

Газовый цех

Автоматизированная система учета

РАСХОДА ГАЗА: ДОМЕННОГО, КОКСОВОГО, ПРИРОДНОГО

**РУКОВОДСТВО ПРОГРАММИСТА**

**Программный модуль учета газа**

на 22 листах

СОГЛАСОВАНО:

Начальник УВСИТЦУ

\_\_\_\_\_ К.С.Теличко

«\_\_\_» \_\_\_\_\_ 2017 г.

Челябинск 2017

## **ЛИСТ СОГЛАСОВАНИЯ**

к Руководству программиста

СОГЛАСОВАНО:

УВСИТЦУ:

Начальник отдела автоматизации

И.Н.Резепин

Начальник бюро программирования

А.А.Загиров

Начальник бюро ДПиУЭ

А.В.Суковичин

РАЗРАБОТАЛ:

Математик

Н.А.Иванов

## Оглавление

1. Основная информация.....	4
2. Алгоритм работы программы MathTU .....	4
3. Инициализация функции расчета расхода ДГ,КГ,ПГ.....	6
4. Описание структуры программного модуля.....	7
4.1. Глобальные константы.....	7
4.2. Описание функций .....	7
4.2.4. Функция расчета молярной доли компонентов $X$ .....	10
4.2.5. Функция расчета псевдо-критической температуры $T_{nk}$ .....	11
4.2.6. Функция расчета псевдо-критического давления $P_{nk}$ .....	11
4.2.7. Функция расчета абсолютной температуры $T_a$ .....	11
4.2.8. Функция расчета абсолютного давления $P_a$ .....	11
4.2.9. Функция расчета фактора сжимаемости в рабочих условиях $Z$ .....	12
4.2.10. Функция расчета фактора сжимаемости в стандартных условиях $Z_c$ .....	12
4.2.11. Функция расчета плотности при стандартных условиях $Plot_c$ .....	12
4.2.12. Функция расчета плотности в рабочих условиях $Plot$ .....	13
4.2.13. Функция расчета диаметра ИТ в рабочих условиях $D_{it}$ .....	13
4.2.14. Функция расчета диаметра СУ в рабочих условиях $D_{cy}$ .....	13
4.2.15. Функция расчета относительного диаметра отверстия СУ при рабочей температуре $Otn_D$ .....	14
4.2.16. Функция расчета перевода перепада давления в Па $dP_{pr}$ .....	14
4.2.17. Функция расчета коэффициента скорости входа $E$ .....	14
4.2.18. Функция расчета коэффициента расширения $Eps$ .....	14
4.2.19. Функция расчета коэффициента истечения $C$ .....	15
4.2.20. Функция расчета коэффициента поправки на закругление входной кромки СУ $Kn$ .....	15
4.2.21. Функция расчета вязкости $U$ .....	15
4.2.22. Функция расчета коэффициента шероховатости $Kw$ .....	15
4.2.23. Функция расчета критерия Рейнольдса $Re$ .....	16
4.2.24. Функция расчета объёмного расхода среды приведённого к стандартным условиям $Qc$ .....	16
5. Список используемой литературы .....	17
Приложение №1.....	18

## **1. Основная информация**

Автоматизированная система контроля и учета энергоресурсов АСУ ЭНЕРГО имеет в своем составе одну базу данных (БД), которая хранит всю информацию о расходах энергоносителей. БД представляет собой реляционную нераспределенную базу данных, разработанную при помощи пакета программ Sybase SQL Anywhere 5.00. В качестве носителя информации используется жесткий диск сервера системы. Диспетчер данных также хранится на данном ПК.

Для корректного функционирования базы данных необходимы следующие условия:

- компьютер с характеристиками не ниже Celeron 1800, RAM 256Mb, HD 10Gb;
- операционная среда QNX 4.25;
- установленное стандартное программное обеспечение для работы с сервером базы данных (БД) Sybase SQL Anywhere 5.00;
- настроенная по протоколу TCP/IP связь с сервером базы данных

Для работы всей математики программы задействуется порядка 17 мбайт ОЗУ.

Программа написана на объектно-ориентированном языке программирования Watcom C++10.6 с использованием стандартных методов и функций данного языка. Программа работает в ОС QNX4.25.

В приложении №1 указаны примеры листинга программы.

## **2. Алгоритм работы программы MathTU**

Вся информация с датчиков поступает на устройство сбора данных (УСД) далее программным кодом (драйвер) при помощи программы Transport заносится в поле диспетчера данных (ДД) №1. Далее программа Normal нормализует данные из поля ДД №1 и переправляет их в поле ДД №2.

За работу математики отвечает программа MathTU, которая выполняет следующий алгоритм действий:

- 1) Запрашивает с базы данных (БД) формулы для расчета, константы, поля для вывода информации.

- 2) Регистрируется в ДД в поле №2 (в нём содержатся все нормализованные данные с датчиков) и №255 (передает № полей ДД, в которые нужно будет переправить обработанные программой MathTU данные). Указывает значения полей данных, которые ему необходимы для расчета данных формул (указано в БД), переходит в режим ожидания прокси-сигнала (сообщение о наличии значения в поле).
- 3) Как только приходит прокси-сигнал о том, что в поле появилось значение или если сработал таймер (периодическая проверка состояния поля), получает значение с поля данных.
- 4) Распаковывает данные.
- 5) Рассчитывает физические величины для точек учета и групп по формулам, определенным в базе данных для каждой конкретной точки учета или группы.
- 6) Выбранная из базы данных формула передается в функцию `calc_func()`, которая находится в файле `/energo/src/MathBlock/calc.cpp`, эта функция проверяет соответствует ли данная формула проинициализированным функциям в файле `/energo/src/MathBlock/func.h`, если да, то вызывает такую функцию из файла `/energo/src/MathBlock/func.cpp`.
- 7) Упаковывает данные в структуры массивов данных двух типов:
  - Структура `IMPULS`, которая формирует точку учета (номера счетчика, количество накопленных импульсов, номер группы учета, статус канала, ошибки, выход за пределы, наличие обрыва связи, формулы, поля: 0-датчик, 1-точка учета, 2-группа, 3-телесигнализация). Тело структуры находится в файле `/energo/src/MathBlock/ktsern.h`.
  - Структура `POLE9`, которая содержит номера полей в которые математика отправила данные и размеры этих данных. Тело структуры находится в файле `/energo/src/MathBlock/ktsern.h`.

Далее программа MathTU отправляет структуры 1ого типа в указанные поля (№11-№224) и создает поле ДД №9, в которое отправляет структуру 2ого типа.

### 3. Инициализация функции расчета расхода ДГ,КГ,ПГ

Для того чтобы программа MathTU обработала новую функцию расчета расхода доменного, коксового и природного газов, её нужно инициализировать.

Тело функции расчета расхода газа добавлено в файл который находится на ПК с функциями приема и обработки данных газового цеха: /energo/src/MathBlock/func.cpp.

На выходе данные сформированы в виде функций: PG\_2016 (12 входных параметров), DG\_2016 (16 входных параметров), KG\_2016 (17 входных параметров). Название функции и все входные параметры описаны в БД.

Эти функции проинициализированы в следующих файлах:

1) /energo/src/MathBlock/calc.h

В структуре func\_t[] увеличен размер массива обрабатываемых входных параметров par[] до 17(максимальное количество входных параметров функции).

2) /energo/src/MathBlock/func.h

Объявлены функции (тип функции, данных, спецификатор extern): PG\_2016(), DG\_2016(), KG\_2016().

3) /energo/src/MathBlock/calc.cpp

В массиве nfunc[32] добавлены элементы: " PG\_2016, DG\_2016, KG\_2016 ".

В функции calc\_func добавлены строки:

```
if (strcmp(Func[n].name,"KG_2016")==0)
return
    KG_2016(Func[n].par[0], Func[n].par[1], Func[n].par[2], Func[n].par[3],
    Func[n].par[4], Func[n].par[5], Func[n].par[6], Func[n].par[7], Func[n].par[8],
    Func[n].par[9], Func[n].par[10], (int)Func[n].par[11], (int)Func[n].par[12],
    Func[n].par[13], Func[n].par[14], (int)Func[n].par[15], (int)Func[n].par[16] );
if (strcmp(Func[n].name,"DG_2016")==0)
return
    DG_2016(Func[n].par[0], Func[n].par[1], Func[n].par[2], Func[n].par[3],
    Func[n].par[4], Func[n].par[5], Func[n].par[6], Func[n].par[7], Func[n].par[8],
    Func[n].par[9], (int)Func[n].par[10], (int)Func[n].par[11], Func[n].par[12],
    Func[n].par[13], (int)Func[n].par[14], (int)Func[n].par[15] );
if (strcmp(Func[n].name,"PG_2016")==0)
return
    PG_2016(Func[n].par[0], Func[n].par[1], Func[n].par[2], Func[n].par[3],
    Func[n].par[4], Func[n].par[5], (int)Func[n].par[6], (int)Func[n].par[7],
    Func[n].par[8], Func[n].par[9], (int)Func[n].par[10], (int)Func[n].par[11] );
```

4) /energo/src/MathBlock

Скомпилирована программа MathTU, для этого набрана команда: “make MathTU”. После этого заменён файл MathTU из /energo/bin на новый из файла /energo/src/MathBlock. Далее перезапущена математика командой: “slay MathTU”.

## **4. Описание структуры программного модуля gaz2016**

### **4.1. Глобальные константы**

Константы, перечисленные в [1] пункт 4 определены как:

- Тип переменных: double.
- Столбцы таб. №4.1,4.2 объединены в массивы.

Константы, перечисленные в [1] пункт 2.1.1 таб. №1 определены как:

- Тип переменных: double.
- Значения таблицы объединены в двумерный массив Steel[26 строк][3 столбца].
- Значения первой строки [0,0,0].

### **4.2. Описание функций**

#### **4.2.1. Функция интерфейс для природного газа PG\_2016**

Входные данные для функции:

- **P\_izb** - избыточное давление в (кгс/см<sup>2</sup>);
- **P\_bar** - атмосферное давление в (кгс/см<sup>2</sup>);
- **T\_cel** - температура среды в (градусах Цельсия);
- **dP** - перепад давления в (кгс/м<sup>2</sup>);
- **D\_it\_20** - диаметр ИТ в (мм) при температуре 20 градусов Цельсия;
- **D\_cy\_20** - диаметр СУ в (мм) при температуре 20 градусов Цельсия;
- **nt** - номер материала ИТ;
- **nd** - номер материала СУ;
- **rn** - начальный радиус закругления входной кромки в (мм);
- **Ra** - коэффициент шероховатости;
- **data** - время эксплуатации РУ в (год);

- **method** - метод отбора давления (угловой = 0, трёхрадиусный = 1, фланцевый = 2).

Работа функции:

- тип функции: double;
- записывает входные данные в структуру;
- проводит проверку на соблюдение граничных условий (смотри [1] пункт 3), в случае ошибки производит запись в файл /energo/report при помощи команды `info.report("1","1","сообщение")`;
- проводит проверку на нулевое и отрицательное значение расхода, в случае успеха записывает в файл /energo/report/text сообщение об ошибке, значение всех констант и расчетные значения функций.

Выходные данные:

- Возвращает значение объемного расхода, которое ей передает функция **Qc**.

#### 4.2.2. Функция интерфейс для доменного газа *DG\_2016*

Входные данные для функции:

- **CH4\_dg** - объемная доля метана в (%);
- **CO2\_dg** - объемная доля диоксида углерода в (%);
- **H2\_dg** - объемная доля водорода в (%);
- **CO\_dg** - объемная доля монооксида углерода в (%);
- **P\_izb** - избыточное давление в (кгс/см<sup>2</sup>);
- **P\_bar** - атмосферное давление в (кгс/см<sup>2</sup>);
- **T\_cel** - температура среды в (градусах Цельсия);
- **dP** - перепад давления в (кгс/м<sup>2</sup>);
- **D\_it\_20** - диаметр ИТ в (мм) при температуре 20 градусов Цельсия;
- **D\_cy\_20** - диаметр СУ в (мм) при температуре 20 градусов Цельсия;
- **nt** - номер материала ИТ;
- **nd** - номер материала СУ;
- **rn** - начальный радиус закругления входной кромки в (мм);



- **Ra** - коэффициент шероховатости;
- **data** - время эксплуатации РУ в (год);
- **method** - метод отбора давления (угловой = 0, трёхрадиусный = 1, фланцевый = 2).

Работа функции:

- тип функции: double;
- записывает входные данные в структуру;
- проводит проверку на соблюдение граничных условий (смотри [1] пункт 3), в случае ошибки производит запись в файл /energo/report при помощи команды info.report("1","1","сообщение");
- проводит проверку на нулевое и отрицательное значение расхода, в случае успеха записывает в файл /energo/report/text сообщение об ошибке, значение всех констант и расчетные значения функций.

Выходные данные:

- Возвращает значение объемного расхода, которое ей передает функция **Qc**.

#### 4.2.3. Функция интерфейс для коксового газа **KG\_2016**

Входные данные для функции:

- **CH4\_dg** - объемная доля метана в (%);
- **CO2\_dg** - объемная доля диоксида углерода в (%);
- **H2\_dg** - объемная доля водорода в (%);
- **CO\_dg** - объемная доля монооксида углерода в (%);
- **CnHn\_kg** - объемная доля непрерывных углеводородов в (%);
- **P\_izb** - избыточное давление в (кгс/см<sup>2</sup>);
- **P\_bar** - атмосферное давление в (кгс/см<sup>2</sup>);
- **T\_cel** - температура среды в (градусах Цельсия);
- **dP** - перепад давления в (кгс/м<sup>2</sup>);
- **D\_it\_20** - диаметр ИТ в (мм) при температуре 20 градусов Цельсия;
- **D\_cy\_20** - диаметр СУ в (мм) при температуре 20 градусов Цельсия;

- **nt** - номер материала ИТ;
- **nd** - номер материала СУ;
- **rn** - начальный радиус закругления входной кромки в (мм);
- **Ra** - коэффициент шероховатости;
- **data** - время эксплуатации РУ в (год);
- **method** - метод отбора давления (угловой = 0, трёхрадиусный = 1, фланцевый = 2).

Работа функции:

- тип функции: double;
- записывает входные данные в структуру;
- проводит проверку на соблюдение граничных условий (смотри [1] пункт 3), в случае ошибки производит запись в файл /energo/report при помощи команды info.report("1","1","сообщение");
- проводит проверку на нулевое и отрицательное значение расхода, в случае успеха записывает в файл /energo/report/text сообщение об ошибке, значение всех констант и расчетные значения функций.

Выходные данные:

- Возвращает значение объемного расхода, которое ей передает функция **Qc**.

#### 4.2.4. Функция расчета молярной доли компонентов X

Входные данные:

- Значение типа integer, которое определяет, для какого газа проводить вычисления при помощи оператора переключения switch(int) - case (3 – ПГ, 5 – ДГ, 7 – КГ). Цифры соответствуют количеству составных компонент газа.
- Указатель на элемент массива double \*out\_x, размерностью (3 – ПГ, 5 – ДГ, 7 – КГ).

Работа функции:

- Тип функции double\*.
- Алгоритм расчета описан в [1] пункт 2.2.6.

Выходные данные:

- Функция на выходе выдает значение out\_x (тип: double).

#### **4.2.5. Функция расчета псевдо-критической температуры $T_{nk}$**

Входные данные:

- Функция принимает значение типа integer, которое определяет, для какого газа проводить вычисления при помощи оператора переключения switch(int) - case (5 – ДГ, 7 – КГ). Цифры соответствуют количеству составных компонент газа.

Работа функции:

- Алгоритм расчета описан в [1] пункт 2.2.8.

Выходные данные:

- Функция на выходе выдает значение типа: double.

#### **4.2.6. Функция расчета псевдо-критического давления $P_{nk}$**

Входные данные:

- Функция принимает значение типа integer, которое определяет, для какого газа проводить вычисления при помощи оператора переключения switch(int) - case (5 – ДГ, 7 – КГ). Цифры соответствуют количеству составных компонент газа.

Работа функции:

- Алгоритм расчета описан в [1] пункт 2.2.9.

Выходные данные:

- Функция на выходе выдает значение типа: double.

#### **4.2.7. Функция расчета абсолютной температуры $T_a$**

Работа функции:

- Алгоритм расчета описан в [1] пункт 2.2.5.

Выходные данные:

- Функция на выходе выдает значение типа: double.

#### **4.2.8. Функция расчета абсолютного давления $P_a$**

Работа функции:

- Алгоритм расчета описан в [1] пункт 2.2.4.

Выходные данные:

- Функция на выходе выдает значение типа: double.

#### **4.2.9. Функция расчета фактора сжимаемости в рабочих условиях $Z$**

Входные данные:

- Функция принимает значение типа integer, которое определяет, для какого газа проводить вычисления при помощи оператора переключения switch(int) - case (3 – ПГ, 5 – ДГ, 7 – КГ). Цифры соответствуют количеству составных компонент газа.

Работа функции:

- Алгоритм расчета описан в [1] пункт 2.2.10.
- В расчете для ПГ используется абсолютное давление в Мпа, учитывая это, во всех формулах для ПГ сделана поправка (значение которое передает функция  $P_a$  делим на миллион)  $P_a/1000000$ .

Выходные данные:

- Функция на выходе выдает значение типа: double.

#### **4.2.10. Функция расчета фактора сжимаемости в стандартных условиях $Z_c$**

Входные данные:

- Функция принимает значение типа integer, которое определяет, для какого газа проводить вычисления при помощи оператора переключения switch(int) - case (3 – ПГ, 5 – ДГ, 7 – КГ). Цифры соответствуют количеству составных компонент газа.

Работа функции:

- Алгоритм расчета описан в [1] пункт 2.2.11.

Выходные данные:

- Функция на выходе выдает значение типа: double.

#### **4.2.11. Функция расчета плотности при стандартных условиях $\rho_{sc}$**

Входные данные:

- Функция принимает значение типа integer, которое определяет, для какого газа проводить вычисления при помощи оператора переключения switch(int) - case (3 – ПГ, 5 – ДГ, 7 – КГ). Цифры соответствуют количеству составных компонент газа.

Работа функции:

- Алгоритм расчета описан в [1] пункт 2.2.13.
- В расчете для ПГ используется абсолютное давление в Мпа, учитывая это, во всех формулах для ПГ сделана поправка (значение которое передает функция  $P\_a$  делим на миллион)  $P\_a()/1000000$ .

Выходные данные:

- Функция на выходе выдает значение типа: double.

#### **4.2.12. Функция расчета плотности в рабочих условиях $Plot$**

Входные данные:

- Функция принимает значение типа integer, которое определяет, для какого газа проводить вычисления при помощи оператора переключения switch(int) - case (3 – ПГ, 5 – ДГ, 7 – КГ). Цифры соответствуют количеству составных компонент газа.

Работа функции:

- Алгоритм расчета описан в [1] пункт 2.2.14.
- В расчете для ПГ используется абсолютное давление в Мпа, учитывая это, во всех формулах для ПГ сделана поправка (значение которое передает функция  $P\_a$  делим на миллион)  $P\_a()/1000000$ .

Выходные данные:

- Функция на выходе выдает значение типа: double.

#### **4.2.13. Функция расчета диаметра ИТ в рабочих условиях $D\_it$**

Работа функции:

- Алгоритм расчета описан в [1] пункт 2.2.1.

Выходные данные:

- Функция на выходе выдает значение типа: double.

#### **4.2.14. Функция расчета диаметра СУ в рабочих условиях $D\_su$**

Работа функции:

- Алгоритм расчета описан в [1] пункт 2.2.1.

Выходные данные:

- Функция на выходе выдает значение типа: double.

#### **4.2.15.      Функция расчета относительного диаметра отверстия $S_U$ при рабочей температуре $Otn\_D$**

Работа функции:

- Алгоритм расчета описан в [1] пункт 2.2.1.

Выходные данные:

- Функция на выходе выдает значение типа: double.

#### **4.2.16.      Функция расчета перевода перепада давления в Па $dP\_pr$**

Работа функции:

- Алгоритм расчета описан в [1] пункт 2.2.4.

Выходные данные:

- Функция на выходе выдает значение типа: double.

#### **4.2.17.      Функция расчета коэффициента скорости входа $E$**

Работа функции:

- Алгоритм расчета описан в [1] пункт 2.2.3.

Выходные данные:

- Функция на выходе выдает значение типа: double.

#### **4.2.18.      Функция расчета коэффициента расширения $E_{ps}$**

Входные данные:

- Функция принимает значение типа integer, которое определяет, для какого газа проводить вычисления при помощи оператора переключения switch(int) - case (3 – ПГ, 5 – ДГ, 7 – КГ). Цифры соответствуют количеству составных компонент газа.

Работа функции:

- Алгоритм расчета описан в [1] пункт 2.2.16.
- В расчете для ПГ используется абсолютное давление в Мпа, учитывая это, во всех формулах для ПГ сделана поправка (значение которое передает функция  $P\_a$  делим на миллион)  $P\_a/1000000$ .
- Вычисляет показатель адиабаты  $k$  (см. [1] пункт 2.2.15) необходимый для расчета коэффициента расширения  $\varepsilon$ .

Выходные данные:

- Функция на выходе выдает значение типа: double.

#### **4.2.19.      Функция расчета коэффициента истечения $C$**

Входные данные:

- Функция принимает значение, которое возвращает функция **Re** (тип double).

Работа функции:

- Алгоритм расчета описан в [1] пункт 2.2.18.

Выходные данные:

- Функция на выходе выдает значение типа: double.

#### **4.2.20.      Функция расчета коэффициента поправки на закругление входной кромки $C_{\kappa}$**

Работа функции:

- Алгоритм расчета описан в [1] пункт 2.2.2.

Выходные данные:

- Функция на выходе выдает значение типа: double.

#### **4.2.21.      Функция расчета вязкости $\mu$**

Входные данные:

- Функция принимает значение типа integer, которое определяет, для какого газа проводить вычисления при помощи оператора переключения switch(int) - case (3 – ПГ, 5 – ДГ, 7 – КГ). Цифры соответствуют количеству составных компонент газа.

Работа функции:

- Алгоритм расчета описан в [1] пункт 2.2.7.
- В расчете для ПГ используется абсолютное давление в Мпа, учитывая это, во всех формулах для ПГ сделана поправка (значение которое передает функция  $P_a$  делим на миллион)  $P_a/1000000$ .

Выходные данные:

- Функция на выходе выдает значение типа: double.

#### **4.2.22.      Функция расчета коэффициента шероховатости $K_w$**

Входные данные:

- Функция принимает значение, которое возвращает функция **Re** (тип double).

Работа функции:

- Алгоритм расчета описан в [1] пункт 2.2.20.

Выходные данные:

- Функция на выходе выдает значение типа: double.

#### **4.2.23. Функция расчета критерия Рейнольдса $Re$**

Входные данные:

- Функция принимает значение типа integer, которое определяет, для какого газа проводить вычисления при помощи оператора переключения switch(int) - case (3 – ПГ, 5 – ДГ, 7 – КГ). Цифры соответствуют количеству составных компонент газа.

Работа функции:

- Алгоритм расчета описан в [1] пункт 2.2.22.

Выходные данные:

- Функция на выходе выдает значение типа: double.

#### **4.2.24. Функция расчета объёмного расхода среды приведённого к стандартным условиям $Q_c$**

Входные данные:

- Функция принимает значение типа integer, которое определяет, для какого газа проводить вычисления при помощи оператора переключения switch(int) - case (3 – ПГ, 5 – ДГ, 7 – КГ). Цифры соответствуют количеству составных компонент газа.

Работа функции:

- Алгоритм расчета описан в [1] пункт 2.2.23.

Выходные данные:

- Функция на выходе выдает значение типа: double.



## **5. Список используемой литературы**

1. Математическое обеспечение. Алгоритм расчета доменного, коксового и природного газов. Челябинск, 2017.
2. Общеотраслевые руководящие методические материалы по созданию и применению автоматизированных систем управления технологическими процессами в отраслях промышленности. 1986.
3. Автоматизированная система диспетчерского управления газообразным топливом на ОАО "Мечел" (Газовый цех) том 3. Программное обеспечение. Книга 3. Описание программ нижнего уровня (QNX). Челябинск, 2004.
4. Автоматизированная система контроля и учета энергоресурсов АСУ ЭНЕРГО. Описание организации информационной базы данных. Челябинск 2004.
5. Н. Н. Мартынов. Программирование для Windows на C/C++ том 1. ООО «Бином-Пресс», 2004.

## Приложение №1

### Описание констант.

```
//коэффициенты для расчета фактора сжимаемости
double W_a = 0.42748;
double W_b = 0.08664;
double B_pg[3] = {0.0436,0.0173,0.0728};

//температура при стандартных условиях (в Кельвинах)
double T_c = 293.15;

//давление при стандартных условиях (в Паскалях)
double P_c = 101325;

//универсальная газовая постоянная
double R = 8.31451;

//число Пи
double Pi = 3.14159;

//объёмная доля (i-ого компонента природного газа)
double r_pg[3]={99.09,0.79,0.12};

// температура критическая i-ого компонента доменного газа (в Кельвинах)
double T_k_dg[5] = {190.6,126.2,304.2,33.2,132.9};

// температура критическая i-ого компонента коксового газа (в Кельвинах)
double T_k_kg[7] = {190.6,126.2,304.2,33.2,132.9,154.6,493.1};

// давление критическое i-ого компонента доменного газа (в Паскалях)
double P_k_dg[5] = {4587579.2,3385108,7356294.4,1293414.4,3486156};

// давление критическое i-ого компонента коксового газа (в Паскалях)
double P_k_kg[7] = {4587579,3385108,7356294,1293414,3486156,5032190,4984192};

// фактор сжимаемости при стандартных условиях i-ого компонента природного
// газа
double z_c_pg[3] = {0.9981,0.9997,0.9947};

// фактор сжимаемости при стандартных условиях
double z_c_dg[5] = {0.9981,0.9997,0.9947,1.0006,0.9996};

// фактор сжимаемости при стандартных условиях i-ого компонента коксового газа
double z_c_kg[7] = {0.9981,0.9997,0.9947,1.0006,0.9996,0.9993,0.9537};

// молярная масса i-ого компонента природного газа (в кг/кмоль)
double M_pg[3] = {16.043,28.135,44.01};

// молярная масса i-ого компонента доменного газа (в кг/моль)
double M_dg[5] = {0.01604,0.02813,0.04401,0.00201,0.02801};
```

```

// молярная масса i-ого компонента коксового газа (в кг/моль)
double M_kg[7] = {0.01604,0.02813,0.04401,0.00201,0.02801,0.03199,0.06509};

// плотность при стандартных условиях i-ого компонента доменного газа (в кг/м³)
double Plot_c_dg[5] = {0.66692,1.16455,1.82954,0.083803,1.1644};

// плотность при стандартных условиях i-ого компонента коксового газа (в кг/м³)
double Plot_c_kg[7] = {0.66692,1.16455,1.82954,0.083803,1.1644,1.33022,2.435467};

// показатель адиабаты i-ого компонента доменного газа
double k_dg[5] = {1.295,1.4,1.285,1.405,1.4};

// показатель адиабаты i-ого компонента коксового газа
double k_kg[7] = {1.295,1.4,1.285,1.405,1.4,1.395,1.225};

```

### Инициализация функций.

```

double* x(int j, double *out_x);
double T_nk(int j);
double P_nk( int j);
double P_a();
double T_a();
double Z(int j);
double Z_c(int j);
double Plot_c(int j);
double Plot(int j);
double D_it();
double D_cy();
double Otn_D();
double dP_pr();
double E();
double Eps(int j);
double C(double pRe);
double Kn();
double U(int j);
double Kw(double pRe);
double Re(int j);
double Qc(int j);
double Qm(int j);

double PG_2016(double P_izb, double P_bar, double T_cel, double dP, double
D_it_20, double D_cy_20, int nt, int nd, double rn, double Ra, int data, int method);

double DG_2016(double CH4_dg, double CO2_dg, double H2_dg, double CO_dg,
double P_izb, double P_bar, double T_cel, double dP, double D_it_20, double D_cy_20,
int nt, int nd, double rn, double Ra, int data, int method);

double KG_2016(double CH4_kg, double CO2_kg, double H2_kg, double CO_kg,
double CnHn_kg, double P_izb, double P_bar, double T_cel, double dP, double
D_it_20, double D_cy_20, int nt, int nd, double rn, double Ra, int data, int method);

```

### Функция интерфейс доменного газа.

```
double DG_2016(double CH4_dg, double CO2_dg, double H2_dg, double CO_dg,  
double P_izb, double P_bar, double T_cel, double dP, double D_it_20, double D_cy_20,  
int nt, int nd, double rn, double Ra, int data, int method)
```

```
{  
    // запись входных параметров в структуру  
    double N2_dg      = 100 - (CH4_dg+CO2_dg+H2_dg+CO_dg);  
    Gaz.VC_r_dg[0] = CH4_dg;  
    Gaz.VC_r_dg[1] = N2_dg;  
    Gaz.VC_r_dg[2] = CO2_dg;  
    Gaz.VC_r_dg[3] = H2_dg;  
    Gaz.VC_r_dg[4] = CO_dg;  
    Gaz.VC_P_izb = P_izb;  
    Gaz.VC_P_bar = P_bar;  
    Gaz.VC_T_cel = T_cel;  
    Gaz.VC_dP = dP;  
    Gaz.VC_D_it_20 = D_it_20;  
    Gaz.VC_D_cy_20 = D_cy_20;  
    Gaz.VC_nt = nt;  
    Gaz.VC_nd = nd;  
    Gaz.VC_rn = rn;  
    Gaz.VC_Ra = Ra;  
    Gaz.VC_data = data;  
    Gaz.VC_method = method;  
  
    // проверка граничных условий, если произошёл выход за предел  
    // происходит запись в файл «energo/report»  
    if(D_cy_20<12.5)  
    {  
        info.Report("1","1","DG: ПАРАМЕТР НЕ ВХОДИТ В ДИАПАЗОН!  
D_cy_20(mm) >= 12.5 ! ");  
    }  
  
    if(D_it_20<50 || D_it_20>1000)  
    {  
        info.Report("1","1","DG: ПАРАМЕТР НЕ ВХОДИТ В ДИАПАЗОН!  
50 <= D_it_20(mm) <= 1000 ! ");  
    }  
  
    if(Otn_D()<0.1 || Otn_D()>0.75)  
    {  
        info.Report("1","1","DG: ПАРАМЕТР НЕ ВХОДИТ В ДИАПАЗОН!  
0.1 <= Otn_D <= 0.75 ! ");  
    }  
  
    if((dP_pr()/P_a())>0.25)  
    {  
        info.Report("1","1","DG: ПАРАМЕТР НЕ ВХОДИТ В ДИАПАЗОН!
```

```

dP/P <= 0.25 ! ! ");
    }

    if(T_a())>350 || T_a()<250)
    {
        info.Report("1","1","DG: ПАРАМЕТР НЕ ВХОДИТ В ДИАПАЗОН!
250 <= T(K) <= 350 ! ");
    }

    if((P_a()/1000000)>7.5 || (P_a()/1000000)<0.1)
    {
        info.Report("1","1","DG: ПАРАМЕТР НЕ ВХОДИТ В ДИАПАЗОН!
0.1 <= P_a(mPa) <= 7.5 ! ");
    }

    if(Otn_D()<=0.56)
    {
        if(Re(5)<5000)
        {
            info.Report("1","1","DG: ПАРАМЕТР НЕ ВХОДИТ В ДИАПАЗОН!
Re >= 5000, при Otn_D <= 0.56 ! ");
        }
    }
    else
    {
        if((int)method == 2)
        {
            if(Re(5)<170000*pow(Otn_D(),2)*D_it())
            {
                info.Report("1","1","DG: ПАРАМЕТР НЕ ВХОДИТ В
ДИАПАЗОН! Re >= 16000*(Otn_D^2), при Otn_D > 0.56 ! ");
            }
        }
        else
        {
            if(Re(5)<16000*pow(Otn_D(),2))
            {
                info.Report("1","1","DG: ПАРАМЕТР НЕ ВХОДИТ В
ДИАПАЗОН! Re >= 16000*(Otn_D^2), при Otn_D > 0.56 ! ");
            }
        }
    }

    // запуск функции расчета молярной доли компонентов ДГ
    double px[5];
    x(5,px);

    // проверка значения расхода, если меньше или равно нулю, то записывает
    значения всех констант и расчетных величин в файл «energo/report/text»
    if(Qc(5)<=0)
    {

```

```

        info.Report("/energo/report/text.txt","1","Error: Qc_DG = %f(m3/h) !!!\n
Vhod_const: \n CH4=%f, CO2=%f, H2=%f, CO=%f, N2=%f \n P_izb=%f(kJc/cm2),
P_bar=%f(kJc/cm2), T_cel=%f, dP=%f(kJc/m2), \n D_it_20=%f(mm), D_cy_20=%f(mm),
nt=%d, nd=%d, \n rn=%f(mm), Ra=%f, data=%d, method=%d
\n",Qc(5),CH4_dg,CO2_dg,H2_dg,CO_dg,N2_dg,P_izb,P_bar,T_cel,dP,D_it_20,D_cy_
20,nt,nd,rn,Ra,data,method);

```

```

        info.Report("/energo/report/text.txt","1","Rabota_func: \n x[CH4]=%f,
x[N2]=%f, x[CO2]=%f, x[H2]=%f, x[CO]=%f \n T_nk=%f(K), P_nk=%f(Pa), P_a=%f(Pa),
T_a=%f(K), \n Z=%f, Z_c=%f, Plot_c=%f(kJ/m3), Plot=%f(kJ/m3), \n D_it=%f(m),
D_cy=%f(m), Otn_D=%f, dP_pr=%f(Pa), \n E=%f, Eps=%f, Kn=%f, U=%f(Pa*c), Re=%f,
Kw(1000000)=%f \n
",px[0],px[1],px[2],px[3],px[4],T_nk(5),P_nk(5),P_a(),T_a(),Z(5),Z_c(5),Plot_c(5),Plot(5),
D_it(),D_cy(),Otn_D(),dP_pr(),E(),Eps(5),Kn(),U(5),Re(5),Kw(1000000));

```

```

        info.Report("/energo/report/text.txt","1","Global_const: \n W_a=%f,
W_b=%f, T_c=%f(K), P_c=%f(Pa), R=%f, Pi=%f, \n T_k_dg[CH4]=%f(K),
T_k_dg[N2]=%f, T_k_dg[CO2]=%f, T_k_dg[H2]=%f, \n T_k_dg[CO]=%f, \n
P_k_dg[CH4]=%f, P_k_dg[N2]=%f(Pa), P_k_dg[CO2]=%f, P_k_dg[H2]=%f, \n
P_k_dg[CO]=%f, \n
",W_a,W_b,T_c,P_c,R,Pi,T_k_dg[0],T_k_dg[1],T_k_dg[2],T_k_dg[3],T_k_dg[4],P_k_dg[
0],P_k_dg[1],P_k_dg[2],P_k_dg[3],P_k_dg[4]);

```

```

        info.Report("/energo/report/text.txt","1","z_c_dg[CH4]=%f, z_c_dg[N2]=%f,
z_c_dg[CO2]=%f, z_c_dg[H2]=%f, \n z_c_dg[CO]=%f, \n M_dg[CH4]=%f(kJ/mol),
M_dg[N2]=%f, M_dg[CO2]=%f, M_dg[H2]=%f, \n M_dg[CO]=%f, \n
Plot_c_dg[CH4]=%f(kJ/m3), Plot_c_dg[N2]=%f, Plot_c_dg[CO2]=%f, Plot_c_dg[H2]=%f,
\n Plot_c_dg[CO]=%f, \n k_dg[CH4]=%f, k_dg[N2]=%f, k_dg[CO2]=%f, k_dg[H2]=%f, \n
k_dg[CO]=%f \n
\n",z_c_dg[0],z_c_dg[1],z_c_dg[2],z_c_dg[3],z_c_dg[4],M_dg[0],M_dg[1],M_dg[2],M_dg
[3],M_dg[4],Plot_c_dg[0],Plot_c_dg[1],Plot_c_dg[2],Plot_c_dg[3],Plot_c_dg[4],k_dg[0],k
_dg[1],k_dg[2],k_dg[3],k_dg[4]);
    }

```

```

    return Qc(5);
}

```