число или NOVAL()
NOVAL	()	возвращает признак отсутствия значения	-9999.99
NUM_CRV_POINTS	()	возвращает число точек кривой в интервале и с шагом расчета	целое число

NUM_GIS_POINTS	()	возвращает число пропластков в интервале расчета	целое число или NOVAL()
NULL		это не функция, а встроенная константа (без скобок), признак отсутствия значения	NOVAL(), т.е. -9999.99
PI		это не функция, а встроенная константа (без скобок), равна ПИ	3.14...
Planshet НОВОЕ!	("путь к шаблону планшета pln") или ("путь к шаблону планшета pln", "путь к выходной папке")	формирует файл планшета по заданному шаблону и запускает Планшет	ничего
Plasty_k_GIS НОВОЕ!	(целый тип пласта, max сдвиг по глубине для отождествления)	выравнивает границы пластов к границам пропластков	ничего
pow	(неотрицательное_число, степень)	возводит число в степень (аналогично оператору ^)	неотрицательное число
protocol_path НОВОЕ!	("строка1", "строка2", ...)	формирует имя файла протокола соединением строк-аргументов и запоминает для использования в protocol	ничего
Query НОВОЕ!	("запрос к БД")	выполняет запрос на чтение числа из БД	число
QueryStr НОВОЕ!	("запрос к БД")	выполняет запрос на чтение строки из БД	строка
Report НОВОЕ!	("путь к шаблону hid") или ("путь к шаблону hid", "путь к выходной папке")	формирует файл текстового документа-отчета по заданному шаблону	
round	(число)	вычисляет целую часть числа	целое число
SaveAsLAS	(*Кривая1[], *Кривая2[], ...)	сохраняет Кривую1, Кривую2, ... в LAS-файл с именем WELL() в той же папке, где находится файл базы	ничего
SaveGraphAsLAS	(*Кривая1[], *Кривая2[], ..., *Zabs[])	переводит Кривую1, Кривую2, ... в абсолютные глубины по кривой Zabs и сохраняет в LAS-файл с именем WELL()+"(DEPTH="+Zabs+")" в той же папке, где находится файл базы	ничего
Segmentation	(*Кривая1[], *Кривая2[], ...)	создает разбиение на пропластки по комплексу кривых	ничего
sin	(угол в радианах)	вычисляет синус угла	число от 0 до 1
sizeof НОВОЕ!	(число) или ("строка")	вычисляет размер аргумента (длину числа в байтах, длину строки в символах)	неотрицательное целое число
SortGIS НОВОЕ!	()	упорядочивает пропластки по глубине	ничего
SortLayerOfType НОВОЕ!	(целый тип пласта)	упорядочивает пласты по глубине	ничего
sqrt	(неотрицательное число)	вычисляет квадратный корень числа	неотрицательное число
STEP	()	возвращает шаг расчета	неотрицательное число

strcmp	("строка1", "строка2")	сравнивает строки по алфавитному порядку (все заглавные буквы предшествуют всем строчным)	-1, если строка1 предшествует строке2, 0, если они совпадают, 1 - иначе.
strfrom НОВОЕ!	("строка1", целое число)	выделяет левую или правую подстроку указанной длины	левая, если число >=0, правая - иначе.
stricmp	("строка1", "строка2")	сравнивает строки по алфавитному порядку без учета различий заглавных и строчных букв	-1, если строка1 предшествует строке2, 0, если они совпадают, 1 - иначе.
strlen НОВОЕ!	("строка")	возвращает длину строки	неотрицательное целое число
strncmp НОВОЕ!	("строка1", "строка2", натуральное число)	сравнивает левые подстроки указанной длины по алфавитному порядку без учета различий заглавных и строчных букв	-1, если строка1 предшествует строке2, 0, если они совпадают, 1 - иначе.
strrstr НОВОЕ!	("строка1", "строка2")	выделяет последнюю подстроку строки1, которая начинается со строки2	строка
strrstri НОВОЕ!	("строка1", "строка2")	выделяет последнюю подстроку строки1, которая начинается со строки2 без учета различий заглавных и строчных букв	строка
strstr НОВОЕ!	("строка1", "строка2")	выделяет первую подстроку строки1, которая начинается со строки2	строка
strstri НОВОЕ!	("строка1", "строка2")	выделяет первую подстроку строки1, которая начинается со строки2 без учета различий заглавных и строчных букв	строка
tan	(угол в радианах)	вычисляет тангенс угла	число
tg	(угол в радианах)	вычисляет тангенс угла	число
VNK НОВОЕ!	("имя/код пласта")	возвращает глубину ВНК пласта вдоль ствола	число или NOVAL ()
WELL	()	возвращает номер текущей скважины	строка
ZK	()	возвращает кровалью интервала расчета	число
ZP	()	возвращает подошву интервала расчета	число