



ТОМСКИЙ
ПОЛИТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ



СИСТЕМНЫЙ АНАЛИЗ ПРОЦЕССОВ ХИМИЧЕСКОЙ ТЕХНОЛОГИИ

РЕАЛИЗАЦИЯ РАСЧЕТА ХИМИКО- ТЕХНОЛОГИЧЕСКОЙ СИСТЕМЫ НА ЯЗЫКЕ PASCAL

доцент ОХИ ИШПР ТПУ, к.т.н.
Чузлов Вячеслав Алексеевич

2020



```
Unit UHTS;

interface

type
  TArrayOfDouble = array of double;
  TArrayOfArrOfDouble = array of array of double;

function get_comp_Cp(comp_count: integer; flows_count: integer;
                    flows_param: TArrayOfArrOfDouble): TArrayOfArrOfDouble;

function get_flow_Cp(comp_count: integer; flows_count: integer;
                    flows_composition: TArrayOfArrOfDouble;
                    comp_Cp: TArrayOfArrOfDouble): TArrayOfDouble;

procedure mixer(comp_count: integer; flows_count: integer; flows_param: TArrayOfArrOfDouble;
                flows_composition: TArrayOfArrOfDouble; var flow_rate, flow_Cp,
                flow_Temp: double; var flow_composition: TArrayOfDouble);

procedure heater(comp_count: integer; flows_count: integer; heater_flow_rate: double;
                 heater_flow_Temp: double; flow_rate: double; flow_Cp: double;
                 var flow_Temp: double);

procedure splitter(split_ratio: double; flow_rate: double; var flow_rate1,
                  flow_rate2: double);
```



```
function get_comp_Cp(comp_count: integer; flows_count: integer;  
                    flows_param: TArrOfArrOfDouble): TArrOfArrOfDouble;  
const  
    A: array of double = (0.869, 0.963);  
    B: array of double = (1e-5, 1.1e-5);  
  
var  
    i, j: integer;  
  
begin  
    SetLength(Result, flows_count);  
  
    for i := 0 to flows_count-1 do  
        SetLength(Result[i], comp_count);  
  
        for i := 0 to comp_count-1 do  
            for j := 0 to flows_count-1 do  
                Result[j, i] := A[i] + B[i] * flows_param[j, 1]  
            end;  
        end;  
    end;
```



```
function get_flow_Cp(comp_count: integer; flows_count: integer;  
                    flows_composition: TArrOfArrOfDouble;  
                    comp_Cp: TArrOfArrOfDouble): TArrOfDouble;  
  
var  
    i, j: integer;  
    s: double;  
  
begin  
    SetLength(Result, flows_count);  
  
    for j := 0 to flows_count-1 do  
        begin  
            s := 0;  
            for i := 0 to comp_count-1 do  
                s := s + flows_composition[j, i] * comp_Cp[j, i];  
            Result[j] := s;  
        end;  
  
end;
```



```
procedure mixer(comp_count: integer; flows_count: integer;  
                flows_param: TArrOfArrOfDouble;  
                flows_composition: TArrOfArrOfDouble;  
                var flow_rate, flow_Cp, flow_Temp: double;  
                var flow_composition: TArrOfDouble);  
  
var  
    i, j: integer;  
    comp_Cp: TArrOfArrOfDouble;  
    flows_Cp: TArrOfDouble;  
  
begin  
    SetLength(comp_Cp, flows_count);  
  
    for i := 0 to flows_count-1 do  
        SetLength(comp_Cp[i], comp_count);  
  
    SetLength(flows_Cp, flows_count);  
  
    flow_rate := 0;  
    flow_Cp := 0;  
    flow_Temp := 0;
```




```
// Определяем теплоемкость каждого компонента
comp_Cp := get_comp_Cp(comp_count, flows_count, flows_param);

// Определяем теплоемкость потоков смесителя
flows_Cp := get_flow_Cp(comp_count, flows_count, flows_composition, comp_Cp);

(* Определение расхода смешиваемого потока (G4) *)
for i:= 0 to flows_count-1 do
    flow_rate:= flow_rate + flows_param[i, 0];

(* Определение состава смешиваемого потока (G4) *)
for i:= 0 to comp_count-1 do
    for j:= 0 to flows_count-1 do
        flow_composition[i]:= flow_composition[i] + flows_param[j, 0]
            * flows_composition[j, i] / flow_rate;

(* Определение теплоемкости смешиваемого потока (G4) *)
for i:= 0 to flows_count-1 do
    flow_Cp:= flow_Cp + flows_param[i, 0] / flow_rate * flows_Cp[i];

(* Определение температуры смешиваемого потока (G4) *)
for i:= 0 to flows_count-1 do
    flow_Temp:= flow_Temp + flows_param[i, 0] * flows_Cp[i] * (flows_param[i, 1]
        + 273.15) / (flow_Cp * flow_rate);

end;
```



```
procedure heater(comp_count: integer; flows_count: integer;  
    heater_flow_rate: double; heater_flow_Temp: double;  
    flow_rate: double; flow_Cp: double;  
    var flow_Temp: double);  
  
const  
    d_vnutr = 0.05; // Внутр. диаметр труб, м  
    d_vnesh = 0.057; // Внеш. диаметр труб, м  
    kol_trub = 3280; //Количество труб  
    S_sech = 5.85; //Площ. сечения теплообмена, м2  
    r_term = 0.127; //Термич. сопротивление стенки, м2/Вт  
    S = 3860; //Площадь теплообмена, м2  
    Cp_hot = 0.920; //Теплоемкость гор. теплоносителя, Дж/(кг*К)  
  
var  
    W_cold: double; // Водяной эквивалент холодного потока  
    W_hot: double; // Водяной эквивалент горячего потока  
    N, M, a1, a2, Kt: double;
```



begin

W_cold:= flow_rate * flow_Cp;

W_hot:= heater_flow_rate * Cp_hot;

N:= W_hot / W_cold;

a1:= 1 / d_vnutr * exp(0.8 * ln(heater_flow_Temp / (kol_trub * d_vnutr)))
* (3.43 + 2.71e-3 * heater_flow_Temp);

a2:= 1 / d_vnesh * exp(0.6 * ln(flow_rate * d_vnesh / S_sech))
* (3.483 + 7.27e-3 * (flow_Temp - 273.15));

Kt:= 1 / (1 / a1 + r_term + 1 / a2);

M:= Kt * S / W_hot;

flow_Temp := (flow_Temp - 273.15) + (heater_flow_Temp - (flow_Temp - 273.15))
* N * (exp(M * (1 - N)) - 1) / (exp(M * (1 - N)) - N) + 273.15

end;



```
procedure splitter(split_ratio: double; flow_rate: double;  
                  var flow_rate1, flow_rate2: double);  
begin  
  if split_ratio <= 1 then  
    begin  
      flow_rate1 := split_ratio * flow_rate;  
      flow_rate2 := (1 - split_ratio) * flow_rate;  
    end  
  end;  
end;
```



```
program hts;

uses
  UHTS;

const
  comp_count = 2;
  flows_count = 3;

var
  flows_composition: TArrOfArrOfDouble; // Состав потоков, подаваемых на смешение
  flows_param: TArrOfArrOfDouble; // Параметры потоков
  heater_flow_rate: double; // Расход горячего потока теплообменника
  heater_flow_Temp: double; // Начальная температура горячего потока
                           теплообменника
  flow_rate: double; // Расход потока
  flow_Cp: double; // Теплоемкость потока
  flow_Temp: double; // Конечная температура холодного потока
  flow_composition: TArrOfDouble; // Состав потока после смешения
  flow_rate1: double; // Расходы потока после делителя
  flow_rate2: double; // Расходы потока после делителя
  f1, f2: text;
  i, j: integer;
```



begin

```
SetLength(flows_param, flows_count);
```

```
for i := 0 to flows_count-1 do
```

```
    SetLength(flows_param[i], comp_count);
```

```
SetLength(flows_composition, flows_count);
```

```
for i := 0 to flows_count-1 do
```

```
    SetLength(flows_composition[i], comp_count);
```

```
SetLength(flow_composition, comp_count);
```

```
assign(f1, 'flows_param.txt');
```

```
reset(f1);
```

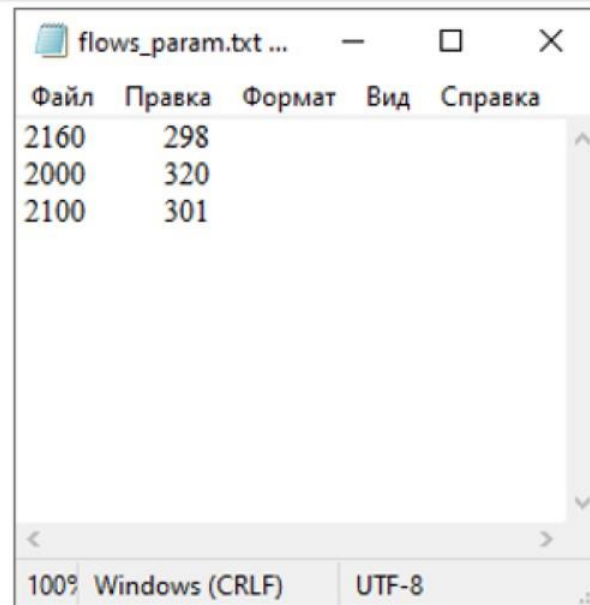
```
assign(f2, 'flows_composition.txt');
```

```
reset(f2);
```



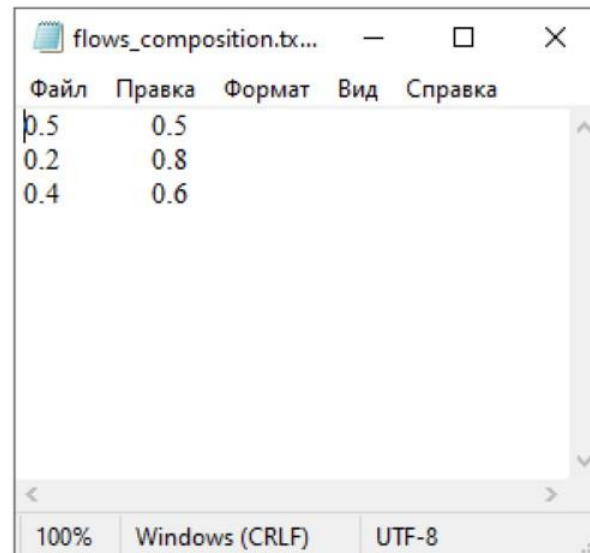
```
//Считывание параметров потоков
for i:= 0 to flows_count-1 do
  begin
    for j:= 0 to comp_count-1 do
      read(f1, flows_param[i, j]);
      readln(f1)
    end;

//Считывание составов потоков
for i:= 0 to flows_count-1 do
  begin
    for j:= 0 to comp_count-1 do
      read(f2, flows_composition[i, j]);
      readln(f2)
    end;
```



Файл	Правка	Формат	Вид	Справка
2160	298			
2000	320			
2100	301			

100% Windows (CRLF) UTF-8



Файл	Правка	Формат	Вид	Справка
0.5	0.5			
0.2	0.8			
0.4	0.6			

100% Windows (CRLF) UTF-8



```
writeln('=====Смеситель=====');
  mixer(comp_count, flows_count, flows_param, flows_composition, flow_rate,
        flow_Cp, flow_Temp, flow_composition);
  writeln('Расход потока', flow_rate:10, ' кг / ч');
  for i:= 0 to comp_count-1 do
    writeln('Доля компонента ', i+1, flow_composition[i]:8:4);
  writeln('Теплоемкость потока', flow_Cp:10:4, ' Дж / (кг * К)');
  writeln('Температура потока', flow_Temp:13:4, ' К');

  writeln('=====Теплообменник=====');
  heater(comp_count, flows_count, 6200, 360, flow_rate, flow_Cp, flow_Temp);
  writeln('Т нагретого потока', flow_Temp:10:4, ' К');
  writeln('F нагретого потока', flow_rate:6, ' кг/ч');
  for i:= 0 to comp_count-1 do
    writeln('Доля компонента ', i, flow_composition[i]:9:4);

  writeln('=====Делитель=====');
  Splitter(0.8, flow_rate, flow_rate1, flow_rate2);
  writeln('F потока ', 1, flow_rate1:9);
  writeln('F потока ', 2, flow_rate2:9);

  close(f1);
  close(f2)
end.
```