

Приведенная выше простая программа имеет ограниченные возможности. Более интересный программный код (в свободном формате размещения операторов) приведен в следующем примере.

Пример. Для математической модели предыдущего примера определить зависимость температуры трех объектов, которые характеризуются коэффициентами $K_1 = 0,2 \text{ с}^{-1}$, $K_2 = 0,6 \text{ с}^{-1}$, $K_3 = 0,8 \text{ с}^{-1}$, соответственно, в зависимости от времени t . Принять начальную температуру объектов $T_0 = 16^\circ \text{C}$, температуру среды $T_{CP} = 32,2 - 0,12 \cdot t_2$, время наблюдения $t_{\max} = 12 \text{ с}$. По методу Эйлера получить таблицу значений для 6000 узлов по времени и 6 узлов по времени. Предусмотреть поиск максимальных и минимальных значений табличных функций, которые находят. Построить графики зависимостей искомых величин (температуры среды и температуры объекта при различных значениях K) со дня.

Исходный код программы:

```
PROGRAM EulerT
```

```
! Массивы для результатов работы функций поиска мин. и макс.
```

```
DIMENSION mLocMaxT1 (1), mLocMaxT2 (1), mLocMaxT3 (1), &  
mLocMinT1 (1), mLocMinT2 (1), mLocMinT3 (1), mLocMaxTcp (1), mLocMinTcp  
(1)
```

```
Allocatable :: T1 (:), T2 (:), T3 (:), Tcp (:)! Динамические массивы
```

```
OPEN (1, file = 'EulerTin.txt')! Файл исходных данных
```

```
READ (1, *)! Первая строка не содержит данных
```

```
READ (1, *) TauMax, mMax, T0, aK1, aK2, aK3! Выходные данные
```

```
CLOSE (1)! Файл не нужен - закрываем
```

```
ALLOCATE (T1 (mMax), T2 (mMax), T3 (mMax), Tcp (mMax))! Предоставим
```

```
Dtau = TauMax / (mMax-1)! Шаг
```

```
CALL SolveTcp (mMax, Dtau, Tcp)! Рассчитываем TCP ()
```

```
CALL Euler (mMax, Dtau, T0, aK1, T1)! Рассчитываем T1 ()
```

```
CALL Euler (mMax, Dtau, T0, aK2, T2)! Рассчитываем T2 ()
```

```
CALL Euler (mMax, Dtau, T0, aK3, T3)! Рассчитываем T3 ()
```

```
T1max = MAXVAL (T1)! Макс. T1
```

```
T2max = MAXVAL (T2)! Макс. T2
```

```
T3max = MAXVAL (T3)! Макс. T3
```

```
T1min = MINVAL (T1)! Мин. T1
```

```
T2min = MINVAL (T2)! Мин. T2
```

```
T3min = MINVAL (T3)! Мин. T3
```

```
TcpMax = MAXVAL (Tcp)! Макс. TSP
```

```
TcpMin = MINVAL (Tcp)! Мин. TSP
```

```
  mLocMaxT1 = MAXLOC (T1)! Индекс Макс. T1
```

```
  mLocMaxT2 = MAXLOC (T2)! Индекс Макс. T2
```

```
mLocMaxT3 = MAXLOC (T3)! Индекс Макс. T3
```

```
mLocMinT1 = MINLOC (T1)! Индекс Мин. T1
```

```
mLocMinT2 = MINLOC (T2)! Индекс Мин. T2
```

```
mLocMinT3 = MINLOC (T3)! Индекс Мин. T3
```

```

mLocMaxTcp = MAXLOC (Ttcp)! Индекс Макс. TCP
  mLocMinTtcp = MINLOC (Ttcp)! Индекс Мин. TCP
WRITE (*, *) '***** Input data: "! На экран
WRITE (*, *) 'TauMax =', TauMax, "mMax = ', mMax," T0 =', T0
WRITE (*, *) 'aK1 =', aK1, "aK2 = ', aK2," aK3 =', aK3
WRITE (*, *) '***** Output data: "
OPEN (2, file = 'EulreTout.txt')! Файл вывода
WRITE (2, *) '***** Input data: "! Файл
WRITE (2, *) 'aK1 =', aK1, "aK2 = ', aK2," aK3 =', aK3
WRITE (2, *) 'TauMax =', TauMax, "mMax = ', mMax," T0 =', T0
WRITE (2, *) '***** Output data: "
WRITE (2, *) 'T1max =', T1max, "mLocMaxT1 = ', mLocMaxT1
WRITE (2, *) 'T2max =', T2max, "mLocMaxT2 = ', mLocMaxT2
WRITE (2, *) 'T3max =', T3max, "mLocMaxT3 = ', mLocMaxT3
WRITE (2, *) 'T1min =', T1min, "mLocMinT1 = ', mLocMinT1
WRITE (2, *) 'T2min =', T2min, "mLocMinT2 = ', mLocMinT2
WRITE (2, *) 'T3min =', T3min, "mLocMinT3 = ', mLocMinT3
WRITE (2, *) 'TtcpMax =', TtcpMax, "mLocMaxTtcp = ', mLocMaxTtcp
WRITE (2, *) 'TtcpMin =', TtcpMin, "mLocMinTtcp = ', mLocMinTtcp
WRITE (2, *) ' _____ "
Tau = 0.
WRITE (*, *) 'm Tau Ttcp T1 T2 T3!' шапка
WRITE (2, *) 'm Tau Ttcp T1 T2 T3!' шапка
DO m = 1, mMax
WRITE (*, 10) m, Tau, Ttcp (m), T1 (m), T2 (m), T3 (m)! На экран
WRITE (2,10) m, Tau, Ttcp (m), T1 (m), T2 (m), T3 (m)! К файлу
Tau = Tau + Dtau
ENDDO
10 FORMAT (2x, i5, F12.7, 4F12.2)
CLOSE (2)! Файл не нужен - закрываем
DEALLOCATE (T1, T2, T3, Ttcp)! Освобождаем память
END PROGRAM EulerT
! _____
SUBROUTINE SolveTtcp (iMax, Dtau, Ttcp)! Рассчитываем TCP ()
DIMENSION Ttcp (iMax)! Массив TCP
Tau = 0.
DO i = 1, iMax
Ttcp (i) = FTtcp (Tau)! Заполняем массив TCP по функции FTtcp (Tau)
Tau = Tau + Dtau
ENDDO
END SUBROUTINE SolveTtcp
! _____
FUNCTION FTtcp (Tau)! Функция температуры среды (см. Задание)
FTtcp = 32.2-.12 * Tau * Tau
END FUNCTION FTtcp
! _____

```

```
SUBROUTINE Euler (iMax, Dx, U0, aK, U)! Метод Эйлера
DIMENSION U (iMax)
U (1) = U0! Начальные условия
x = 0.
DO i = 2, iMax
x = x + Dx! Координата
U (i) = U (i-1) + Dx * F (aK, U (i-1), x)! Формула Эйлера
ENDDO
END SUBROUTINE Euler
!
```

```
FUNCTION F (aK, T, Tau)! Правая часть диф. уравнения
F = -aK * (T-FTcp (Tau))
END FUNCTION F
```

Файл исходных данных EulerTin.txt для случая расчета на 6-ти узлах содержит следующие данные.

TauMax mMax T0 aK1 aK2 aK3
12. 6 16. .2 .6 .8

Программа формирует файл результатов расчета EulerTout.txt следующего содержания.

***** Input data:

aK1 = 2.000000E-01 aK2 = 6.000000E-01 aK3 = 8.000000E-01

TauMax = 12.000000 mMax = 6 T0 = 16.000000

***** Output data:

T1max = 26.319890 mLocMaxT1 = 3

T2max = 38.332680 mLocMaxT2 = 2

T3max = 45.776900 mLocMaxT3 = 2

T1min = 16.000000 mLocMinT1 = 1

T2min = 13.158650 mLocMinT2 = 6

T3min = 6.889544 mLocMinT3 = 5

TcpMax = 32.200000 mLocMaxTcp = 1

TcpMin = 14.920000 mLocMinTcp = 6

m	Tau	Tcp	T1	T2	T3
1	.0000000	32.20	16.00	16.00	16.00
2	2.4000000	31.51	23.44	38.33	45.78
3	4.8000000	29.44	26.32	25.52	14.40
4	7.2000000	25.98	26.16	26.18	36.63
5	9.6000000	21.14	23.75	18.92	6.89
6	12.0000000	14.92	19.51	13.16	22.31