

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
Государственное образовательное учреждение
высшего профессионального образования
"ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ"

А.А. Черный

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ
С ПРИМЕНЕНИЕМ ГРАФИЧЕСКИХ ПОСТРОЕНИЙ В EXCEL
Учебное пособие

Пенза 2010

Черный А.А.

Математическое моделирование с применением графических построений в EXCEL: учебное пособие/ А.А. Черный. – Пенза: Пензенский государственный университет, 2010. - 91 с.

Изложены рекомендации по практическому применению математического моделирования. Приводятся алгоритм математического моделирования применительно к программированию на языках Бейсик и Турбо Паскаль, компьютерные программы, результаты расчетов, графические построения в Excel, анализ моделирования, контрольные вопросы.

Учебное пособие разработано применительно к учебному процессу по кафедре «Сварочное, литейное производство и материаловедение». Оно может быть использовано студентами при изучении курсов «Математическое моделирование в литейном производстве», «Вычислительная техника в инженерных расчетах», а также при выполнении курсовых и научно-исследовательских работ. При разработке учебного пособия использованы компьютерные программы NV6 и W4 (три модуля trp3), являющиеся интеллектуальной собственностью Черного А.А.

Рецензенты:

научный совет Пензенского научного центра;
А.П. Белоусов, главный металлург ОАО «Пензадизельмаш».

В В Е Д Е Н И Е

В литейном производстве используется много способов, устройств, веществ, позволяющих получать разнообразные отливки с требуемыми свойствами. Однако, несмотря на длительное развитие и совершенствование процессов литья и достигнутые успехи в литейном производстве существуют проблемы, которые необходимо решать: надо многие процессы оптимизировать, сделать дешевле, экологически чистыми, безопасными, привлекательными для молодых специалистов, более механизированными и автоматизированными. Необходимы в литейном производстве новые усовершенствования и изобретения.

Но процессы литейного производства зачастую сложны, на них могут влиять много факторов. Поэтому для совершенствования литейного производства рационально применять моделирование.

Выполнены оригинальные разработки математического моделирования при планировании экспериментов на двух и более уровнях факторов. Основы математического моделирования применительно к литейному производству изложены в работах автора [1, 2, 4 – 9].

В данной работе приводятся усовершенствованные программы математического моделирования и расчетов по математическим моделям NV6 на языке Бейсик и W4 (три модуля trp3) на языке Турбо Паскаль. Программы проверены при использовании экспериментальных и практических данных исследованных процессов литейного производства. Они носят универсальный характер.

Применение математического моделирования становится понятным, если приведены примеры, изложены методические указания.

Для совершенствования процессов литейного производства проводят эксперименты, выявляют математические модели, выполняют расчеты по математическим моделям, анализируют результаты расчетов и на основе анализов полученных данных проводят оптимизацию, разрабатывают изобретения. Результаты расчетов рационально представлять в виде графиков, построенных с применением Excel по цифровым данным системного анализа.

Так как при выполнении математического моделирования и расчетов на ЭВМ необходимо устно объяснять методику работы, то потребовалось изложить на примере использования программы NV6 практические особенности математического моделирования и анализа результатов расчетов.

АЛГОРИТМ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ И РАСЧЕТОВ ПО МАТЕМАТИЧЕСКИМ МОДЕЛЯМ

Применительно к использованию ЭВМ разработан следующий алгоритм математического моделирования:

1. Начало выполнения программы, ввод количества опытов по плану, величин факторов на принятых уровнях и показателей степени в уравнении регрессии.
2. Расчет коэффициентов ортогонализации.
3. Ввод величин показателей процесса.
4. Расчет коэффициентов регрессии до их анализа.
5. Ввод количества опытов на среднем уровне факторов.
6. Расчет показателей до анализа коэффициентов регрессии.
7. Выявление дисперсии опытов, расчетных величин t-критерия для каждого коэффициента регрессии.
8. Ввод табличного t-критерия.
9. Выявление статистически значимых коэффициентов регрессии.
10. Ввод табличного F-критерия.
11. Расчет показателей после анализа коэффициентов регрессии.
12. Выявление расчетной величины F-критерия и адекватности модели, системное представление для анализов результатов по математической модели, вывод математической модели.
13. Выполнение расчетов по модели и проверка точности модели.
14. Вычисления показателей по математической модели с использованием циклов и построение графиков.
15. Конец выполнения программы.

Для персональных компьютеров программирование выполнено на языках Бейсик и Турбо Паскаль.

Разработаны программы математического моделирования с добавлением системного представления для анализов результатов расчетов по математической модели. Ниже представлена компьютерная программа NV6 на языке Бейсик для случаев планирования 4^2 ($X = 16$).

ПЛАН ПРОВЕДЕНИЯ ЭКСПЕРИМЕНТОВ

Таблица 1

План 4^2 ($X = 16$)

Номер опыта	Факторы		Показатель Y(J)
	F(J)	H(J)	
1	A1	A2	Y(1)
2	B1	A2	Y(2)
3	A1	B2	Y(3)
4	B1	B2	Y(4)
5	A1	C2	Y(5)
6	B1	C2	Y(6)
7	A1	D2	Y(7)
8	B1	D2	Y(8)
9	C1	A2	Y(9)
10	C1	C2	Y(10)
11	C1	D2	Y(11)
12	C1	B2	Y(12)
13	D1	A2	Y(13)
14	D1	C2	Y(14)
15	D1	D2	Y(15)
16	D1	B2	Y(16)

ТАБЛИЦА СИСТЕМНОГО ПРЕДСТАВЛЕНИЯ РЕЗУЛЬТАТОВ
РАСЧЕТОВ

Таблица 2

Системное представление для анализов результатов расчетов
по математическим моделям при планировании 4^2

Номера зави- симостей и результатов	Факторы		Показатели		
			действительные величины	относительные величины	
1	2	3	4	5	
1	1.1	A2	A1	Z(1)	Z1(1)
	1.2	A2	C1	Z(9)	Z1(9)
	1.3	A2	D1	Z(13)	Z1(13)
	1.4	A2	B1	Z(2)	Z1(2)
2	2.1	C2	A1	Z(5)	Z1(5)
	2.2	C2	C1	Z(10)	Z1(10)
	2.3	C2	D1	Z(14)	Z1(14)
	2.4	C2	B1	Z(6)	Z1(6)
3	3.1	D2	A1	Z(7)	Z1(7)
	3.2	D2	C1	Z(11)	Z1(11)
	3.3	D2	D1	Z(15)	Z1(15)
	3.4	D2	B1	Z(8)	Z1(8)
4	4.1	B2	A1	Z(3)	Z1(3)
	4.2	B2	C1	Z(12)	Z1(12)
	4.3	B2	D1	Z(16)	Z1(16)
	4.4	B2	B1	Z(4)	Z1(4)
5	5.1	A1	A2	Z(1)	Z1(1)
	5.2	A1	C2	Z(5)	Z1(5)
	5.3	A1	D2	Z(7)	Z1(7)
	5.4	A1	B2	Z(3)	Z1(3)
6	6.1	C1	A2	Z(9)	Z1(9)
	6.2	C1	C2	Z(10)	Z1(10)
	6.3	C1	D2	Z(11)	Z1(11)
	6.4	C1	B2	Z(12)	Z1(12)
7	7.1	D1	A2	Z(13)	Z1(13)
	7.2	D1	C2	Z(14)	Z1(14)
	7.3	D1	D2	Z(15)	Z1(15)
	7.4	D1	B2	Z(16)	Z1(16)
8	8.1	B1	A2	Z(2)	Z1(2)
	8.2	B1	C2	Z(6)	Z1(6)
	8.3	B1	D2	Z(8)	Z1(8)
	8.4	B1	B2	Z(4)	Z1(4)

ПРОГРАММА НА ЯЗЫКЕ БЕЙСИК NV6

```
5 PRINT "ПРОГРАММА NV6,РАЗРАБОТКА А.А.ЧЕРНОГО"
6 CLS
7 PRINT "РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ПРОГРАММЫ NV6"
8 PRINT "ЗАНОСЯТСЯ В ФАЙЛ,ИМЯ КОТОРОГО НАДО ВВЕСТИ,"
9 PRINT "НАПРИМЕР, ВВЕСТИ ИМЯ ФАЙЛА NV61"
10 INPUT "ВВОД ИМЕНИ ФАЙЛА ",FA$
14 OPEN "O", #1, FA$
17 PRINT "РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ПРОГРАММЫ NV6 ЗАНОСЯТСЯ В ФАЙЛ "; FA$
40 PRINT " РАЗРАБОТКА ДЛЯ X=16"
41 PRINT #1, "РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ПРОГРАММЫ VN6, РАЗРАБОТАННОЙ А.А. ЧЕРНЫМ"
42 DIM F(20), H(20), L(20), Y(20), I(20), K(20), P(20)
44 DIM Q(20), U(20), V(20), O(20), B(20), Z(20), G(20), T(20)
46 DIM K6(20), K7(20), K8(20), J7(20), J8(20), J9(20)
47 DIM Z1(50)
51 PRINT "ВВОД X-КОЛИЧЕСТВО ОПЫТОВ ПО ПЛАНУ "
52 PRINT #1, "КОЛИЧЕСТВО ОПЫТОВ ПО ПЛАНУ "
60 INPUT X: PRINT #1, "X="; X
61 PRINT #1, "ВЕЛИЧИНЫ ФАКТОРОВ И ПОКАЗАТЕЛЕЙ СТЕПЕНИ"
62 PRINT "ВВОД ВЕЛИЧИН ФАКТОРОВ И ПОКАЗАТЕЛЕЙ СТЕПЕНИ"
200 PRINT "ВВОД A1,C1,D1,B1,J1,O1,P1"
210 INPUT A1, C1, D1, B1, J1, O1, P1
213 PRINT #1, "A1="; A1; " C1="; C1; " D1="; D1
215 PRINT #1, " B1="; B1; " J1="; J1; " O1="; O1; " P1="; P1
220 A = A1: B = B1: C = C1: D = D1: N = J1: R = O1: S = P1: GOSUB 3710
230 V1 = V0: U1 = U0: Q1 = Q0: I1 = I0: M1 = M0: F1 = F0
240 PRINT #1, "V1="; V1; " U1="; U1; " Q1="; Q1
243 PRINT #1, " I1="; I1; " M1="; M1; " F1="; F1
245 IF X = 16 GOTO 400
400 PRINT "ВВОД A2,C2,D2,B2,J2,O2,P2"
410 INPUT A2, C2, D2, B2, J2, O2, P2: PRINT #1, "A2="; A2
413 PRINT #1, "C2="; C2; " D2="; D2; " B2="; B2; " J2="; J2
415 PRINT #1, "O2="; O2; "P2="; P2
420 A = A2: B = B2: C = C2: D = D2: N = J2: R = O2: S = P2: GOSUB 3710
430 V2 = V0: U2 = U0: Q2 = Q0: I2 = I0: M2 = M0: F2 = F0
440 PRINT #1, "V2="; V2; " U2="; U2; " Q2="; Q2; " I2="; I2
443 PRINT #1, "M2="; M2; " F2="; F2
450 IF X = 16 GOTO 770
770 F(1) = A1: H(1) = A2: F(2) = B1: H(2) = A2: F(3) = A1: H(3) = B2
780 F(4) = B1: H(4) = B2: F(5) = A1: H(5) = C2: F(6) = B1: H(6) = C2
790 F(7) = A1: H(7) = D2: F(8) = B1: H(8) = D2: F(9) = C1: H(9) = A2
800 F(10) = C1: H(10) = C2: F(11) = C1: H(11) = D2: F(12) = C1
805 H(12) = B2
810 F(13) = D1: H(13) = A2: F(14) = D1: H(14) = C2: F(15) = D1
815 H(15) = D2
820 F(16) = D1: H(16) = B2: GOTO 1130
1130 PRINT "IF I0=6 GOTO 40-НАЧАЛО"
1135 PRINT "IF I0=7 GOTO 1160-ПРОДОЛЖЕНИЕ"
1140 INPUT I0: IF I0 = 6 GOTO 40
1150 IF I0 = 7 GOTO 1160
1160 PRINT "ВВОД ВЕЛИЧИН ПОКАЗАТЕЛЕЙ ПО ПЛАНУ Y(J) "
1161 PRINT #1, "ВЕЛИЧИНЫ ПОКАЗАТЕЛЕЙ В СООТВЕТСТВИИ С ПЛАНОМ Y(J)"
1165 FOR J = 1 TO X
1166 PRINT "Y("; J; ")": INPUT Y(J)
1170 PRINT #1, "Y("; J; ")="; Y(J): NEXT J
1180 PRINT "IF I0=1 GOTO 1160-ПОВТОРЕНИЕ ВВОДА ПОКАЗАТЕЛЕЙ"
1185 PRINT "IF I0=2 GOTO 1270-ПРОДОЛЖЕНИЕ"
1190 PRINT "ВВОД I0": INPUT I0: IF I0 = 1 GOTO 1160
1200 IF I0 = 2 GOTO 1270
1270 IF X = 16 GOTO 1370
```

```

1370 GOSUB 4210: GOSUB 4250: GOTO 1410
1410 S = 0: O(1) = 0: FOR J = 1 TO X: S = S + Y(J): O(1) = O(1) + 1: NEXT J
1420 B(1) = S / O(1): S = 0: O(2) = 0: FOR J = 1 TO X: S = S + I(J) * Y(J)
1430 O(2) = O(2) + I(J) ^ 2: NEXT J: B(2) = S / O(2): S = 0: O(3) = 0
1440 FOR J = 1 TO X: S = S + K(J) * Y(J): O(3) = O(3) + K(J) ^ 2: NEXT J
1450 B(3) = S / O(3)
1480 S = 0: O(4) = 0: FOR J = 1 TO X: S = S + P(J) * Y(J)
1490 O(4) = O(4) + P(J) ^ 2: NEXT J: B(4) = S / O(4): S = 0: O(5) = 0
1500 FOR J = 1 TO X: S = S + I(J) * P(J) * Y(J): O(5) = O(5) + (I(J) * P(J)) ^ 2
1510 NEXT J: B(5) = S / O(5): S = 0: O(6) = 0: FOR J = 1 TO X
1520 S = S + Q(J) * Y(J): O(6) = O(6) + Q(J) ^ 2: NEXT J: B(6) = S / O(6)
1530 S = 0: O(7) = 0: FOR J = 1 TO X: S = S + I(J) * Q(J) * Y(J)
1540 O(7) = O(7) + (I(J) * Q(J)) ^ 2: NEXT J: B(7) = S / O(7): S = 0
1550 O(8) = 0: FOR J = 1 TO X: S = S + P(J) * K(J) * Y(J)
1560 O(8) = O(8) + (P(J) * K(J)) ^ 2: NEXT J: B(8) = S / O(8): S = 0: O(9) = 0
1570 FOR J = 1 TO X: S = S + K(J) * Q(J) * Y(J): O(9) = O(9) + (K(J) * Q(J)) ^ 2
1580 NEXT J: B(9) = S / O(9)
1600 S = 0: O(10) = 0: FOR J = 1 TO X: S = S + U(J) * Y(J): O(10) = O(10) + U(J) ^ 2
1610 NEXT J: B(10) = S / O(10): S = 0: O(11) = 0: FOR J = 1 TO X
1620 S = S + I(J) * U(J) * Y(J): O(11) = O(11) + (I(J) * U(J)) ^ 2: NEXT J
1630 B(11) = S / O(11): S = 0: O(12) = 0: FOR J = 1 TO X: S = S + K(J) * U(J) * Y(J)
1640 O(12) = O(12) + (K(J) * U(J)) ^ 2: NEXT J: B(12) = S / O(12)
1660 IF X = 16 GOTO 1930
1930 S = 0: O(13) = 0: FOR J = 1 TO X: S = S + L(J) * Y(J): O(13) = O(13) + L(J) ^ 2
1940 NEXT J: B(13) = S / O(13): S = 0: O(14) = 0: FOR J = 1 TO X
1950 S = S + P(J) * L(J) * Y(J): O(14) = O(14) + (P(J) * L(J)) ^ 2: NEXT J
1960 B(14) = S / O(14): S = 0: O(15) = 0: FOR J = 1 TO X: S = S + Q(J) * L(J) * Y(J)
1970 O(15) = O(15) + (Q(J) * L(J)) ^ 2: NEXT J: B(15) = S / O(15): S = 0
1980 O(16) = 0: FOR J = 1 TO X: S = S + L(J) * U(J) * Y(J)
1990 O(16) = O(16) + (L(J) * U(J)) ^ 2: NEXT J: B(16) = S / O(16): GOTO 2390
2390 PRINT #1, "B(J) ДО АНАЛИЗА": FOR J = 1 TO X
2395 PRINT #1, "B("; J; ")="; B(J)
2397 NEXT J: PRINT
2400 PRINT "ВВОД N0-КОЛИЧЕСТВО ОПЫТОВ НА СРЕДНЕМ УРОВНЕ ФАКТОРОВ"
2407 INPUT N0
2408 PRINT #1, "КОЛИЧЕСТВО ОПЫТОВ НА СРЕДНЕМ УРОВНЕ ФАКТОРОВ"
2410 PRINT #1, "N0="; N0
2470 IF X = 16 GOTO 2570
2570 GOSUB 4580: GOTO 2610
2610 PRINT #1, "РАСЧЕТНЫЕ ВЕЛИЧИНЫ ПОКАЗАТЕЛЕЙ Z(J) ДО АНАЛИЗА B(J)"
2620 FOR J = 1 TO X: PRINT #1, "Z("; J; ")="; Z(J): NEXT J
2630 PRINT "ВВОД F8=N0-1": INPUT F8
2633 PRINT #1, "F8=N0-1="; F8
2635 PRINT "F8="; F8
2640 PRINT #1, "ПРОВЕРКА ПО РАЗНОСТИ Y(J)-Z(J)"
2641 PRINT #1, "B ПРОЦЕНТАХ (Y(J)-Z(J)) * (100/Y(J))"
2650 FOR J = 1 TO X: PRINT #1, "Y("; J; ") - Z("; J; ")="; Y(J) - Z(J)
2651 PRINT #1, "(Y("; J; ") - Z("; J; ")) * (100 / Y("; J; ")) = "; (Y(J) - Z(J)) * (100 / Y(J))
2655 NEXT J
2660 PRINT "IF I0=3 GOTO 2720-ВВОД РЕЗУЛЬТАТОВ ОПЫТОВ "
2663 PRINT "      НА СРЕДНЕМ УРОВНЕ ФАКТОРОВ"
2666 PRINT "IF I0=4 GOTO 2770-ВВОД ДИСПЕРСИИ ОПЫТОВ"
2670 PRINT "IF I0=5 GOTO 3240-ПРОВЕРКА ТОЧНОСТИ И "
2672 PRINT "      РАСЧЕТЫ ПО МОДЕЛИ"
2773 PRINT "IF I0=6 GOTO 40-НАЧАЛО"
2677 PRINT "IF I0=20 GOTO 6830-КОНЕЦ"
2678 PRINT "IF I0=25 GOTO 4880-МАТЕМАТИЧЕСКАЯ МОДЕЛЬ"
2679 PRINT "IF I0=27 GOTO 7000-ВЫЧИСЛЕНИЯ ПОКАЗАТЕЛЕЙ Z(K5)"
2681 PRINT "      С ИСПОЛЬЗОВАНИЕМ ЦИКЛОВ И "

2684 PRINT "      ПОСТРОЕНИЕ ГРАФИКОВ"
2689 PRINT "ВВОД I0": INPUT I0: IF I0 = 3 GOTO 2720
2690 IF I0 = 4 GOTO 2770

```



```

2700 IF I0 = 5 GOTO 3240
2710 IF I0 = 6 GOTO 40
2715 IF I0 = 20 GOTO 6830
2717 IF I0 = 25 GOTO 4880
2718 IF I0 = 27 GOTO 7000
2720 PRINT "ВВОД G(J)-РЕЗУЛЬТАТЫ ОПЫТОВ ";
2721 PRINT "НА СРЕДНЕМ УРОВНЕ ФАКТОРОВ "
2722 PRINT #1, "РЕЗУЛЬТАТЫ ОПЫТОВ ";
2723 PRINT #1, "НА СРЕДНЕМ УРОВНЕ ФАКТОРОВ "
2724 FOR J = 1 TO N0: PRINT "G("; J; ")": INPUT G(J)
2730 PRINT #1, "G("; J; ")="; G(J): NEXT J: S = 0: FOR J = 1 TO N0: S = S + G(J)
2740 NEXT J: S0 = S / N0: PRINT "S0="; S0: S = 0: FOR J = 1 TO N0
2750 S = S + (G(J) - S0) ^ 2: NEXT J: U9 = S / F8
2751 PRINT #1, "ДИСПЕРСИЯ ОПЫТОВ U9="; U9
2760 GOTO 2780
2770 PRINT "ВВОД U9-ДИСПЕРСИЯ ОПЫТОВ": INPUT U9
2771 PRINT #1, "ДИСПЕРСИЯ ОПЫТОВ U9="; U9
2780 PRINT #1, "РАСЧЕТНЫЕ ВЕЛИЧИНЫ T(J)": FOR J = 1 TO X
2790 T(J) = ABS(B(J) / SQR(U9 / O(J))): PRINT #1, "T("; J; ")="; T(J): NEXT J
2800 PRINT "          ДЛЯ УРОВНЯ ЗНАЧИМОСТИ 5%"
2801 PRINT "          ПРИ F8  2 ; 3 ; 4 ; 5 ; 6"
2802 PRINT "СООТВЕТСТВЕННО T0  4.303 ;3.182 ;2.776 ;2.571 ;2.447"
2803 PRINT "F8=N0-1="; N0; "-1="; F8
2804 PRINT "ВВОД T0-ТАБЛИЧНЫЙ T-КРИТЕРИЙ"
2805 INPUT T0
2806 PRINT #1, "ТАБЛИЧНЫЙ T-КРИТЕРИЙ T0="; T0
2810 PRINT #1, "B(J) ПОСЛЕ АНАЛИЗА": FOR J = 1 TO X
2820 IF T(J) < T0 GOTO 2840
2830 IF T(J) >= T0 GOTO 2850
2840 B(J) = 0
2850 PRINT #1, "B("; J; ")="; B(J): NEXT J
2860 K9 = 0: FOR J = 1 TO X: IF B(J) = 0 GOTO 2871

2870 K9 = K9 + 1
2871 NEXT J
2872 PRINT #1, "КОЛИЧЕСТВО СТАТИСТИЧЕСКИ ЗНАЧИМЫХ"
2873 PRINT #1, "          КОЭФФИЦИЕНТОВ РЕГРЕССИИ K9="; K9
2881 PRINT #1, "F9=X-1": F9 = X - 1
2882 PRINT #1, "F9="; F9: CLS
2883 PRINT "! ! ЗНАЧЕНИЯ F-КРИТЕРИЯ F7 ДЛЯ 5%-ГО УРОВНЯ ЗНАЧИМОСТИ"
2884 PRINT "! !-----"
2885 PRINT "!F8!          F9          "
2886 PRINT "! !-----"
2887 PRINT "! ! 2 ! 3 ! 4 ! 8 ! 11 ! 14  "
2888 PRINT "!-----"
2889 PRINT "! 2! 19.0 ! 19.16 ! 19.25 ! 19.37 ! 19.4 ! 19.42 "
2890 PRINT "! 3!  9.55 !  9.28 !  9.12 !  8.84 !  8.76 !  8.71 "
2891 PRINT "! 4!  6.94 !  6.59 !  6.39 !  6.04 !  5.93 !  5.87 "
2892 PRINT "! 5!  5.79 !  5.41 !  5.19 !  4.82 !  4.7 !  4.64 "
2893 PRINT "! 6!  5.14 !  4.76 !  4.53 !  4.15 !  4.03 !  3.96 "
2894 PRINT "!===== "
2895 PRINT "! F8\F9 ! 15...16 ! 19...20 ! 24  ! 26...30 !"
2896 PRINT "!-----"
2897 PRINT "! 2 !    19.43 !  19.44 !  19.45 !  19.46 !"
2898 PRINT "! 3 !    8.69 !  8.66 !  8.64 !  8.62 !"
2899 PRINT "! 4 !    5.84 !  5.8 !  5.77 !  5.74 !"
2900 PRINT "! 5 !    4.6 !  4.56 !  4.53 !  4.5 !"
2901 PRINT "! 6 !    3.92 !  3.87 !  3.84 !  3.81 !"
2902 PRINT "!-----"
2907 PRINT "F8="; F8; "F9="; F9
2908 PRINT "ВВОД F7-ТАБЛИЧНЫЙ F-КРИТЕРИЙ"
2909 INPUT F7: PRINT #1, "ТАБЛИЧНЫЙ F-КРИТЕРИЙ F7="; F7
2970 IF X = 16 GOTO 3070

```

```

3070 GOSUB 4580: GOTO 3110
3110 PRINT #1, "РАСЧЕТНЫЕ ВЕЛИЧИНЫ ПОКАЗАТЕЛЯ";
3115 PRINT #1, " Z(J) ПОСЛЕ АНАЛИЗА В(J)"
3120 FOR J = 1 TO X: PRINT #1, "Z("; J; ")="; Z(J): NEXT J
3121 PRINT #1, "ПРОВЕРКА ПО РАЗНОСТИ Y(J)-Z(J)"
3122 PRINT #1, "В ПРОЦЕНТАХ (Y(J)-Z(J)) * (100/Y(J))"
3123 FOR J = 1 TO X: PRINT #1, "Y("; J; ") - Z("; J; ")="; Y(J) - Z(J)
3124 PRINT #1, "(Y("; J; ") - Z("; J; ")) * (100 / Y("; J; ")) = "; (Y(J) - Z(J)) * (100 / Y(J))
3125 NEXT J
3130 S = 0: FOR J = 1 TO X: S = S + (Z(J) - Y(J)) ^ 2: NEXT J
3140 F6 = S / (F9 * U9)
3145 PRINT #1, "РАСЧЕТНАЯ ВЕЛИЧИНА F-КРИТЕРИЯ F6="; F6
3150 IF F6 <= F7 GOTO 3152
3151 IF F6 > F7 GOTO 3153
3152 PRINT "АДЕКВАТНО": PRINT #1, "АДЕКВАТНО,ТАК КАК F6<=F7": GOTO 3154
3153 PRINT "НЕАДЕКВАТНО": PRINT #1, "НЕАДЕКВАТНО,ТАК КАК F6>F7": GOTO 3190
3154 PRINT #1, "СИСТЕМНОЕ ПРЕДСТАВЛЕНИЕ"
3155 PRINT #1, "ДЛЯ АНАЛИЗОВ РЕЗУЛЬТАТОВ РАСЧЁТОВ"
3156 PRINT #1, "ПО МАТЕМАТИЧЕСКОЙ МОДЕЛИ"
3157 S = 0: FOR J = 1 TO X: S = S + ABS(Z(J)): NEXT J
3158 Z1 = 0: FOR J = 1 TO X: Z1(J) = Z(J) / (S / X): NEXT J
3159 PRINT #1, "ОТНОСИТЕЛЬНАЯ ВЕЛИЧИНА ПОКАЗАТЕЛЯ"
3160 PRINT #1, "Z1(J)=Z(J)/(S/X),ГДЕ S- СУММА"
3161 PRINT #1, "АБСОЛЮТНЫХ ВЕЛИЧИН ПОКАЗАТЕЛЕЙ,S="; S
3162 PRINT #1, "S/X-СРЕДНЯЯ АРИФМЕТИЧЕСКАЯ ВЕЛИЧИН,S/X="; S / X
3167 IF X = 16 GOTO 3177
3177 GOSUB 11250: GOTO 3190
3190 PRINT "IF I0=7 GOTO 3240-ПРОВЕРКА ТОЧНОСТИ И "
3193 PRINT "          РАСЧЕТЫ ПО МОДЕЛИ"
3194 PRINT "IF I0=8 GOTO 40-НАЧАЛО"
3197 PRINT "IF I0=17 GOTO 4880-МАТЕМАТИЧЕСКАЯ МОДЕЛЬ"
3198 PRINT "IF I0=22 GOTO 7000-ВЫЧИСЛЕНИЯ ПОКАЗАТЕЛЕЙ Z(K5)"
3200 PRINT "          С ИСПОЛЬЗОВАНИЕМ ЦИКЛОВ И "
3203 PRINT "          ПОСТРОЕНИЕ ГРАФИКОВ"
3207 PRINT "IF I0=9 GOTO 6830-КОНЕЦ": PRINT "ВВОД I0": INPUT I0
3210 IF I0 = 7 GOTO 3240
3220 IF I0 = 8 GOTO 40
3227 IF I0 = 17 GOTO 4880
3228 IF I0 = 22 GOTO 7000
3230 IF I0 = 9 GOTO 6830
3240 PRINT "ПРОВЕРКА ТОЧНОСТИ И РАСЧЕТЫ ПО МОДЕЛИ"
3241 PRINT "F(S),H(S) -1, 2-й ФАКТОРЫ,"
3243 PRINT "ГДЕ S=X="; X; "-КОЛИЧЕСТВО ОПЫТОВ ПО ПЛАНУ"
3245 PRINT #1, "РАСЧЕТЫ ПО МОДЕЛИ"
3310 IF X = 16 GOTO 3420
3420 FOR S = 1 TO X: F(S) = 0: H(S) = 0: Z(S) = 0
3430 PRINT "ВВОД F("; S; "),H("; S; ")": INPUT F(S), H(S)
3432 PRINT #1, " ФАКТОРЫ F("; S; ")="; F(S); "H("; S; ")="; H(S)
3470 IF X = 16 GOTO 3530

3530 GOSUB 4210: GOSUB 4250: GOSUB 4580: GOTO 3552
3552 PRINT #1, "Z("; S; ")="; Z(S): NEXT S: GOTO 3610
3610 PRINT "IF I0=10 GOTO 3240-ПРОВЕРКА ТОЧНОСТИ И ";
3611 PRINT "РАСЧЕТЫ ПО МОДЕЛИ"
3612 PRINT "IF I0=11 GOTO 4880 - МАТЕМАТИЧЕСКАЯ МОДЕЛЬ"
3615 PRINT "IF I0=14 GOTO 7000-ВЫЧИСЛЕНИЯ ПОКАЗАТЕЛЕЙ Z(K5)"
3616 PRINT "          С ИСПОЛЬЗОВАНИЕМ ЦИКЛОВ И "
3617 PRINT "          ПОСТРОЕНИЕ ГРАФИКОВ"
3620 PRINT "IF I0=12 GOTO 6830-КОНЕЦ"
3625 PRINT "ВВОД I0": INPUT I0
3630 IF I0 = 10 GOTO 3240
3640 IF I0 = 11 GOTO 4880
3650 IF I0 = 12 GOTO 6830

```

```

3653 IF I0 = 14 GOTO 7000
3710 N0 = (A ^ N + B ^ N + C ^ N + D ^ N) / 4
3720 R0 = (A ^ R + B ^ R + C ^ R + D ^ R) / 4
3730 S0 = (A ^ S + B ^ S + C ^ S + D ^ S) / 4; L2 = 2 * N
3740 N3 = (A ^ L2 + B ^ L2 + C ^ L2 + D ^ L2) / 4; K2 = 2 * R
3750 R3 = (A ^ K2 + B ^ K2 + C ^ K2 + D ^ K2) / 4; N4 = N + R
3760 N5 = (A ^ N4 + B ^ N4 + C ^ N4 + D ^ N4) / 4; N6 = N + S
3770 N7 = (A ^ N6 + B ^ N6 + C ^ N6 + D ^ N6) / 4; R4 = R + S
3780 R5 = (A ^ R4 + B ^ R4 + C ^ R4 + D ^ R4) / 4; V0 = -N0
3790 U0 = (N0 * R0 - N5) / (N3 - N0 ^ 2); Q0 = -(R0 + U0 * N0)
3800 P0 = (N0 * S0 - N7) / (N3 - N0 ^ 2); Z1 = R0 * S0 - R5 + P0 * (N0 * R0 - N5)
3810 Z2 = U0 * (N0 * S0 - N7) + U0 * P0 * (N0 ^ 2 - N3)
3820 Z3 = R3 - R0 ^ 2 + 2 * U0 * (N5 - N0 * R0)
3830 I0 = (Z1 + Z2) / (Z3 + (N3 - N0 ^ 2) * U0 ^ 2); M0 = I0 * U0 + P0
3840 F0 = -(S0 + I0 * R0 + M0 * N0)
3850 PRINT #1, "КОЭФФИЦИЕНТЫ ОРТОГОНАЛИЗАЦИИ": RETURN
4210 FOR J = 1 TO X: I(J) = F(J) ^ J1 + V1
4220 K(J) = F(J) ^ O1 + U1 * F(J) ^ J1 + Q1
4230 L(J) = F(J) ^ P1 + I1 * F(J) ^ O1 + M1 * F(J) ^ J1 + F1: NEXT J
4240 RETURN
4250 FOR J = 1 TO X: P(J) = H(J) ^ J2 + V2
4260 Q(J) = H(J) ^ O2 + U2 * H(J) ^ J2 + Q2
4270 U(J) = H(J) ^ P2 + I2 * H(J) ^ O2 + M2 * H(J) ^ J2 + F2: NEXT J
4280 RETURN
4400 FOR J = 1 TO X: Z(J) = B(1) + B(2) * I(J) + B(3) * K(J) + B(4) * L(J)
4410 NEXT J: RETURN
4580 FOR J = 1 TO X: N3 = B(1) + B(2) * I(J) + B(3) * K(J) + B(4) * P(J)
4590 N4 = B(5) * I(J) * P(J) + B(6) * Q(J) + B(7) * I(J) * Q(J) + B(8) * P(J) * K(J)
4600 N5 = B(9) * K(J) * Q(J) + B(10) * U(J) + B(11) * I(J) * U(J) + B(12) * K(J) * U(J)
4610 N6 = B(13) * L(J) + B(14) * P(J) * L(J) + B(15) * Q(J) * L(J)
4620 Z(J) = N3 + N4 + N5 + N6 + B(16) * L(J) * U(J): NEXT J: RETURN
4880 PRINT #1, "МАТЕМАТИЧЕСКАЯ МОДЕЛЬ"
4905 IF X = 16 GOTO 6280
6280 PRINT #1, "Z(J)="; B(1); "+"; B(2); "*I(J)+"; B(3); "*K(J)+";
6290 PRINT #1, "+"; B(4); "*P(J)+"; B(5); "I(J)*P(J)+";
6300 PRINT #1, "+"; B(6); "*Q(J)+"; B(7); "*I(J)*Q(J)+";

6310 PRINT #1, "+"; B(8); "*P(J)*K(J)+"; B(9); "*K(J)*Q(J)+";
6320 PRINT #1, "+"; B(10); "*U(J)+"; B(11); "*I(J)*U(J)+";
6330 PRINT #1, "+"; B(12); "*K(J)*U(J)+"; B(13); "*L(J)+";
6340 PRINT #1, "+"; B(14); "*P(J)*L(J)+"; B(15); "*Q(J)*L(J)+";
6350 PRINT #1, "+"; B(16); "*L(J)*U(J),"
6360 IF X = 16 GOTO 6600
6600 PRINT #1, "ГДЕ"
6610 PRINT #1, "I(J)=F(J)^"; J1; "+"; V1; ","
6620 PRINT #1, "K(J)=F(J)^"; O1; "+"; U1; "*F(J)^"; J1; "+"; Q1
6621 PRINT #1, "ОБОЗНАЧЕНИЕ: F(J)- 1-й ФАКТОР "
6650 PRINT #1, "L(J)=F(J)^"; P1; "+"; I1; "*F(J)^"; O1; "+";
6660 PRINT #1, "+"; M1; "F(J)^"; J1; "+"; F1
6661 PRINT #1, "ОБОЗНАЧЕНИЕ: F(J)- 1-й ФАКТОР "
6673 IF X = 16 GOTO 6710
6710 PRINT #1, "P(J)=H(J)^"; J2; "+"; V2; ","
6720 PRINT #1, "Q(J)=H(J)^"; O2; "+"; U2; "*H(J)^"; J2; "+"; Q2; ","
6730 PRINT #1, "U(J)=H(J)^"; P2; "+"; I2; "*H(J)^"; O2; "+";
6740 PRINT #1, "+"; M2; "*H(J)^"; J2; "+"; F2
6741 PRINT #1, "ОБОЗНАЧЕНИЕ: H(J)- 2-й ФАКТОР"
6750 IF X = 16 GOTO 6790
6790 PRINT "IF I0=18 GOTO 2660-ПЕРЕХОДЫ"
6792 PRINT "IF I0=19 GOTO 3190-ПЕРЕХОДЫ "
6793 PRINT "IF I0=35 GOTO 1160-ВВОД НОВЫХ Y(J)"
6795 PRINT "IF I0=44 GOTO 6830-КОНЕЦ"
6796 PRINT "IF I0=50 GOTO 40-НАЧАЛО"
6797 PRINT "IF I0=51 GOTO 3240-ПРОВЕРКА ТОЧНОСТИ И "

```

```

6798 PRINT "          РАСЧЕТЫ ПО МОДЕЛИ"
6799 PRINT "IF I0=52 GOTO 7000-"
6800 PRINT "          ВЫЧИСЛЕНИЯ ПОКАЗАТЕЛЕЙ Z(K5) "
6802 PRINT "          С ИСПОЛЬЗОВАНИЕМ ЦИКЛОВ "
6803 PRINT "          И ПОСТРОЕНИЕ ГРАФИКОВ"
6805 PRINT "ВВОД I0": INPUT I0
6810 IF I0 = 18 GOTO 2660
6820 IF I0 = 19 GOTO 3190
6823 IF I0 = 35 GOTO 1160
6825 IF I0 = 44 GOTO 6830
6827 IF I0 = 50 GOTO 40
6828 IF I0 = 51 GOTO 3240
6829 IF I0 = 52 GOTO 7000
6830 CLOSE #1
6832 PRINT "РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ПРОГРАММЫ СМОТРИ В ";
6835 PRINT "ФАЙЛЕ "; FA$
6840 END
7000 PRINT #1, "ВЫЧИСЛЕНИЯ ПОКАЗАТЕЛЕЙ Z(K5)"
7004 PRINT #1, " ПО МАТЕМАТИЧЕСКОЙ МОДЕЛИ"
7005 PRINT #1, " С ИСПОЛЬЗОВАНИЕМ ЦИКЛОВ"
7006 PRINT #1, "И ПОСТРОЕНИЕ ГРАФИКОВ"
7010 PRINT "ВВОД I0=61 GOTO 7360"
7040 PRINT "IF I0=64 GOTO 40-НАЧАЛО"
7050 PRINT "IF I0=65 GOTO 6830-КОНЕЦ"
7060 INPUT I0
7100 IF I0 = 64 GOTO 40
7110 IF I0 = 65 GOTO 6830
7360 F3 = 0: F4 = 0: H3 = 0: H4 = 0: K5 = 0: PRINT #1, "ФАКТОР F(1)=F3+F4"
7361 PRINT "ФАКТОР F(1)=F3+F4"
7365 FOR J = 1 TO X: F(J) = 0: H(J) = 0: Z(J) = 0: NEXT J: X = 0
7370 PRINT #1, "F4-ШАГ ПРИРАЩЕНИЯ 1-ГО ФАКТОРА"
7371 PRINT "F4-ШАГ ПРИРАЩЕНИЯ 1-ГО ФАКТОРА"
7380 PRINT #1, "ФАКТОР H(1)=H3+H4"
7381 PRINT "ФАКТОР H(1)=H3+H4"
7390 PRINT #1, "H4-ШАГ ПРИРАЩЕНИЯ 2-ГО ФАКТОРА"
7391 PRINT "H4-ШАГ ПРИРАЩЕНИЯ 2-ГО ФАКТОРА"
7392 PRINT #1, "X-КОЛИЧЕСТВО ЗНАЧЕНИЙ 1, 2-ГО ФАКТОРОВ"
7393 PRINT "X-КОЛИЧЕСТВО ЗНАЧЕНИЙ 1, 2-ГО ФАКТОРОВ"
7400 PRINT "ВВОД ПРИНЯТЫХ ВЕЛИЧИН X,F3,F4,H3,H4"
7410 INPUT X, F3, F4, H3, H4: PRINT #1, "X="; X; "F3="; F3; "F4="; F4
7420 PRINT #1, "H3="; H3; "H4="; H4
7430 FOR K5 = 1 TO X: F(K5) = F3 + K5 * F4
7435 PRINT #1, "F("; K5; ")="; F(K5)
7440 H(K5) = H3 + K5 * H4: PRINT #1, "H("; K5; ")="; H(K5)
7540 GOSUB 4210: GOSUB 4250: GOSUB 4580: GOTO 7570
7570 PRINT #1, "Z("; K5; ")="; Z(K5)
7575 NEXT K5: GOTO 8001
8001 PRINT #1, "ВЫЯВЛЕНИЕ MAX Z(K5) И MIN Z(K5)": K8 = 0: K8 = Z(1)
8002 PRINT "ВВОД I0=90-ПРОДОЛЖЕНИЕ"
8004 INPUT I0
8010 FOR K5 = 1 TO X
8020 IF Z(K5) >= K8 THEN K8 = Z(K5)
8040 NEXT K5: PRINT #1, "MAX Z(K5)="; K8
8041 FOR K5 = 1 TO X
8042 IF Z(K5) = K8 THEN PRINT #1, "MAX Z("; K5; ")="; Z(K5)
8044 NEXT K5
8050 K7 = 0: K7 = Z(1)
8060 FOR K5 = 1 TO X
8070 IF Z(K5) <= K7 THEN K7 = Z(K5)
8090 NEXT K5: PRINT #1, "MIN Z(K5)="; K7

8091 FOR K5 = 1 TO X
8092 IF Z(K5) = K7 THEN PRINT #1, "MIN Z("; K5; ")="; Z(K5)

```

```

8094 NEXT K5: K6 = 0: PRINT #1, "MIN Z(K5)=K7, MAX Z(K5)=K8"
8095 PRINT #1, "K6(K5)=(Z(K5)+ABS(K7))/(ABS(K7)+ABS(K8))"
8096 FOR K5 = 1 TO X: K6(K5) = (Z(K5) + ABS(K7)) / (ABS(K7) + ABS(K8))
8097 PRINT #1, "K6("; K5; ")="; K6(K5): NEXT K5
8098 J5 = 0: J5 = ABS(K7) / (ABS(K7) + ABS(K8))
8099 PRINT #1, "J5=ABS(K7)/(ABS(K7)+ABS(K8))"
8111 PRINT #1, "J5="; J5
8112 PRINT "IF I0=70 GOTO 7000-ПОВТОРЕНИЕ ";
8113 PRINT " ВЫЧИСЛЕНИЙ ПОКАЗАТЕЛЕЙ Z(K5) ";
8114 PRINT " И ПОСТРОЕНИЕ ГРАФИКОВ"
8115 PRINT "IF I0=80 GOTO 9000-ПОСТРОЕНИЕ ГРАФИКА"
8120 INPUT I0
8125 IF I0 = 70 GOTO 7000
8130 IF I0 = 80 GOTO 9000
9000 PRINT "X0="; X0; "Y0="; Y0; "K0="; K0; "K3="; K3
9001 K0 = 0: K3 = 0: K4 = 0: K4 = X: K7 = 0: K8 = 0: X0 = 0: Y0 = 0
9010 PRINT #1, "ПОСТРОЕНИЕ ГРАФИКА"
9015 PRINT #1, "ЗАВИСИМОСТЬ K6(K5) ОТ ФАКТОРА"
9020 PRINT #1, "K6(K5)-ОТНОСИТЕЛЬНАЯ ВЕЛИЧИНА ПОКАЗАТЕЛЯ"
9025 PRINT #1, "K5-НОМЕР ВЕЛИЧИНЫ ФАКТОРА И ПОКАЗАТЕЛЯ"
9030 PRINT #1, "ВЕЛИЧИНЫ ФАКТОРОВ ЗАДАНЫ "
9035 PRINT "ВВОД: X0-ОТСТУП ВПРАВО ПО ОСИ X (ПРЕДПОЧТИТЕЛЬНО X0=20)"
9036 PRINT " Y0-ОТСТУП ВНИЗ ПО ОСИ Y (ПРЕДПОЧТИТЕЛЬНО Y0=180)"
9037 PRINT " K0-ДЛИНА ГРАФИКА ПО ОСИ X"
9038 PRINT " K3-ВЫСОТА ГРАФИКА ПО ОСИ Y"
9045 INPUT X0, Y0, K0, K3
9046 PRINT #1, "X0="; X0; "Y0="; Y0; "K0="; K0; "K3="; K3; ", ГДЕ"
9047 PRINT #1, " X0-ОТСТУП ВПРАВО ПО ОСИ X "
9048 PRINT #1, " Y0-ОТСТУП ВНИЗ ПО ОСИ Y "
9049 PRINT #1, " K0-ДЛИНА ГРАФИКА ПО ОСИ X"
9050 PRINT #1, " K3-ВЫСОТА ГРАФИКА ПО ОСИ Y"
9054 KEY OFF: CLS
9055 COLOR 0, 0: SCREEN 2
9056 FOR K5 = 1 TO K4: K7(K5) = K5 * K0: K8(K5) = K3 * K6(K5)
9057 LINE (K7(K5) - X0, Y0)-(K7(K5) - X0, Y0 - K8(K5)): NEXT K5
9059 J6 = 0: J6 = X - 1: J7 = 0: J8 = 0: J9 = 0: K7 = 0: K8 = 0: J9 = K3 * J5
9060 FOR K5 = 1 TO J6: K7(K5) = K5 * K0: K8(K5) = K3 * K6(K5)
9061 J7(K5) = (K5 + 1) * K0: J8(K5) = K3 * K6(K5 + 1)
9062 LINE (K7(K5) - X0, Y0)-(J7(K5) - X0, Y0)
9063 LINE (K7(K5) - X0, Y0 - J9)-(J7(K5) - X0, Y0 - J9)
9065 LINE (K7(K5) - X0, Y0 - K8(K5))-(J7(K5) - X0, Y0 - J8(K5)): NEXT K5
9071 A$ = ""
9072 A$ = INKEY$: IF A$ = "" THEN 9072
9073 SCREEN 0: CLS : COLOR 2, 0
9074 PRINT "ВВОД I0=75 GOTO 9000-ПОВТОРЕНИЕ ПОСТРОЕНИЯ ГРАФИКА"
9075 PRINT "ВВОД I0=85 GOTO 7000-ПОВТОРЕНИЕ"
9076 PRINT " ВЫЧИСЛЕНИЙ ПОКАЗАТЕЛЕЙ Z(K5)"
9078 PRINT " С ИСПОЛЬЗОВАНИЕМ ЦИКЛОВ "
9079 PRINT " И ПОСТРОЕНИЕ ГРАФИКОВ"
9080 PRINT "ВВОД I0=95 GOTO 6830-КОНЕЦ": PRINT
9081 INPUT I0
9083 IF I0 = 75 GOTO 9000
9090 IF I0 = 85 GOTO 7000
9095 IF I0 = 95 GOTO 6830
11250 PRINT #1, "1.1."; "A2="; A2; "A1="; A1; "Z(1)="; Z(1); "Z1(1)="; Z1(1)
11260 PRINT #1, "1.2."; "A2="; A2; "C1="; C1; "Z(9)="; Z(9); "Z1(9)="; Z1(9)
11270 PRINT #1, "1.3."; "A2="; A2; "D1="; D1; "Z(13)="; Z(13); "Z1(13)="; Z1(13)
11280 PRINT #1, "1.4."; "A2="; A2; "B1="; B1; "Z(2)="; Z(2); "Z1(2)="; Z1(2)
11290 PRINT #1, "2.1."; "C2="; C2; "A1="; A1; "Z(5)="; Z(5); "Z1(5)="; Z1(5)
11300 PRINT #1, "2.2."; "C2="; C2; "C1="; C1; "Z(10)="; Z(10); "Z1(10)="; Z1(10)
11310 PRINT #1, "2.3."; "C2="; C2; "D1="; D1; "Z(14)="; Z(14); "Z1(14)="; Z1(14)
11320 PRINT #1, "2.4."; "C2="; C2; "B1="; B1; "Z(6)="; Z(6); "Z1(6)="; Z1(6)
11330 PRINT #1, "3.1."; "D2="; D2; "A1="; A1; "Z(7)="; Z(7); "Z1(7)="; Z1(7)

```

```
11340 PRINT #1, "3.2."; "D2="; D2; "C1="; C1; "Z(11)="; Z(11); "Z1(11)="; Z1(11)
11350 PRINT #1, "3.3."; "D2="; D2; "D1="; D1; "Z(15)="; Z(15); "Z1(15)="; Z1(15)
11360 PRINT #1, "3.4."; "D2="; D2; "B1="; B1; "Z(8)="; Z(8); "Z1(8)="; Z1(8)
11370 PRINT #1, "4.1."; "B2="; B2; "A1="; A1; "Z(3)="; Z(3); "Z1(3)="; Z1(3)
11380 PRINT #1, "4.2."; "B2="; B2; "C1="; C1; "Z(12)="; Z(12); "Z1(12)="; Z1(12)
11390 PRINT #1, "4.3."; "B2="; B2; "D1="; D1; "Z(16)="; Z(16); "Z1(16)="; Z1(16)
11400 PRINT #1, "4.4."; "B2="; B2; "B1="; B1; "Z(4)="; Z(4); "Z1(4)="; Z1(4)
11410 PRINT #1, "5.1."; "A1="; A1; "A2="; A2; "Z(1)="; Z(1); "Z1(1)="; Z1(1)
11420 PRINT #1, "5.2."; "A1="; A1; "C2="; C2; "Z(5)="; Z(5); "Z1(5)="; Z1(5)
11430 PRINT #1, "5.3."; "A1="; A1; "D2="; D2; "Z(7)="; Z(7); "Z1(7)="; Z1(7)
11440 PRINT #1, "5.4."; "A1="; A1; "B2="; B2; "Z(3)="; Z(3); "Z1(3)="; Z1(3)
11450 PRINT #1, "6.1."; "C1="; C1; "A2="; A2; "Z(9)="; Z(9); "Z1(9)="; Z1(9)
11460 PRINT #1, "6.2."; "C1="; C1; "C2="; C2; "Z(10)="; Z(10); "Z1(10)="; Z1(10)
11470 PRINT #1, "6.3."; "C1="; C1; "D2="; D2; "Z(11)="; Z(11); "Z1(11)="; Z1(11)
11480 PRINT #1, "6.4."; "C1="; C1; "B2="; B2; "Z(12)="; Z(12); "Z1(12)="; Z1(12)
11490 PRINT #1, "7.1."; "D1="; D1; "A2="; A2; "Z(13)="; Z(13); "Z1(13)="; Z1(13)
11500 PRINT #1, "7.2."; "D1="; D1; "C2="; C2; "Z(14)="; Z(14); "Z1(14)="; Z1(14)
11510 PRINT #1, "7.3."; "D1="; D1; "D2="; D2; "Z(15)="; Z(15); "Z1(15)="; Z1(15)
11520 PRINT #1, "7.4."; "D1="; D1; "B2="; B2; "Z(16)="; Z(16); "Z1(16)="; Z1(16)
11530 PRINT #1, "8.1."; "B1="; B1; "A2="; A2; "Z(2)="; Z(2); "Z1(2)="; Z1(2)
11540 PRINT #1, "8.2."; "B1="; B1; "C2="; C2; "Z(6)="; Z(6); "Z1(6)="; Z1(6)
11550 PRINT #1, "8.3."; "B1="; B1; "D2="; D2; "Z(8)="; Z(8); "Z1(8)="; Z1(8)
11560 PRINT #1, "8.4."; "B1="; B1; "B2="; B2; "Z(4)="; Z(4); "Z1(4)="; Z1(4)
11570 RETURN
```

ОПЕРАТОРЫ ЯЗЫКА ПРОГРАММИРОВАНИЯ БЕЙСИК

PRINT (печатать)

Вывод данных на экран дисплея. Форматы:

PRINT [<список выражений>][;]

? [<список выражений>][;]

Выражения в списке отделяются друг от друга запятой, точкой с запятой или пробелом (пробелами). Точка с запятой и пробел (пробелы) дают одинаковый результат (между цифрами пробелы не воспринимаются). Если <список выражений> опущен, выводится пустая строка. ? используется как сокращенная запись слова PRINT.

DATA (данные)

DATA <константа> [, <константа>]...

Операторы DATA могут быть размещены в любом месте программы; все данные в DATA – непрерывный список для операторов READ. Типы констант в READ и DATA должны совпадать.

READ (читать)

Присваивание переменным значений, заданных в операторах DATA. Формат:

READ <переменная> [, <переменная>]...

Операторы READ читают подряд данные из операторов DATA.

RESTORE (восстановить)

Переустанавливает список данных, определенных операторами DATA, для повторного или выборочного чтения. Формат:

RESTORE [<номер строки>]

где <номер строки> - номер строки, содержащий оператор DATA, куда будет обращаться следующий оператор READ. Если номер строки не задан, то следующий оператор READ будет обращаться к первому элементу первого оператора DATA программы.

INPUT (ввести)

Ввод данных с клавиатуры во время выполнения программы или в командном режиме. Формат:

INPUT [;] [<подсказка>] <переменная> [, <переменная>]...

Текст подсказки поясняет, какие данные требуется ввести. При выполнении оператора INPUT на экране появляется ?, и программа ожидает ввода данных с клавиатуры. Если после подсказки использовать <, > вместо <;>, то ? не появляется. Вводимые данные следует разделять на экране запятыми. Тип вводимого элемента данных должен соответствовать типу переменной в перечне INPUT. Строки символов могут не заключаться в кавычки, если они не содержат запятых, начальных или конечных пробелов. Если в ответ на запрос оператора INPUT введено слишком мало или слишком много элементов данных, или не согласуются типы данных, то выдается сообщение об ошибке, после чего необходимо повторить весь ввод. Если требуется единственная переменная, то нажатие клавиши <Enter> без ввода данных на экран (пустой

ввод) приводит к тому, что числовая переменная принимает значение 0, а строковая – пустой строки. Если после слова INPUT стоит <;>, то нажатие клавиши ввода <Enter> не переводит курсор в начало следующей строки.

LINE INPUT (ввод строки)

Ввод строки символов с клавиатуры. Формат:

LINE INPUT [;] [«подсказка»,] <строковая переменная>

Оператор LINE INPUT позволяет вводить с клавиатуры строки с любыми символами. Знак ? на экране не появляется

Оператор присваивания **LET** (пусть, допустим)

Присваивание переменной значения выражения. Формат:

[LET] <переменная> = <выражение>

Слово LET обычно опускается.

PRINT USING (печать с использованием шаблона)

Вывод данных на экран в указанном формате. Формат:

PRINT USING <формат>; <список выражений> [;]

<Формат> для вывода строки символов может быть одним из следующих:

“!” – задает вывод только первого символа строки;

“\<n пробелов>” – задает вывод 2+n первых символов строки. Если не указано ни одного пробела (“\”), то выводятся два символа;

“&” – задает вывод всей строки символов.

Формат для вывода числа задается с помощью шаблона, который описывает каждую позицию, занимаемую числом в выводимой строке. Шаблон составляется из символов:

- описывает цифровую позицию числа. Цифровая позиция всегда присутствует в выводной строке и может содержать цифру или пробел. Пробелы появляются в крайних левых позициях, если в выводимом числе меньше цифр, чем определено цифровых позиций;

. - описывает местоположение десятичной точки в выводном формате числа;

+ - описывает знаковую позицию числа и может быть первым или последним символом в шаблоне. При выводе числа в эту позицию будет вставлен знак “+” или “-”;

^^^ - определяют экспоненциальный формат представления числа при выводе и могут быть указаны в шаблоне только после цифровых позиций.

Если выводимое число не вмещается в поле, определенное шаблоном, перед числом в выводную строку будет вставлен знак “%”. Разделители в конце оператора выполняют ту же роль, что и в операторе PRINT.

WRITE (писать)

Вывод данных на экран дисплея. Формат:

WRITE <список выражений>

Выражения в списке отделяются друг от друга запятой или точкой с запятой. Оператор WRITE выполняется аналогично оператору PRINT, но есть отличия: при выводе данные разделяются запятыми; строки символов заключаются в кавычки; перед положительным числом и после чисел не ставится пробел.

LPRINT, LPRINT USING

Вывод данных на принтер. Форматы:

LPRINT [<список выражений>] [;]

LPRINT USING <формат>; <список выражений>[;]

Операторы аналогичны операторам PRINT и PRINT USING.

Оператор **MID\$** (MIDDL – середина)

Замена заданной части строки символов другой строкой. Формат:

MID\$ (<строковая переменная>, n [,m]) = <строковое выражение> где n = 1...255, m = 0...255.

Оператор позволяет выполнить замену указанной подстроки в значении <строковой переменной>. Первые m символов строки, заданной <строковым выражением>, будут замещать m символов в значении <строковой переменной>, начиная с позиции n. Если m опущено, пересылается вся строка. Оператор не изменяет длину <строковой переменной>.

SWAP (обмен)

Обмен значениями двух переменных. Формат:

SWAP <переменная 1>, <переменная 2>

В результате выполнения оператора SWAP <переменная 1> получает значение <переменной 2>, а <переменная 2> - значение <переменной 1>.

RANDOMIZE (RANDOM – случайный)

Переустановка базы генерации случайных чисел. Формат:

RANDOMIZE [n]

где n – целочисленное выражение, значение которого используется в качестве базы генерации и равно -32768 ... +32767. База генерации предназначена для функции RND [(x)], где x – фиктивный параметр.

KEY (клавиша)

Установка или отображение значений функциональных клавиш F1...F10.

Форматы:

KEY n, <строка символов>

KEY LIST

KEY ON

KEY OFF

Здесь n – номер функциональной клавиши; <строка символов> - строковое выражение до 15 символов, значение которого назначается функциональной клавише; LIST – вывод на экран полных значений всех десяти функциональных клавиш; OFF – отменяет вывод на 25-ю строку экрана значений функциональных клавиш, но не отменяет эти значения; ON – выводит на 25-ю строку экрана значения функциональных клавиш.

При запуске Бейсика автоматически выполняется оператор KEY ON.

Если значение функциональной клавиши заканчивается символом CHR\$ (13) (<Enter>), то после нажатия функциональной клавиши нет необходимости нажимать <Enter> - соответствующая команда выполняется сразу.

Операторы переадресации управления

GOTO (перейти к ...)

Переход к заданной строке программы (безусловный переход). Формат:

GOTO <номер строки>

Оператор передает управление строке, номер которой указан в операторе. Если <номер строки> указывает на невыполняемый оператор (REM, DATA), то выполнение программы продолжается с первого последующего выполняемого оператора.

Оператор GOTO удобно исполнить также в режиме прямого выполнения команд, например, для запуска программы с заданной строки без потери значений переменных.

GOSUB, RETURN (программа ... возврат)

Переход к программе и возврат из нее. Формат:

GOSUB <номер строки> **RETURN** <номер строки>

Оператор GOSUB передает управление заданной подпрограмме, выполнение которой завершается оператором RETURN. По оператору RETURN без <номера строки> осуществляется возврат к оператору, следующему за оператором GOSUB. Вход в подпрограмму возможен в разных точках до оператора RETURN. Подпрограммы могут располагаться в любом месте программы, но следует позаботиться об их обходе оператором GOTO. Вызовы подпрограммы могут быть вложены в другие подпрограммы, причем вложенность ограничена только объемом свободной памяти.

ON ... GOTO, ON ... GOSUB (при ... перейти к)

Передача управления в зависимости от значения выражения. Форматы:

ON <числовое выражение> **GOTO** <номер строки> ... [, <номер строки>] ...

ON <числовое выражение> **GOSUB** <номер строки> ... [, <номер строки>] ...

При необходимости <числовое выражение> = 0...255 округляются до целого. Управление передается в ту строку программы, порядковый номер которой в списке оператора равен значению числового выражения. Если, например, L=3, то оператору

50 ON L GOTO 100,150, 300, 480

управление передается строке 300, так как она третья в списке.

В операторе ON ... GOSUB каждый номер строки должен указывать первую строку подпрограммы. Завершающий подпрограмму оператор RETURN без <номера строки> возвращает управление в последующий за оператором ON ... GOSUB оператор программы.

Если значение <числового выражения> равно 0 или превышает число указанных в списке номеров строк, управление передается следующему оператору.

IF (если)

Оператор условного перехода – управление ходом выполнения программы в зависимости от результата выполнения выражения. Форматы:

IF <выражение> [,] **THEN** <фраза> [**ELSE** <фраза>]

(если... тогда... в противном случае...)

IF <выражение> [,] **GOTO** <номер строки> [[,] **ELSE** <фраза>]

(если... то идти к ... в противном случае...)

Здесь <фраза> - оператор либо последовательность операторов Бейсика, разделенных двоеточиями, или номер строки, к которой должен быть осуществлен переход.

Результат выполнения оператора IF зависит от значения указанного в нем выражения.

Если значение <выражения> «истина», то выполняется действие, определенное во <фразе> за THEN, или GOTO. Если значение выражения «ложь», выполняется действие, определенное <фразой> за ELSE. Если ELSE отсутствует, выполняется следующая строка программы (именно следующая строка, а не следующий оператор данной строки; дело в том, что все операторы данной строки подчиняются одному и тому же IF ... THEN).

Операторы IF могут быть вложенными, при этом каждый ELSE объединяется с ближайшим THEN.

Операторы циклов

FOR... TO... STEP... NEXT

(для... до... с шагом... следующий)

Выполнение последовательностей инструкций в цикле. Форматы:

FOR<переменная> = x TO y [STEP z] <операторы> ... NEXT ...

[<переменная>] [<переменная>]...

(для ... от ... до ... с шагом ... <тело цикла> ... следующее значение счетчика или счетчиков)

Здесь: <переменная> - имя целочисленной переменной или переменной с простой точностью, которая используется в качестве счетчика цикла; x – числовое выражение, начальное значение счетчика; y – конечное значение счетчика; z – значение шага приращения счетчика; <операторы>, образующие тело цикла, выполняются до встречи с NEXT. После этого счетчик цикла увеличивается на z и полученное значение сравнивается с y. Если счетчик превышает y, то цикл заканчивается и управление передается оператору, следующему за NEXT. В противном случае <операторы> снова выполняются и т.д. Если STEP опущен, то по умолчанию z=1. Когда z<0, счетчик уменьшается при каждом проходе цикла, пока не станет меньше, чем y. Если условие цикла сразу не выполняется, то цикл пропускается. При z=0 получается бесконечный цикл.

Циклы могут быть вложены друг в друга. При этом имена счетчиков должны отличаться. Операторы NEXT для внутренних циклов располагаются раньше операторов NEXT для внешних циклов. Если вложенные циклы заканчиваются в одной точке, они могут быть завершены одним NEXT с перечнем имен счетчиков всех циклов (сначала внутренних, потом внешних через запятые). Переменные в операторе NEXT могут быть опущены, тогда NEXT считается относящимся к ближайшему FOR. Если используются вложенные циклы, то в каждом NEXT должна находиться своя переменная.

WHILE и WEND

(пока и WHILE END – конец цикла WHILE)

Организация цикла с предусловием. Формат:

WHILE <числовое выражение>...<тело цикла>... WEND.

Здесь <тело цикла> - последовательность операторов Бейсика.

Операторы WHILE и WEND организуют циклическое выполнение операторов, входящих в <тело цикла>. Если значение <числового выражения> не равно 0, операторы, заключенные между WHILE и WEND, выполняются и управление снова возвращается к оператору WHILE. Этот процесс повторяется до тех пор, пока <числовое выражение> не примет значение 0. В этом случае выполнение программы продолжается с оператора, следующего за WEND.

Допускается использование вложенных циклов WHILE ... WEND.

Соответствие между операторами WHILE и WEND устанавливается таким образом, что каждому WEND ставится в соответствие ближайший предшествующий ему WHILE, который еще не поставлен в соответствие никакому WEND.

Оператор **CLS** (очистить экран)

Стирание экрана. Формат:

CLS

Вся информация с экрана удаляется, а курсор размещается в левом верхнем углу экрана.

Оператор **LOCATE** (расположить)

Управление курсором. Формат:

LOCATE [<строка>] [, [<колонка>] [, [<курсor>]]]

LOCATE перемещает курсор в заданную позицию экрана <строка>, <колонка> (обычно <строка> = 1...24, <колонка> = 1...80).

<Курсор> = 0 или 1. Если <курсor> = 1, то во время выполнения программы курсор виден на экране. Если <курсor> = 0, то курсор невидим. По умолчанию везде используются старые значения.

Оператор **WIDTH** (ширина)

Установка ширины выходной строки на дисплее или на принтере.

Форматы:

WIDTH <ширина строки> (на дисплее для PRINT, LIST и др.)

WIDTH LPRINT <ширина строки> (на принтере для LPRINT и LLIST).

Здесь <ширина строки> - числовое выражение, которое может принимать значения 40 или 80 (экран) или 1...255 (принтер).

Функция **SPC**

Вставка указанного числа пробелов в выводную строку. Формат:

SPC (n)

где n = 0...255.

Функция участвует только в операторах типа PRINT.

Функция **TAB**

Переход к указанной позиции выводной строки. Формат:

TAB (n)

где n- номер позиции в строке. Используется только в операторах типа PRINT. Если n меньше текущего номера позиции в строке, осуществляется переход к позиции n в следующей строке.

Функция **SCREEN**

Функция символа, находящегося на экране. Формат:

SCREEN (<строка>, <столбец>, <режим>)

где <строка>, <столбец> - числовые выражения, задающие координаты позиции экрана; <режим> = 0 или 1.

Если <режим> = 0, то определяется код символа; если <режим> = 1, то определяется цвет символа и фона.

CHAIN (соединять)

Загрузка программы в основную память и передача ей управления.

Формат:

CHAIN [**MERGE**] <имя файла>[, [**строка**] [, [**ALL**] [, **DELETE** <границы>]]]

Здесь <имя файла> - загружаемая с диска программа; <строка> - номер строки, с которой начинается выполнение программы; <границы> - диапазон номеров удаляемых строк программы.

Загруженная программа выполняется с указанной или первой строки. Если задан **MERGE**, то загружаемая программа объединяется с программой, размещенной в памяти (как по команде **MERGE**). В противном случае новая программа полностью заменяет старую. **ALL** указывает, что все переменные текущей программы становятся доступными загруженной программе. Если **ALL** отсутствует, для передачи переменных можно воспользоваться оператором **COMMON**.

COMMON (общий)

Передача данных загружаемой программе. Формат:

COMMON <переменная> [, <переменная>]...

Если передается массив, его имя должно сопровождаться парой круглых скобок. Оператор может быть указан в любом месте программы. В программе может использоваться любое число операторов **COMMON**, но одна и та же переменная не должна появляться более, чем в одном операторе **COMMON**.

DEF FN (определение функции)

Определение функции пользователя. Формат:

DEF FN <имя> [(<параметр> [, <параметр>]...)] = <выражение>

Здесь <имя> - имя функции, любое допустимое имя переменной, используется для вызова функции; <параметр> - определяет имена параметров, которые получают значения во время вызова функции; тип <выражения> должен соответствовать типу, объявленному именем функции.

Параметр, который используется в определении функции, действует только в определении функции. Он не связан с переменными программы, имеющими те же имена. В <выражении>, определяющем функцию, можно использовать переменные, не являющиеся параметрами.

Перед вызовом функции необходимо выполнить оператор **DEF FN** для того, чтобы определить функцию.

DEFINT, **DEFISNG**, **DEFDBL**, **DEFSTR**

(целый, простой, двойной, строка)

Установка способа определения типа переменной по умолчанию. Формат:

DIF ... <буква> [-<буква>] [, <буква> [-<буква>]]...

<Буква> может быть любой буквой латинского алфавита и задает начальную букву имени переменной. Если используется форма <буква> -<буква>, то определяется диапазон начальных букв. Операторы объявляют, что переменные, начинающиеся с определенной буквы, будут иметь следующие типы:

... INT – целые (%);

... SNG – с простой точностью (!);

... DBL – с двойной точностью (#);

... STR - строковые (символьные, \$).

Знаки явного определения типа переменной сохраняют свое действие и имеют приоритет над этими операторами. Все переменные без объявленного типа являются по умолчанию переменными с простой точностью.

DIM (измерение)

Определение массива. Формат:

DIM <переменная> (<измерения>) [, <переменная> (<измерения>)]...

Здесь <измерения> - список числовых выражений, разделенных запятыми. Каждое выражение определяет границу измерения массива. Значение измерения должно быть положительным числом, целая часть которого определяет границу измерения. Начальное значение элементов числового массива равно 0; элементы символьного массива имеют переменную длину и начальное значение, равное пустой строке. Оператор DIM в программе должен быть обязательно выполнен ранее операций с элементами объявленных в нем массивов.

OPTION BASE (базовый режим индексирования)

Установка нижней границы индексов массивов. Формат:

OPTION BASE n

где n=0 или 1. По умолчанию нижняя граница индексов равна 0. Если выполнен оператор OPTION BASE 1, то наименьшее значение индекса становится равным 1. Оператор должен быть выполнен до определения или использования массива.

ERASE (стереть)

Удаление массивов. Формат:

ERASE <имя массива> [, <имя массива>]...

В результате выполнения оператора ERASE заданные массивы удаляются. Память, занимаемая массивами, освобождается и может быть использована для других целей. Удаленный массив может быть определен заново с другими границами и размерностью.

ON ERROR (при ошибке)

Определение подпрограммы обработки ошибок. Формат:

ON ERROR GOTO <номер строки>

где <номер строки> - номер строки первого оператора подпрограммы обработки ошибок. После выполнения оператора ON ERROR возникновение любой ошибки вызовет передачу управления указанной в операторе подпро-

грамме. Выход из подпрограммы обработки ошибок осуществляется с помощью оператора RESUME. Чтобы отменить обработку ошибок, необходимо выполнить оператор ON ERROR GOTO 0. После выполнения такого оператора возникновение ошибки вызовет выдачу соответствующего сообщения и прекращение выполнения программы.

RESUME (продолжать после перерыва)

Возврат из подпрограммы обработки ошибок. Формат:

RESUME <номер строки>

где <номер строки> - номер строки, начиная с которой будет продолжаться выполнение программы.

END (конец)

Завершение выполнения программы. Формат:

END

Оператор END переводит Бейсик на уровень команд. Наличие оператора END в конце программы не обязательно.

STOP (остановка)

Остановка выполнения программы. Формат:

STOP

Выполнение программы можно продолжить операторами CONT или GOTO <номер строки> в режиме команд.

Невыполняемый оператор **REM** (комментарий)

Вставка в программу комментария. Формат:

REM <комментарий>

Оператору REM может быть передано управление операторами GOTO и GOSUB. В этом случае выполнение программы продолжается с первого следующего выполняемого оператора. Вместо: REM возможен апостроф'.

В текстовом режиме работы дисплея цвет символа и его фона устанавливается оператором **COLOR**. На экране одновременно могут находиться и участки различного цвета, и разноцветные символы. Формат оператора:

COLOR [<передний план>] [, [<фон>] [, <окаймление>]]

<передний план> - цвет символов, числовое выражение от 0 до 31; <цвет фона>, от 0 до 7; <окаймление> - цвет фона на краях экрана, от 0 до 7 (некоторые дисплеи на команды, связанные с <окаймлением>, не реагируют).

Оператор COLOR <передний план> изменяет только цвет символов; оператор COLOR, <фон> - только цвет фона и т.д. Номера цветов приведены ниже:

Номер	Цвет	Номер	Цвет	Номер	Цвет	Номер	Цвет
	Черный	4	Красный	8	Серый	12	Розовый
	Синий	5	Пурпурный	9	Светлосиний	13	Светлопурпурный
	Зеленый	6	Коричневый	10	Светлозеленый	14	Желтый
	Голубой	7	Белый	11	Светлоголубой	15	Яркобелый

Если к этим номерам прибавить 16, то получим мерцающие знаки того же цвета: $3 + 16 = 19$ – голубой; $14 + 16 = 30$ – желтый и т.п.

Файлы произвольного (прямого) доступа

В файлах прямого доступа данные организуются таким образом, что доступ к ним может быть осуществлен в любом порядке независимо от того, в какой последовательности данные помещаются в файл. Запись и чтение информации с диска в случае прямого доступа осуществляются отдельными блоками, называемыми записями. Каждая запись имеет свой номер, в соответствии с которым она помещается в файл или извлекается из него. Для создания файла прямого доступа в программу должны быть включены следующие операции:

- 1) открытие файла (оператор **OPEN**);
- 2) распределение памяти для переменных в буфере файла – отделе памяти компьютера (оператор **FIELD**);
- 3) пересылка данных в буфер файла (операторы **LSET** и **RSET**). При пересылке в буфер числовые величины должны представляться строками знаков. Для этого можно использовать функции **MKIS**, **MKSS** и **MKDS**;
- 4) запись данных, находящихся в буфере, на диск (оператор **PUT**);
- 5) закрытие файла (оператор **CLOSE**).

Для обработки файла прямого доступа обычно используются операции:

- 1) открытие файла (**OPEN**);
- 2) распределение памяти в буфере (**FIELD**);
- 3) считывание нужной записи в буфер (оператор **GET**);
- 4) выполнение, если необходимо, обратного преобразования строк в числовые значения с помощью функций **CVI**, **CVS** и **CVD**);
- 5) закрытие файла (**CLOSE**).

Оператор **OPEN** (открыть)

Подготовка к выполнению операций ввода-вывода. Форматы:

OPEN "R", # <номер файла>, <имя файла> [, <длина записи>]

OPEN <имя файла> **AS** [#] <номер файла> [**LEN** = <длина записи>]

Здесь: <номер файла> - целочисленное выражение от 1 до 3; <длина записи> - числовое выражение, значение которого определяет длину записи в байтах, 1 ... 32767; по умолчанию – 128.

Если открывается несуществующий файл, то при выполнении оператора **OPEN** файл будет создан на диске.

Оператор **FIELD** (поле)

Распределение памяти для переменных в буфере файла. Формат:

FIELD # <номер файла>, <поле> **AS** <переменная> [, <поле> **AS** <переменная>]...

где <поле> - числовое выражение, задающее длину в байтах поля в буфере файла для соответствующей строковой <переменной>. Длина слова равна числу символов в нем; целое число преобразуется в строку длиной 2 байта; число с простой точностью – 4 байта; число с двойной точностью – 8 байт.

Оператор FIELD на помещает данные в буфер файла. Для этого служат операторы LSET и RSET. Он также не ведет обмен между диском и буфером. Это делают операторы GET и PUT. Оператор FIELD только объявляет переменные, для которых будет отведено место в буфере файла. Если переменная, определенная ранее в операторе FIELD, используется в операторе присваивания слева от "=", то она считается объявленной заново и становится обычной переменной, не связанной с буфером файла.

Операторы **LSET** и **RSET** (поместить слева и справа)

Пересылка данных в буфер файла прямого доступа. Формат:

LSET (или RSET) <строковая переменная> = x \$

где <строковая переменная> - имя переменной, определенной в операторе FIELD.

Операторы LSET и RSET предназначены для пересылки данных в буфер файла при подготовке к выполнению оператора PUT.

Строка, определяемая выражением x\$, помещается в поле, указанное <строковой переменной>. Если длина строки x\$ превышает длину переменной в операторе FIELD, то строка усекается в соответствии с размером переменной. При этом теряются крайние правые символы.

Если длина строки меньше длины переменной, то в случае оператора LSET данные прижимаются к левому краю поля, занимаемого переменной, а в случае RSET – к правому. Незанятые позиции в поле переменной заполняются пробелами. Числовые величины для операторов LSET и RSET должны быть представлены строками символов, для чего служат функции MKI\$, MKS\$ и MKD\$.

Операторы LSET и RSET также можно использовать для пересылки данных в поле, не определенное в операторе FIELD.

Оператор **PUT** (поместить)

Запись из буфера на диск. Формат:

PUT # <номер файла> [, <номер записи>]

где <номер записи> - целое число от 1 до 32767. Под этим номером запись будет помещена в файл и может быть впоследствии извлечена из него. Если <номер записи> не указан, записи присваивается следующий по порядку номер (относительно последнего оператора PUT).

Оператор **GET** (получить)

Считывание с диска в буфер. Формат:

GET # <номер файла> [, <номер записи>]

Оператор **CLOSE** (закрыть)

Завершает обработку файла (закрывает файл). Формат:

CLOSE [#<номер файла> [, #<номер файла>]...]

где <номер файла> - целое число, поставленное в соответствие файлу во время его открытия. Выполнение оператора в формате CLOSE закрывает все файлы. После закрытия файла связанный с ним номер может быть использован для открытия другого файла.

Функции **MKI\$, MKS\$, MKD\$**

(делать ... целый, простой, двойной)

Представление величины числового типа строкой знаков. Формат:

MKI\$ (<целочисленное выражение>)

MKS\$ (<выражение с простой точностью>)

MKD\$ (<выражение с двойной точностью>)

При пересылке числовых величин в буфер файла прямого доступа с помощью операторов LSET и RSET их значения в памяти ПЭВМ должны рассматриваться Бейсиком как строки символов. Функции MKI\$, MKS\$ и MKD\$ обеспечивают такую интерпретацию значений числовых величин. В отличие от STR\$ эти функции не изменяют вид данных в основной памяти. Они изменяют только способ интерпретации этих данных. Функция MKI\$ позволяет рассматривать целое число как строку длиной 2 символа, функция MKS\$ - число с простой точностью как 4 символа, MKD\$ - число с двойной точностью как 8 символов.

Функции **CVI**, **CVS**, **CVD** (переделать)

Представление строки символов величиной числового типа. Форматы:

CVI (<строка-2>)

CVS (<строка-4>)

CVD (<строка-8>)

где <строка-2>, <строка-4>, <строка-8> - строки символов длиной соответственно 2, 4 и 8 байт, полученные в результате выполнения функций MKI\$, MKS\$ и MKD\$.

В буфере файла прямого доступа все числовые величины представлены строками символов, полученными в результате выполнения функций MKI\$, MKS\$ и MKD\$. Функции CVI, CVS и CVD выполняют обратное преобразование при извлечении числовых величин из буфера файла. Их применение обязательно, если над числовыми величинами, прочитанными из файла прямого доступа, должны выполняться арифметические операции.

Последовательные файлы

В последовательном файле данные размещаются в той последовательности, в которой поступают в файл. При чтении такого файла данные становятся доступными в том порядке, в котором они были записаны.

Для создания последовательного файла в программу включаются следующие операции:

- 1) открытие файла (оператор **OPEN**);
- 2) запись данных в файл (операторы **PRINT#**, **PRINT#USING**, **WRITE#**);
- 3) закрытие файла (**CLOSE**).

Программа обработки последовательного файла включает следующие операции:

- 1) открытие файла (оператор **OPEN**);
- 2) чтение данных из файла (операторы **INPUT#**, **LINE INPUT#**);
- 3) закрытие файла (**CLOSE**).

Оператор **OPEN**

Подготовка к выполнению операций ввода-вывода. Форматы:

OPEN "O", #<номер файла>, <имя файла>

(для вывода из оперативной памяти ПЭВМ на диск)

OPEN "Г", #<номер файла>, <имя файла>

(для ввода с диска в оперативную память)

OPEN <имя файла> FOR OUTPUT AS #<номер файла>

(для вывода из оперативной памяти ПЭВМ на диск)

OPEN <имя файла> FOR INPUT AS #<номер файла>

(для ввода с диска в оперативную память).

Операторы **PRINT#** и **PRINT# USING**

Вывод данных в последовательный файл. Формат:

PRINT # <номер файла>, [USING <формат>;] <список выражений>

Оператор **WRITE #**

Вывод данных в последовательный файл. Формат:

WRITE # <номер файла>, <список выражений>

Символьные величины записываются на диск в кавычках!

Оператор **INPUT #**

Чтение данных из последовательного файла. Формат:

INPUT # <номер файла>, <переменная> [, <переменная>]...

Оператор INPUT # аналогичен оператору INPUT.

Оператор **LINE INPUT #**

Чтение строки символов из последовательного файла. Формат:

LINE INPUT # <номер файла>, <строковая переменная>

Экран графического дисплея представляет собой сумму точек, способных изменять свой цвет. Точки образуют строки (координата Y) и столбцы (координата X). В левом верхнем углу экрана X=0, Y=0; в правом нижнем углу экрана X=X_{max}, Y=Y_{max}.

Оператор **SCREEN 0** устанавливает текстовый режим работы дисплея;

SCREEN 1 – графический режим средней разрешающей способности (цветной экран 200 строк и 320 столбцов точек, X=0...319, Y=0...199);

SCREEN 2 – графический режим высокой разрешающей способности (монокромное изображение, экран 200 строк x 640 столбцов, X=0...639, Y=0...199).

Переходы от режима к режиму:

от 0 к 1, от 1 к 2, от 0 к 2

SCREEN 1

SCREEN 2

от 2 к 0, от 2 к 1

SCREEN 0

SCREEN 1

от 1 к 0

SCREEN 2: SCREEN 0

В графическом режиме SCREEN 1 цветом управляет оператор COLOR.

Формат:

COLOR [<фон>] [, <палитра>]

где <фон> - цвет фона, от 0 до 15; <палитра> - номер палитры цветов, 0 или 1. Цвета палитры приведены ниже:

<i>Номер цвета</i>	<i>Палитра 0</i>	<i>Палитра 1</i>
0	Фон	Фон
1	Зеленый	Голубой
2	Красный	Пурпурный
3	Коричневый	Белый

Для символов, вводимых с клавиатуры, действует цвет 3. Все эти символы на всем экране окрашены одинаково.

PSET, PRESET (отображение точки)

Форматы:

PSET (x, y) [, <цвет>]

PRESET (x, y) [, <цвет>]

где x, y – координаты точки; <цвет> = 0, 1, 2, 3.

Если <цвет> опущен, то в PSET <цвет> = 3, в PRESET <цвет> = 0.

LINE (отрезок прямой)

Формат:

LINE [(x1, y1) – (x2, y2)] [, [<цвет>] [, [B[F]]]]

где (x1, y1) – координаты начальной точки отрезка; если (x1, y1) не заданы, по умолчанию предполагаются координаты последней выводимой точки; (x2, y2) – координаты конечной точки отрезка; <цвет> = 0, 1, 2, 3; B, BF – строить прямоугольник с вершинами в точках (x1, y1) и (x2, y2); BF – окрасить прямоугольник в заданный <цвет>.

CIRCLE (окружность)

Формат:

CIRCLE (x, y), <радиус> [, [<цвет>] [, [<начало>] [, [<конец>] [, <сжатие>]]]]

где x, y – координаты центра окружности; <сжатие> - отношение размеров центра окружности по вертикали к размерам по горизонтали; если <сжатие> < 1, то <радиус> направлен по x; если <сжатие> >= 1, то <радиус> направлен по y; <начало>, <конец> - радианная мера дуг в начальной и конечной точках неполной окружности (верхняя полуокружность 0, 3.14, правая полуокружность 4.71, 1.57 и т.д.); если углы отрицательные, то они воспринимаются как положительные, но концы дуги соединяются с центром радиусами.

PAINT (раскраска областей экрана)

Область образуется замкнутой кривой заданного цвета вокруг заданной точки с координатами x, y. Формат:

PAINT (x, y) [, [<трафарет>] [, [<контур>]]

где <трафарет> = 0, 1, 2, 3 – цвет раскраски; по умолчанию 3; <контур> - цвет ограничивающей кривой; по умолчанию <контур> = <трафарет>.

Операторы **GET** и **PUT**

GET читает информацию о цветах всех точек заданной прямоугольной области экрана и помещает ее в числовой массив. PUT воспроизводит на экране изображение, хранящееся в числовом массиве.

Формат оператора GET:

GET (x1, y1) – (x2, y2), <имя массива>

где (x1, y1), (x2, y2) – координаты вершин прямоугольной области; <имя массива> - числовой массив, где хранится информация.

Объем памяти в байтах, отведенный для массива, должен быть не меньше, чем $4 + \text{INT}((m * A + 7) / 8 * n)$, где n, m – длины горизонтальной и вертикальной сторон прямоугольника, выраженные числом точек экрана; A=2 при средней разрешающей способности и A=1 при высокой разрешающей способности.

ОСОБЕННОСТИ ПРИМЕНЕНИЯ WINDOWS

Программа для семейства Windows представляет собой последовательный программный код, записанный в нотации используемого языка программирования. Программная среда, применяемая при разработке данной методики математического моделирования – Microsoft QuickBASIC v4.5.

Установка Microsoft QuickBASIC v4.5 в Windows:

Запускаем файл Setup.exe

Следуем запросам программы установщика (при запросе каталога в котором будет находиться Microsoft QuickBASIC v4.5 необходимо создать каталог в корне любого раздела, имя которого состоит из букв латинского алфавита и не превышает 8 символов).

По завершению процесса копирования установочных файлов, программа Microsoft QuickBASIC v4.5 готова к использованию.

Microsoft QuickBASIC v4.5 является MS-DOS приложением и имеет ряд особенностей при использовании в различных версиях Windows.

Режим MS-DOS – возможность в системах Windows 95\98, позволяющая DOS –приложениям запускаться, так как данным приложениям требуется полный контроль над ресурсами компьютера. Режим MS-DOS означает, что Windows полностью выгружает себя из памяти, оставляя только заглушку, позволяющую пользователю снова запустить Windows. При запуске MS-DOS Windows завершает все работающие задачи, и возврат в Windows возможен только путем прекращения работы оболочки DOS. Эта возможность совместимости с DOS реализована только в ветви Windows 9x.

Линейка версий Windows NT не поддерживает режим MS-DOS. Это связано с основополагающими принципами операционных систем класса NT (надежность и безопасность). Хотя в версиях Windows 2000 и XP сохраняется обратная совместимость старых программ. Это достигается за счет встроенного MS-DOS эмулятора, запускающегося автоматически при щелчке на необходимом исполняемом файле программы.

Его работа заключается в следующем:

- Пользователь запускает DOS программу;
- Windows проверяет на совместимость своей версии DOS с программой пользователя.
- В случае удачи разрешает использование этой программы (в случае не удачи выдает сообщение об ошибке).

Microsoft QuickBASIC v4.5 совместима со всеми версиями операционной системы Windows. В Microsoft QuickBASIC v4.5 реализована поддержка драйверов мыши.

Использование Microsoft QuickBASIC v4.5 в среде Windows.

Запускаем файл QB.EXE

Открывается среда разработки, где пользователь должен набирать код создаваемой программы. На рис. 1 приведен вид окна Microsoft QuickBASIC v4.5 в Windows XP.

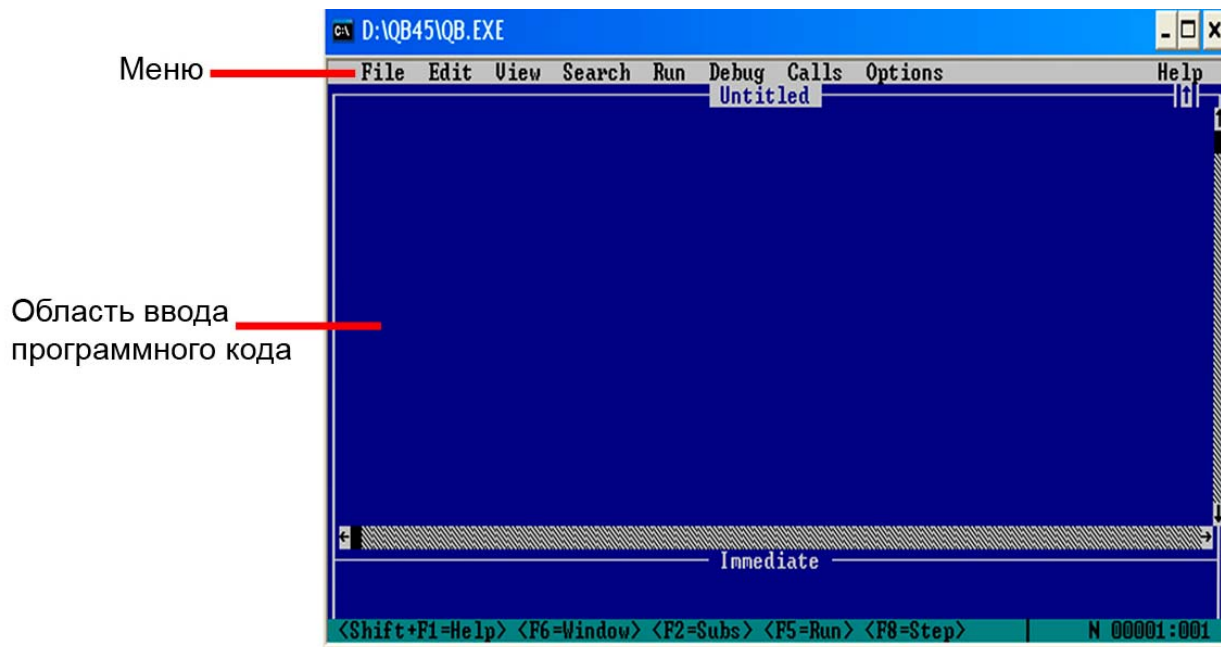


Рис 1. Вид окна Microsoft QuickBASIC v4.5 в среде Windows

ЗАДАНИЕ ДЛЯ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ

Выявить зависимость угара металла $Y_{\text{мет}}$ от количества стали в шихте газовой вагранки $Ш_c$ и температуры воздуха-окислителя T_B при изменении коэффициента расхода воздуха α по формуле $\alpha=1,05-0,000172 \cdot T_B$.

Для моделирования использовать программу NV6, план 4^2 :

$$X=16;$$

$Ш_c$ в процентах на четырех уровнях первого фактора $A1 = 0$; $C1 = 25$; $D1 = 75$; $B1 = 100$; T_B в градусах К на четырех уровнях второго фактора $A2 = 293$; $C2 = 438$; $D2 = 728$; $B2 = 873$.

В соответствии с планом проведения экспериментов 4^2 ($X = 16$)

$Y_{\text{мет}}$ в процентах

$Y(1) = 7,5$; $Y(2) = 100$; $Y(3) = 1,5$; $Y(4) = 15$; $Y(5) = 6$; $Y(6) = 95$; $Y(7) = 2,5$; $Y(8) = 52$; $Y(9) = 20$; $Y(10) = 17,5$; $Y(11) = 10$; $Y(12) = 2,5$; $Y(13) = 67$; $Y(14) = 61$; $Y(15) = 34,5$; $Y(16) = 9$.

Величины показателей степени в уравнении регрессии $J1 = 1$; $O1 = 2$; $P1 = 3$; $J2 = 1$; $O2 = 2$; $P2 = 3$; другие величины $N0 = 4$; $F8 = 3$; $U9 = 0,1667$; $T0 = 3,182$; $F7 = 8,69$.

Для расчетов показателей процесса по математической модели использовать следующие величины факторов:

$F(1) = 0$;	$H(1) = 583$;	$F(2) = 100$;	$H(2) = 583$;
$F(3) = 50$;	$H(3) = 293$;	$F(4) = 50$;	$H(4) = 873$;
$F(5) = 50$;	$H(5) = 583$;	$F(6) = 50$;	$H(6) = 438$;
$F(7) = 50$;	$H(7) = 728$;	$F(8) = 25$;	$H(8) = 583$;
$F(9) = 75$;	$H(9) = 583$;	$F(10) = 0$;	$H(10) = 486$;
$F(11) = 100$;	$H(11) = 486$;	$F(12) = 0$;	$H(12) = 680$;
$F(13) = 100$;	$H(13) = 680$;	$F(14) = 33,3$;	$H(14) = 293$;
$F(15) = 33,3$;	$H(16) = 486$;	$F(16) = 66,7$;	$H(16) = 837$.

С использованием циклов построить графики при $X = 10$; $F3 = 10$; $F4 = 0$; $H3 = 200$; $H4 = 80$.

На основе данных системного анализа результатов расчетов по математической модели с применением Мастера диаграмм Excel построить графики и выполнить их анализ.

ВЫПОЛНЕНИЕ ПРОГРАММЫ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ

Для запуска программы NV6 на решение в Microsoft QuickBASIC v4.5 необходимы следующие действия.

В меню «File» выбираем пункт «Open Program...» путем нажатия клавиши «Enter» («Ввод») на клавиатуре или щелчком левой кнопки мыши.

В открывшемся окне выбираем файл «NV6.BAS» путем нажатия клавиши «Enter» («Ввод») на клавиатуре или щелчком левой кнопки мыши.

В области ввода программного кода открывается текст программы NV6.

В меню «Run» выбираем пункт «Start...» путем нажатия клавиши «Enter» («Ввод») на клавиатуре или щелчком левой кнопки мыши.

Открывается окно ввода данных.

Алгоритм действий пользователя, направленный на реализацию примера математического моделирования по программе NV6, приведен в табл. 3.

Таблица 3

Сообщение на экране монитора	Вводимые величины ¹	Комментарий
Ввод имени файла	NV61	Возможно введение произвольного имени файла
Ввод X-количество опытов по плану	16	$X=16=4^2$, т. к. производится планирование экспериментов на четырех уровнях факторов
Ввод A1,C1,D1,B1,J1,O1,P1	0,25,75,100,1,2,3	Ввод величин факторов и показателей степени производится через запятую без пробелов
Ввод A2,C2,D2,B2,J2,O2,P2	293,438,728,873,1,2,3	Ввод производится через запятую без пробелов
IF I0=6 GOTO 40-НАЧАЛО IF I0=7 GOTO 1160-ПРОДОЛЖЕНИЕ	7	Если предыдущие значения величин введены не верно, то вводится 6; если верно, то 7
Y(1)	7.5	Ввод производится через точку (7.5)
Y(2)	100	
Y(3)	1.5	Ввод производится через точку (1.5)
Y(4)	15	
Y(5)	6	
Y(6)	95	
Y(7)	2.5	Ввод производится через

		точку (2.5)
Y(8)	52	
Y(9)	20	
Y(10)	17.5	Ввод производится через точку (17.5)
Y(11)	10	
Y(12)	2.5	Ввод производится через точку (2.5)
Y(13)	67	
Y(14)	61	
Y(15)	34.5	Ввод производится через точку (34.5)
Y(16)	9	
IF I0=1 GOTO 1160- ПОВТОРИТЬ ВВОД ПО- КАЗАТЕЛЕЙ IF I0=2 GOTO 1270- ПРОДОЛЖЕНИЕ	2	Если предыдущие значения величин введены не верно, то вводится 1; если верно, то 2
Ввод N0-количество опытов на среднем уровне факторов	4	N0 = 4
Ввод F8=N0-1	3	F8 = 4 - 1 = 3
Ввод I0	4	Ввод дисперсии опытов
Ввод U9 – ввод дисперсии опытов	0.1667	Ввод производится через точку (0.1667)
Ввод T0 – табличный T-критерий	3.182	Ввод производится через точку (3.182)
Ввод F7 – табличный F-критерий	8.69	Ввод производится через точку (8.69)
АДЕКВАТНО IF I0=7 GOTO 3240 – проверка точности и расчеты по модели IF I0=8 GOTO 40 – начало IF I0=17 GOTO 4880 – математическая модель IF I0=22 GOTO 7000 – вычисления показателей Z(K5) с использованием циклов и построение графиков IF I0=9 GOTO 6830 – конец	17	
IF I0=18 GOTO 2660 – переходы IF I0=19 GOTO 3190 – переходы IF I0=35 GOTO 1160 – ввод новых Y(J) IF I0=44 GOTO 6830 – конец IF I0=50 GOTO 40 – начало IF I0=51 GOTO 3240 – проверка точности и расчеты по модели	51	

IF I0=52 GOTO 7000 – вычисления показателей Z(K5) с использованием циклов и построение графиков		
Ввод F(1),H(1)	0,583	Ввод производится через запятую без пробелов
Ввод F(2),H(2)	100,583	Ввод производится через запятую без пробелов
Ввод F(3),H(3)	50,293	Ввод производится через запятую без пробелов
Ввод F(4),H(4)	50,873	Ввод производится через запятую без пробелов
Ввод F(5),H(5)	50,583	Ввод производится через запятую без пробелов
Ввод F(6),H(6)	50,438	Ввод производится через запятую без пробелов
Ввод F(7),H(7)	50,728	Ввод производится через запятую без пробелов
Ввод F(8),H(8)	25,583	Ввод производится через запятую без пробелов
Ввод F(9),H(9)	75,583	Ввод производится через запятую без пробелов
Ввод F(10),H(10)	0,486	Ввод производится через запятую без пробелов
Ввод F(11),H(11)	100,486	Ввод производится через запятую без пробелов
Ввод F(12),H(12)	0,680	Ввод производится через запятую без пробелов
Ввод F(13),H(13)	100,680	Ввод производится через запятую без пробелов
Ввод F(14),H(14)	33.3,293	Дробные числа записываются через точку
Ввод F(15),H(15)	33.3,486	Дробные числа записываются через точку
Ввод F(16),H(16)	66.7,837	Дробные числа записываются через точку
Ввод I0	14	Вычисление показателей Z(K)5 с использованием циклов и построение графиков
ВВОД I0=61 GOTO 7360	61	
ВВОД ПРИНЯТЫХ ВЕЛИЧИН X,F3,F4,H3,H4	10,10,0,200,80	Для использования циклов и построения графиков. Ввод производится через запятую без пробелов
ВВОД I0=90-ПРОДОЛЖЕНИЕ	90	
IF I0=80 GOTO 9000-ПОСТРОЕНИЕ ГРАФИКА	80	

ВВОД X0,Y0,K0,K3	20,180,50,100	Ввод производится через запятую без пробелов
------------------	---------------	--

¹ Для перехода к следующей команде после каждой последовательности данных обязательно нажимается клавиша «Enter».

После ввода последних значений на экран монитора выводится график зависимости угара металла.

РАСПЕЧАТКА РЕЗУЛЬТАТОВ ВЫПОЛНЕНИЯ ПРОГРАММЫ VN6 МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ И РАСЧЕТОВ ПО МАТЕМАТИЧЕСКОЙ МЕДЕЛИ

Для распечатки полученного графика используется следующий алгоритм:

1. После появления графика на экране монитора надо использовать сочетания клавиш «Alt + PrintScreen». Данное сочетание клавиш заносит полученный график в буфер обмена персонального компьютера. Чтобы выйти из режима построения графика надо нажать клавишу «Esc».

2. Надо открыть любой графический редактор (для примера взята программа, входящая в состав Windows – Paint).

Расположение программы Paint:

- для Windows 95\98\2000 - «Пуск – Программы – Стандартные – Paint» («Start – Programs – Accessories – Paint»)

3. После выбора в меню «Правка» пункт «Вставить» («Edit» - «Paste»), на экране появится построенный график. По умолчанию, график отрисовывается белым цветом на черном фоне. Чтобы получить черный график на белом фоне, необходимо обратить цвета. Пункт «Инвертировать цвета» меню «Изображение» («Image» - «Invert Colors») позволяет обратить цвета графика.

4. Для печати надо выбрать в меню «Файл» пункт «Печать» («File» - «Print»).

Результаты математической модели можно просмотреть и распечатать из файла, автоматически созданного в начале программы, в данном случае NV61.

Windows XP не поддерживает копирование DOS-графики. Поэтому распечатка графика невозможна. Распечатка графика получается при использовании Windows 98.

РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ПРОГРАММЫ VN6

РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ПРОГРАММЫ VN6, РАЗРАБОТАННОЙ А.А. ЧЕРНЫМ
КОЛИЧЕСТВО ОПЫТОВ ПО ПЛАНУ

X= 16

ВЕЛИЧИНЫ ФАКТОРОВ И ПОКАЗАТЕЛЕЙ СТЕПЕНИ

A1= 0 C1= 25 D1= 75

V1= 100 J1= 1 O1= 2 P1= 3

КОЭФФИЦИЕНТЫ ОРТОГОНАЛИЗАЦИИ

V1=-50 U1=-100 Q1= 937.5

I1=-150 M1= 5375 F1=-18750

A2= 293

C2= 438 D2= 728 B2= 873 J2= 1

O2= 2 P2= 3

КОЭФФИЦИЕНТЫ ОРТОГОНАЛИЗАЦИИ

V2=-583 U2=-1166 Q2= 287326.6 I2=-1749.002

M2= 948184.8 F2=-1.564803E+08

ВЕЛИЧИНЫ ПОКАЗАТЕЛЕЙ В СООТВЕТСТВИИ С ПЛАНОМ Y(J)

Y(1)= 7.5

Y(2)= 100

Y(3)= 1.5

Y(4)= 15

Y(5)= 6

Y(6)= 95

Y(7)= 2.5

Y(8)= 52

Y(9)= 20

Y(10)= 17.5

Y(11)= 10

Y(12)= 2.5

Y(13)= 67

Y(14)= 61

Y(15)= 34.5

Y(16)= 9

B(J) ДО АНАЛИЗА

B(1)= 31.3125

B(2)= .6105

B(3)= 3.866667E-03

B(4)= -.0712931

B(5)= -1.365517E-03

B(6)= -1.109788E-04

B(7)= -2.520805E-06

B(8)= -8.000003E-06

B(9)= -1.479717E-08

B(10)= -3.759823E-08

B(11)= -2.735707E-10

B(12)= 6.924674E-11

B(13)= .000002

B(14)= 9.195452E-09

B(15)= -1.479725E-10

B(16)= 7.289374E-13

КОЛИЧЕСТВО ОПЫТОВ НА СРЕДНЕМ УРОВНЕ ФАКТОРОВ

N0= 4

РАСЧЕТНЫЕ ВЕЛИЧИНЫ ПОКАЗАТЕЛЕЙ Z(J) ДО АНАЛИЗА B(J)

Z(1)= 7.500013

Z(2)= 100.0001

Z(3)= 1.499991

Z(4)= 14.99993

Z(5)= 5.999979

Z(6)= 94.9998

Z(7)= 2.500025

Z(8)= 52.00018

$Z(9) = 20.00003$
 $Z(10) = 17.49995$
 $Z(11) = 10.00004$
 $Z(12) = 2.499985$
 $Z(13) = 67.00008$
 $Z(14) = 60.99985$
 $Z(15) = 34.50011$
 $Z(16) = 8.999953$
 $F8=N0-1=3$
 ПРОВЕРКА ПО РАЗНОСТИ $Y(J)-Z(J)$
 В ПРОЦЕНТАХ $(Y(J)-Z(J)) * (100/Y(J))$
 $Y(1)-Z(1) = -1.335144E-05$
 $(Y(1)-Z(1)) * (100 / Y(1)) = -1.780192E-04$
 $Y(2)-Z(2) = -9.155273E-05$
 $(Y(2)-Z(2)) * (100 / Y(2)) = -9.155273E-05$
 $Y(3)-Z(3) = 9.179115E-06$
 $(Y(3)-Z(3)) * (100 / Y(3)) = 6.11941E-04$
 $Y(4)-Z(4) = 7.05719E-05$
 $(Y(4)-Z(4)) * (100 / Y(4)) = 4.704793E-04$
 $Y(5)-Z(5) = 2.145767E-05$
 $(Y(5)-Z(5)) * (100 / Y(5)) = 3.576279E-04$
 $Y(6)-Z(6) = 1.983643E-04$
 $(Y(6)-Z(6)) * (100 / Y(6)) = 2.088045E-04$
 $Y(7)-Z(7) = -2.479553E-05$
 $(Y(7)-Z(7)) * (100 / Y(7)) = -9.918213E-04$
 $Y(8)-Z(8) = -1.792908E-04$
 $(Y(8)-Z(8)) * (100 / Y(8)) = -3.447899E-04$
 $Y(9)-Z(9) = -2.861023E-05$
 $(Y(9)-Z(9)) * (100 / Y(9)) = -1.430511E-04$
 $Y(10)-Z(10) = 4.959106E-05$
 $(Y(10)-Z(10)) * (100 / Y(10)) = 2.833775E-04$
 $Y(11)-Z(11) = -3.528595E-05$
 $(Y(11)-Z(11)) * (100 / Y(11)) = -3.528595E-04$
 $Y(12)-Z(12) = 1.478195E-05$
 $(Y(12)-Z(12)) * (100 / Y(12)) = 5.912781E-04$
 $Y(13)-Z(13) = -8.392334E-05$
 $(Y(13)-Z(13)) * (100 / Y(13)) = -1.252587E-04$
 $Y(14)-Z(14) = 1.487732E-04$
 $(Y(14)-Z(14)) * (100 / Y(14)) = 2.438905E-04$
 $Y(15)-Z(15) = -1.106262E-04$
 $(Y(15)-Z(15)) * (100 / Y(15)) = -3.206557E-04$
 $Y(16)-Z(16) = 4.673004E-05$
 $(Y(16)-Z(16)) * (100 / Y(16)) = 5.192227E-04$
 ДИСПЕРСИЯ ОПЫТОВ $U9 = .1667$
 РАСЧЕТНЫЕ ВЕЛИЧИНЫ $T(J)$
 $T(1) = 306.7679$
 $T(2) = 236.422$
 $T(3) = 35.51405$
 $T(4) = 160.1318$
 $T(5) = 121.2376$
 $T(6) = 34.28936$
 $T(7) = 30.7871$
 $T(8) = 16.8458$
 $T(9) = 4.286171$
 $T(10) = 2.130663$
 $T(11) = .6128117$
 $T(12) = 3.6789$
 $T(13) = .5808894$
 $T(14) = .6123145$
 $T(15) = 1.355414$
 $T(16) = 1.224641$
 ТАБЛИЧНЫЙ Т-КРИТЕРИЙ $T0 = 3.182$
 В(J) ПОСЛЕ АНАЛИЗА

B(1)= 31.3125
B(2)= .6105
B(3)= 3.866667E-03
B(4)=-.0712931
B(5)=-1.365517E-03
B(6)=-1.109788E-04
B(7)=-2.520805E-06
B(8)=-8.000003E-06
B(9)=-1.479717E-08
B(10)= 0
B(11)= 0
B(12)= 6.924674E-11
B(13)= 0
B(14)= 0
B(15)= 0
B(16)= 0

КОЛИЧЕСТВО СТАТИСТИЧЕСКИ ЗНАЧИМЫХ
КОЭФИЦИЕНТОВ РЕГРЕССИИ K9= 10

F9=X-1

F9= 15

ТАБЛИЧНЫЙ F-КРИТЕРИЙ F7= 8.69

РАСЧЕТНЫЕ ВЕЛИЧИНЫ ПОКАЗАТЕЛЯ Z(J) ПОСЛЕ АНАЛИЗА B(J)

Z(1)= 7.2625

Z(2)= 99.9625

Z(3)= 1.637498

Z(4)= 15.13752

Z(5)= 6.374995

Z(6)= 95.17497

Z(7)= 2.375014

Z(8)= 51.57501

Z(9)= 20.1875

Z(10)= 17.325

Z(11)= 9.674984

Z(12)= 2.51251

Z(13)= 66.53751

Z(14)= 61.72499

Z(15)= 34.27498

Z(16)= 9.262522

ПРОВЕРКА ПО РАЗНОСТИ Y(J)-Z(J)

В ПРОЦЕНТАХ (Y(J)-Z(J)) * (100/Y(J))

Y(1)-Z(1)= .2374997

(Y(1)-Z(1)) * (100 / Y(1)) = 3.166663

Y(2)-Z(2)= 3.749847E-02

(Y(2)-Z(2)) * (100 / Y(2)) = 3.749847E-02

Y(3)-Z(3)=-.1374978

(Y(3)-Z(3)) * (100 / Y(3)) = -9.166519

Y(4)-Z(4)=-.137517

(Y(4)-Z(4)) * (100 / Y(4)) = -.9167798

Y(5)-Z(5)=-.3749948

(Y(5)-Z(5)) * (100 / Y(5)) = -6.249913

Y(6)-Z(6)=-.1749725

(Y(6)-Z(6)) * (100 / Y(6)) = -.1841816

Y(7)-Z(7)= .1249857

(Y(7)-Z(7)) * (100 / Y(7)) = 4.999428

Y(8)-Z(8)= .4249916

(Y(8)-Z(8)) * (100 / Y(8)) = .8172916

Y(9)-Z(9)=-.1875038

(Y(9)-Z(9)) * (100 / Y(9)) = -.9375191

Y(10)-Z(10)= .1749992

(Y(10)-Z(10)) * (100 / Y(10)) = .9999956

Y(11)-Z(11)= .325016

(Y(11)-Z(11)) * (100 / Y(11)) = 3.25016

Y(12)-Z(12)=-.0125103

$(Y(12)-Z(12)) * (100 / Y(12)) = -.500412$
 $Y(13)-Z(13) = .4624939$
 $(Y(13)-Z(13)) * (100 / Y(13)) = .6902894$
 $Y(14)-Z(14) = -.7249947$
 $(Y(14)-Z(14)) * (100 / Y(14)) = -1.188516$
 $Y(15)-Z(15) = .2250175$
 $(Y(15)-Z(15)) * (100 / Y(15)) = .6522248$
 $Y(16)-Z(16) = -.2625217$
 $(Y(16)-Z(16)) * (100 / Y(16)) = -2.916908$

РАСЧЕТНАЯ ВЕЛИЧИНА F-КРИТЕРИЯ F6= .5973794

АДЕКВАТНО, ТАК КАК F6 <= F7

СИСТЕМНОЕ ПРЕДСТАВЛЕНИЕ

ДЛЯ АНАЛИЗОВ РЕЗУЛЬТАТОВ РАСЧЁТОВ

ПО МАТЕМАТИЧЕСКОЙ МОДЕЛИ

ОТНОСИТЕЛЬНАЯ ВЕЛИЧИНА ПОКАЗАТЕЛЯ

$Z1(J) = Z(J) / (S/X)$, ГДЕ S - СУММА

АБСОЛЮТНЫХ ВЕЛИЧИН ПОКАЗАТЕЛЕЙ, S = 501

S/X - СРЕДНЯЯ АРИФМЕТИЧЕСКАЯ ВЕЛИЧИНА, S/X = 31.3125

1.1. A2= 293 A1= 0 Z(1)= 7.2625 Z1(1)= .2319361
 1.2. A2= 293 C1= 25 Z(9)= 20.1875 Z1(9)= .6447107
 1.3. A2= 293 D1= 75 Z(13)= 66.53751 Z1(13)= 2.12495
 1.4. A2= 293 B1= 100 Z(2)= 99.9625 Z1(2)= 3.192415
 2.1. C2= 438 A1= 0 Z(5)= 6.374995 Z1(5)= .2035926
 2.2. C2= 438 C1= 25 Z(10)= 17.325 Z1(10)= .5532934
 2.3. C2= 438 D1= 75 Z(14)= 61.72499 Z1(14)= 1.971257
 2.4. C2= 438 B1= 100 Z(6)= 95.17497 Z1(6)= 3.03952
 3.1. D2= 728 A1= 0 Z(7)= 2.375014 Z1(7)= 7.584876E-02
 3.2. D2= 728 C1= 25 Z(11)= 9.674984 Z1(11)= .3089815
 3.3. D2= 728 D1= 75 Z(15)= 34.27498 Z1(15)= 1.09461
 3.4. D2= 728 B1= 100 Z(8)= 51.57501 Z1(8)= 1.647106
 4.1. B2= 873 A1= 0 Z(3)= 1.637498 Z1(3)= 5.229533E-02
 4.2. B2= 873 C1= 25 Z(12)= 2.51251 Z1(12)= 8.023985E-02
 4.3. B2= 873 D1= 75 Z(16)= 9.262522 Z1(16)= .2958091
 4.4. B2= 873 B1= 100 Z(4)= 15.13752 Z1(4)= .4834336
 5.1. A1= 0 A2= 293 Z(1)= 7.2625 Z1(1)= .2319361
 5.2. A1= 0 C2= 438 Z(5)= 6.374995 Z1(5)= .2035926
 5.3. A1= 0 D2= 728 Z(7)= 2.375014 Z1(7)= 7.584876E-02
 5.4. A1= 0 B2= 873 Z(3)= 1.637498 Z1(3)= 5.229533E-02
 6.1. C1= 25 A2= 293 Z(9)= 20.1875 Z1(9)= .6447107
 6.2. C1= 25 C2= 438 Z(10)= 17.325 Z1(10)= .5532934
 6.3. C1= 25 D2= 728 Z(11)= 9.674984 Z1(11)= .3089815
 6.4. C1= 25 B2= 873 Z(12)= 2.51251 Z1(12)= 8.023985E-02
 7.1. D1= 75 A2= 293 Z(13)= 66.53751 Z1(13)= 2.12495
 7.2. D1= 75 C2= 438 Z(14)= 61.72499 Z1(14)= 1.971257
 7.3. D1= 75 D2= 728 Z(15)= 34.27498 Z1(15)= 1.09461
 7.4. D1= 75 B2= 873 Z(16)= 9.262522 Z1(16)= .2958091
 8.1. B1= 100 A2= 293 Z(2)= 99.9625 Z1(2)= 3.192415
 8.2. B1= 100 C2= 438 Z(6)= 95.17497 Z1(6)= 3.03952
 8.3. B1= 100 D2= 728 Z(8)= 51.57501 Z1(8)= 1.647106
 8.4. B1= 100 B2= 873 Z(4)= 15.13752 Z1(4)= .4834336

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ

$Z(J) = 31.3125 + .6105 * I(J) + 3.866667E-03 * K(J) +$
 $-.0712931 * P(J) + -1.365517E-03 * I(J) * P(J) +$
 $+ -1.109788E-04 * Q(J) + -2.520805E-06 * I(J) * Q(J) +$
 $+ -8.000003E-06 * P(J) * K(J) + -1.479717E-08 * K(J) * Q(J) +$
 $+ 0 * U(J) + 0 * I(J) * U(J) +$
 $+ 6.924674E-11 * K(J) * U(J) + 0 * L(J) +$
 $+ 0 * P(J) * L(J) + 0 * Q(J) * L(J) +$
 $+ 0 * L(J) * U(J),$

ГДЕ

$I(J) = F(J)^1 + -50$;

$K(J) = F(J)^2 + -100 * F(J)^1 + 937.5$

ОБОЗНАЧЕНИЕ: F(J)- 1-й ФАКТОР

$L(J)=F(J)^3 -150 *F(J)^2 +$
 $+ 5375 F(J) -18750$
 ОБОЗНАЧЕНИЕ: F(J)- 1-й ФАКТОР
 $P(J)=H(J) -583 ;$
 $Q(J)=H(J)^2 -1166 *H(J) + 287326.6 ;$
 $U(J)=H(J)^3 -1749.002 *H(J)^2 +$
 $+ 948184.8 *H(J) -1.564803E+08$
 ОБОЗНАЧЕНИЕ: H(J)- 2-й ФАКТОР
 РАСЧЕТЫ ПО МОДЕЛИ
 ФАКТОРЫ F (1)= 0 H (1)= 583
 $Z (1)= 4.350007$
 ФАКТОРЫ F (2)= 100 H (2)= 583
 $Z (2)= 78.64999$
 ФАКТОРЫ F (3)= 50 H (3)= 293
 $Z (3)= 39.94584$
 ФАКТОРЫ F (4)= 50 H (4)= 873
 $Z (4)= 5.054185$
 ФАКТОРЫ F (5)= 50 H (5)= 583
 $Z (5)= 29.88887$
 ФАКТОРЫ F (6)= 50 H (6)= 438
 $Z (6)= 35.775$
 ФАКТОРЫ F (7)= 50 H (7)= 728
 $Z (7)= 20.30831$
 ФАКТОРЫ F (8)= 25 H (8)= 583
 $Z (8)= 14.21665$
 ФАКТОРЫ F (9)= 75 H (9)= 583
 $Z (9)= 51.36664$
 ФАКТОРЫ F (10)= 0 H (10)= 486
 $Z (10)= 5.772266$
 ФАКТОРЫ F (11)= 100 H (11)= 486
 $Z (11)= 90.94593$
 ФАКТОРЫ F (12)= 0 H (12)= 680
 $Z (12)= 2.950122$
 ФАКТОРЫ F (13)= 100 H (13)= 680
 $Z (13)= 61.63277$
 ФАКТОРЫ F (14)= 33.3 H (14)= 293
 $Z (14)= 25.98954$
 ФАКТОРЫ F (15)= 33.3 H (15)= 486
 $Z (15)= 21.47881$
 ФАКТОРЫ F (16)= 66.7 H (16)= 837
 $Z (16)= 13.80674$
 ВЫЧИСЛЕНИЯ ПОКАЗАТЕЛЕЙ Z(K5)
 ПО МАТЕМАТИЧЕСКОЙ МОДЕЛИ
 С ИСПОЛЬЗОВАНИЕМ ЦИКЛОВ
 И ПОСТРОЕНИЕ ГРАФИКОВ
 ФАКТОР F(1)=F3+F4
 F4-ШАГ ПРИРАЩЕНИЯ 1-ГО ФАКТОРА
 ФАКТОР H(1)=H3+H4
 H4-ШАГ ПРИРАЩЕНИЯ 2-ГО ФАКТОРА
 X-КОЛИЧЕСТВО ЗНАЧЕНИЙ 1, 2-ГО ФАКТОРОВ
 $X= 10 F3= 10 F4= 0$
 $H3= 200 H4= 80$
 $F (1)= 10$
 $H (1)= 280$
 $Z (1)= 11.74416$
 $F (2)= 10$
 $H (2)= 360$
 $Z (2)= 10.86527$
 $F (3)= 10$
 $H (3)= 440$
 $Z (3)= 9.827169$
 $F (4)= 10$
 $H (4)= 520$

$Z(4) = 8.637831$
 $F(5) = 10$
 $H(5) = 600$
 $Z(5) = 7.305232$
 $F(6) = 10$
 $H(6) = 680$
 $Z(6) = 5.837348$
 $F(7) = 10$
 $H(7) = 760$
 $Z(7) = 4.242157$
 $F(8) = 10$
 $H(8) = 840$
 $Z(8) = 2.527637$
 $F(9) = 10$
 $H(9) = 920$
 $Z(9) = .7017652$
 $F(10) = 10$
 $H(10) = 1000$
 $Z(10) = -1.227484$
ВЫЯВЛЕНИЕ MAX Z(K5) И MIN Z(K5)
 $MAX Z(K5) = 11.74416$
 $MAX Z(1) = 11.74416$
 $MIN Z(K5) = -1.227484$
 $MIN Z(10) = -1.227484$
 $MIN Z(K5) = K7, MAX Z(K5) = K8$
 $K6(K5) = (Z(K5) + ABS(K7)) / (ABS(K7) + ABS(K8))$
 $K6(1) = 1$
 $K6(2) = .9322456$
 $K6(3) = .8522171$
 $K6(4) = .7605295$
 $K6(5) = .6577978$
 $K6(6) = .5446367$
 $K6(7) = .4216615$
 $K6(8) = .289487$
 $K6(9) = .1487282$
 $K6(10) = 0$
 $J5 = ABS(K7) / (ABS(K7) + ABS(K8))$
 $J5 = 9.462826E-02$
ПОСТРОЕНИЕ ГРАФИКА
ЗАВИСИМОСТЬ K6(K5) ОТ ФАКТОРА
K6(K5)-ОТНОСИТЕЛЬНАЯ ВЕЛИЧИНА ПОКАЗАТЕЛЯ
K5-НОМЕР ВЕЛИЧИНЫ ФАКТОРА И ПОКАЗАТЕЛЯ
ВЕЛИЧИНЫ ФАКТОРОВ ЗАДАНЫ
 $X0 = 20$ $Y0 = 180$ $K0 = 50$ $K3 = 100$, ГДЕ
 $X0$ -ОТСТУП ВПРАВО ПО ОСИ X
 $Y0$ -ОТСТУП ВНИЗ ПО ОСИ Y
 $K0$ -ДЛИНА ГРАФИКА ПО ОСИ X
 $K3$ -ВЫСОТА ГРАФИКА ПО ОСИ Y

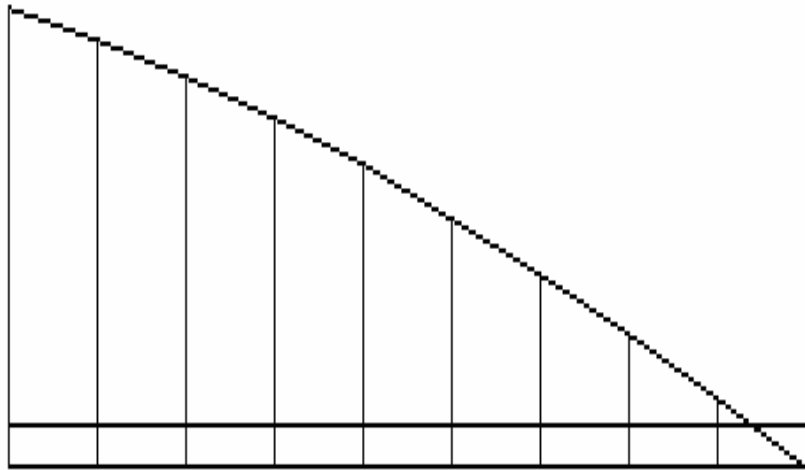


Рис. 2. Угар металла по результатам расчета

АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ РАСЧЕТОВ

В результате выполнения программы был получен график выявления зависимости угара металла $Y_{мет}$ от $Ш_c$ и T_B . Анализируя данные можно сделать вывод, что чем больше стали в шихте, тем больше угар металла и чем выше температура воздуха-окислителя, тем меньше угар. Следовательно, использование большого количества стали в шихте газовых вагранок нецелесообразно. Рационально применение подогрева подаваемого в газовые горелки воздуха при оптимальных величинах коэффициента расхода воздуха.

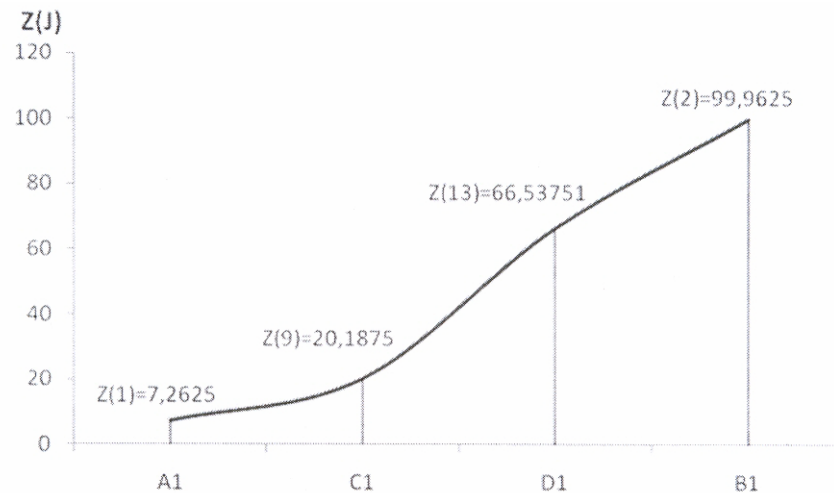


Рис. 3. $Z(J)$ при $A2 = 293$

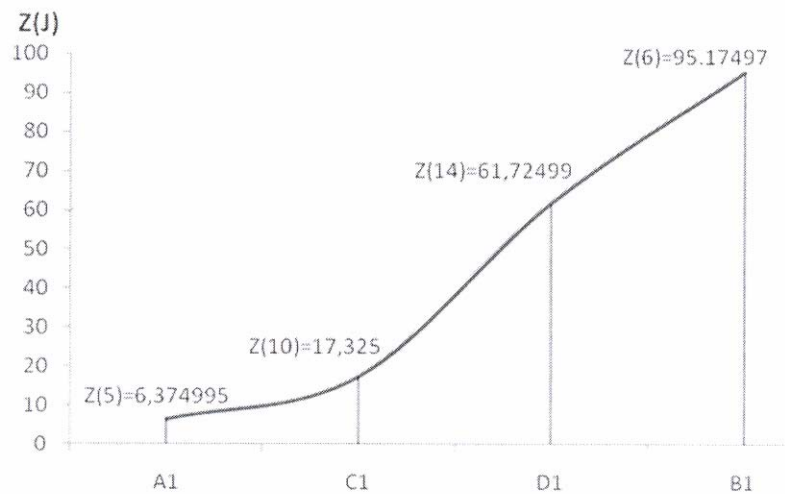


Рис. 4. $Z(J)$ при $C2 = 438$

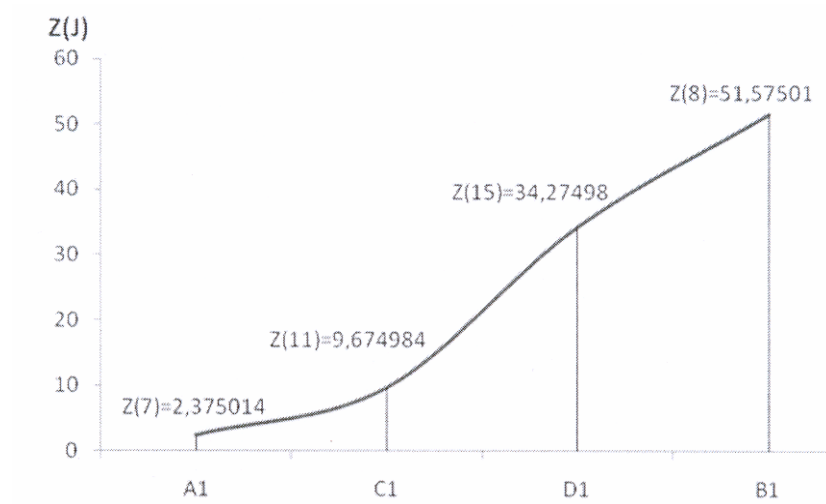


Рис. 5. $Z(J)$ при $D2 = 728$

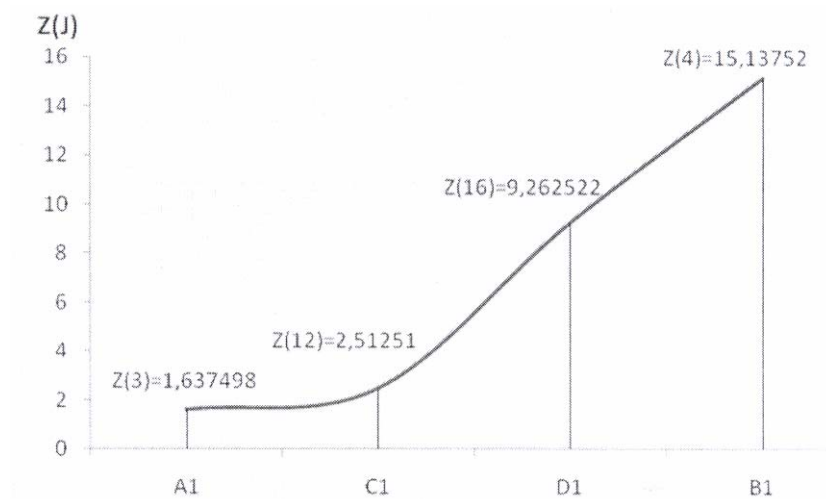


Рис. 6. $Z(J)$ при $B2 = 873$

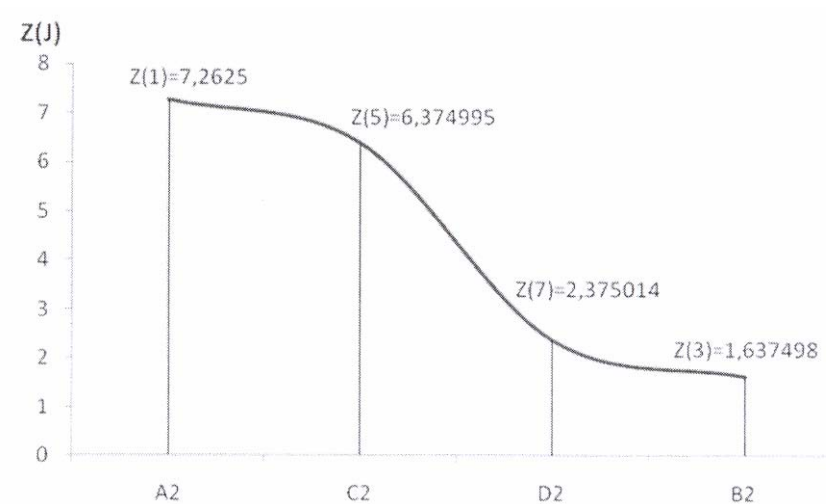


Рис. 7. $Z(J)$ при $A1 = 0$

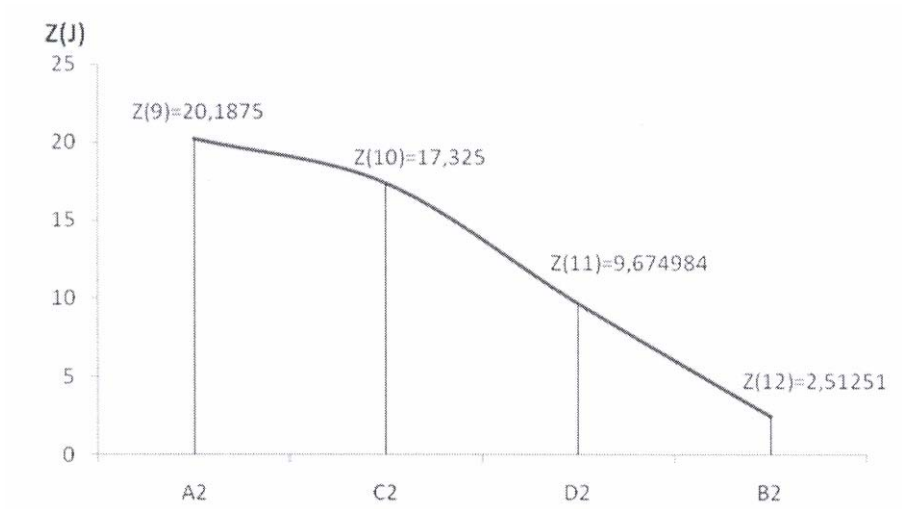


Рис. 8. $Z(J)$ при $C2 = 25$

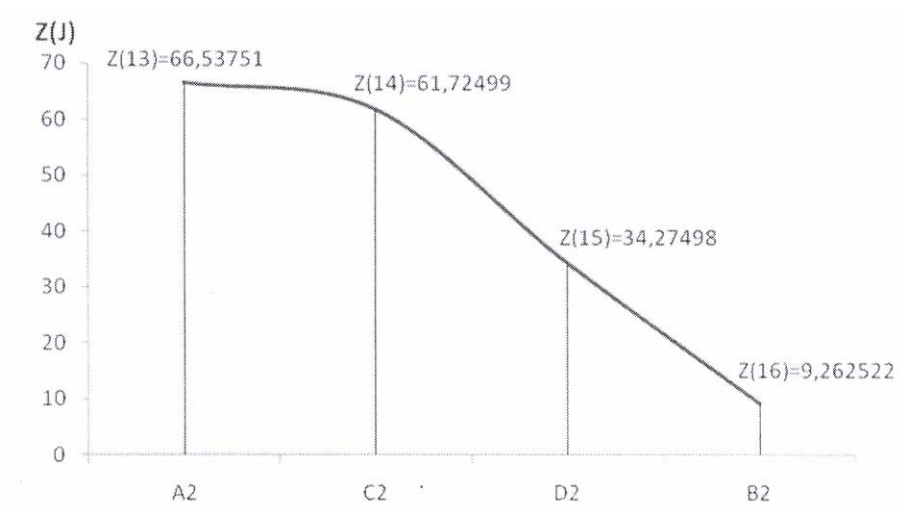


Рис. 9. $Z(J)$ при $D1 = 75$

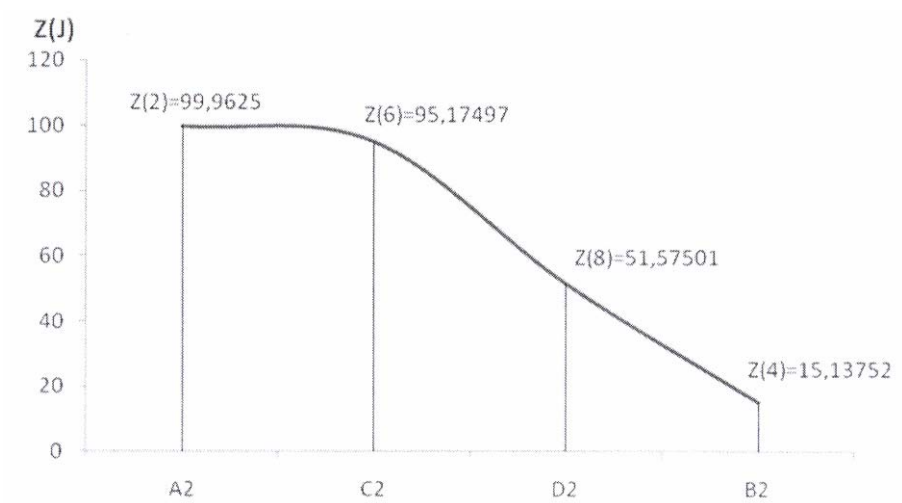


Рис. 10. $Z(J)$ при $B1 = 100$

ПОСТРОЕНИЕ ГРАФИКОВ

Графики удобно строить в приложении MS Office Excel. В данном примере использовался MS Office 2007. Для получения диаграммы создаем новый документ Excel. Данные располагаем в виде таблицы:

A1	7,2652
C1	20,1875
D1	66,53751
B1	99,9625

В левую колонку этой таблицы вводим значения, которые будут отображаться на горизонтальной оси графика. В правой колонке вводим числовые значения $Z(J)$. Они будут отображаться по вертикальной оси. Далее выделяем полученную табличку, жмем по вкладке "Вставка" и в разделе "Диаграммы" выбираем тип диаграммы. Для наглядности лучше подойдет тип "График". Можно выбрать как обычный "график" так и "график с маркерами". В данном случае выбран "график". В дальнейшем тип диаграммы можно изменить.

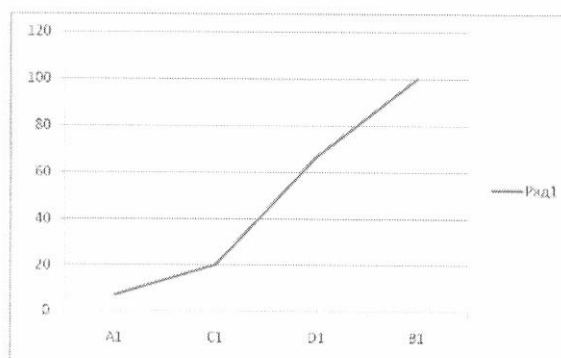


Рис. 11.

Для того, чтобы сделать линию графика сглаженной, нужно щелкнуть правой кнопкой мыши непосредственно на линию и выбрать пункт «Формат ряда данных». В открывшемся диалоговом окне выбрать пункт «Тип линии» и поставить галочку «Сглаженная линия», далее нажать кнопку закрыть.

Чтобы добавить подписи данных на линию графика нужно снова щелкнуть правой кнопкой мыши на линию и выбрать пункт «Добавить подписи данных».

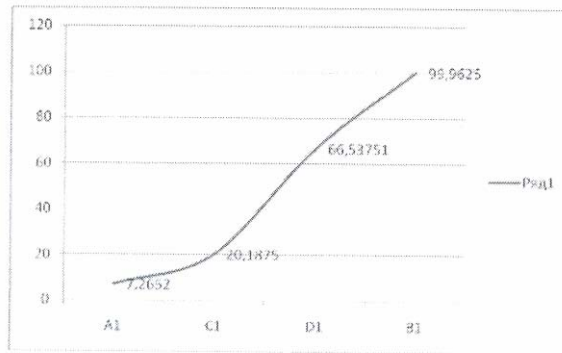


Рис. 12.

Изменить расположение области с данными можно двумя способами.

1. Щелкнуть правой кнопкой мыши на линии графика и выбрать пункт "Форма подписи данных" и во вкладке "Параметры подписи" выбрать положение подписи: справа, слева, в центре и др.

2. Если текст области с данными накладывается на линию графика, то можно переместить их вручную, для чего нужно нажать на ней левой кнопкой мыши и удерживая ее переместить в нужное место.

Подпись с данными можно редактировать, щелкнув на выделенной области левой кнопкой мыши. Так можно получить на графике подпись типа "Z(9)=20,1875".

В области построения диаграммы можно убрать или добавить сетку. В данном случае уже имеется горизонтальная сетка. Чтобы ее убрать, нужно в разделе "Работа с диаграммами" выбрать вкладку "Макет", нажать на панели кнопку "Сетка": Сетка/горизонтальные линии сетки по основной оси/Не показывать горизонтальные линии сетки.

Элемент легенды при желании можно удалить. Для этого нужно щелкнуть на нем правой кнопкой мыши и выбрать пункт "удалить".

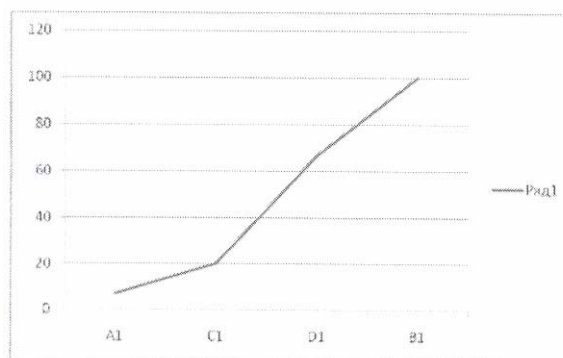


Рис. 13.

Чтобы дать название осям, нужно во вкладке "Макет" щелкнуть на кнопку "название осей" и выбрать ось, которой хотим присвоить название (горизонтальную или вертикальную).

Для наглядности на диаграмму можно добавить линии проекции. Для этого во вкладке "Макет» нужно нажать кнопку «анализ», выбрать пункт «линии» и далее «линии проекции».

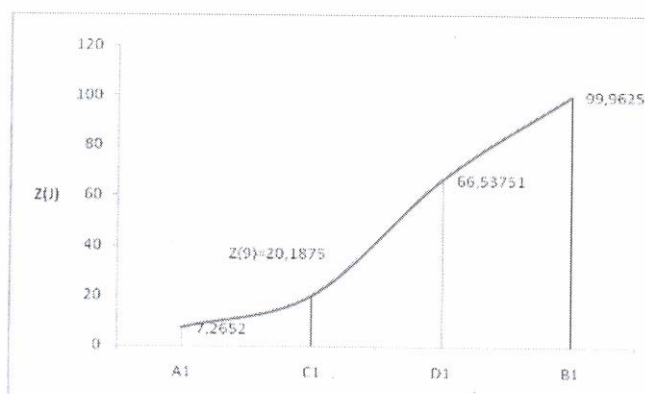


Рис. 14.

Создание диаграмм в Excel может осуществляться автоматически. Для этого надо выделить данные и нажать на клавишу F11. Диаграмма будет создана на отдельном листе (тип диаграммы определяется по умолчанию – обычно это плоская гистограмма).

При представлении данных научного характера используется точечная диаграмма. Данные X следует располагать в одной строке или столбце, а соответствующие значения Y – в смежных строках или столбцах. Можно отображать зависимости данных от величины, меняющейся с переменным шагом. Диаграмма может быть как в виде отдельных точек, так и в виде сглаженных линий. Отдельные точки исходных данных на диаграмме можно помечать маркерами.

При использовании Excel рационально вызывать Мастера диаграмм и выполнять шаги, предусмотренные этой программой с учетом требуемых графических построений [17].

РЕКОМЕНДАЦИИ ПО УЛУЧШЕНИЮ ПРОЦЕССА ПЛАВКИ В ГАЗОВОЙ ВАГРАНКЕ ПО РЕЗУЛЬТАТАМ МОДЕЛИРОВАНИЯ

На основании анализа полученной математической модели (зависимости $Y_{\text{мет}}$ от $Ш_c$ и T_b , связанной с α) сделаны выводы о нецелесообразности использования большого количества стали в шихте газовых вагранок, о рациональности применения подогрева подаваемого в газовые горелки воздуха, снижении величины коэффициента расхода воздуха по мере увеличения температуры воздуха, о необходимости добавки в зону перегрева металла раскислителей, в частности углерода, образующегося при разложении углеродов или вдуваемого в виде порошка, а также загружаемого в виде кусков (электродного боя).

Комплексный подход к моделированию сложных процессов позволяет учитывать влияние на показатели процесса многих факторов путем выбора существенных факторов, связанных с другими факторами, изменение которых происходит в зависимости от обобщенных факторов. Так, зависимость $\alpha = f(T_b)$ позволила учитывать изменение окислительных свойств печной атмосферы, изменение состава и температуры продуктов сгорания в плавильном агрегате. В конечном итоге сложный процесс был сведен к менее сложному, что позволило применить математическое моделирование при ортогональном планировании двухфакторных экспериментов.

Рекомендации по улучшению процесса плавки в газовой вагранке сводятся к следующему.

1. Эффективность процесса плавки металла в газовой вагранке может быть высокой при подаче в горелки горячего воздуха и снижении коэффициента расхода воздуха.
2. При количестве стали в составе шихты 10 % потери металла от окисления можно уменьшить до 0 при T_b 920 K (647°C).
3. Для увеличения количества стали в составе шихты (больше 10 %) необходимо вводить в продукты сгорания раскислители в виде водорода и сажи-стого углерода или плавку и перегрев металла производить на углеродосодержащей огнеупорной колоше, причем с повышением T_b и увеличением количества боя графитовых электродов в огнеупорной колоше угар металла при плавке в газовой вагранке будет снижаться.
4. Рационально чугунную шихту плавить в газовой вагранке при минимальном расходе газообразного топлива, максимальной производительности плавильного агрегата, при малых потерях металла в связи с окислением, а затем расплавленный чугун надо переливать в электропечь, добавлять в жидкий чугун стальную шихту, повышать температуру металла и доводить состав расплава до требуемого, после чего использовать жидкий металл для получения отливок.

ПРОГРАММА W4 на языке Турбо Паскаль
(три модуля tpg3, X=3, X=9, X=27, _X=4_, X=5, X=12, X=15, _X=16_,
X=20, X=25, система после «адекватно»)

```

program tpg3_1;
{Математическое моделирование}
uses tpg3_3, tpg3_2;
label 1,2,3,4,5;
procedure VV_DAN;
begin
  case X of
    4: begin
      VVOD41;
      GB3710(A1,B1,C1,D1,J1,O1,P1,V1,U1,Q1,I1,M1,F1);
      writeln('Коэффициенты ортогонализации');
      writeln(F0,'Коэффициенты ортогонализации');
      writeln('V1=',V1,' U1=',U1,' Q1=',Q1);
      writeln('I1=',I1,' M1=',M1,' F1=',F1);
      writeln(F0,'V1=',V1,' U1=',U1,' Q1=',Q1);
      writeln(F0,'I1=',I1,' M1=',M1,' F1=',F1);
    end;
    16:begin
      VVOD41;
      GB3710(A1,B1,C1,D1,J1,O1,P1,V1,U1,Q1,I1,M1,F1);
      writeln('Коэффициенты ортогонализации');
      writeln(F0,'Коэффициенты ортогонализации');
      writeln('V1=',V1,' U1=',U1,' Q1=',Q1);
      writeln('I1=',I1,' M1=',M1,' F1=',F1);
      writeln(F0,'V1=',V1,' U1=',U1,' Q1=',Q1);
      writeln(F0,'I1=',I1,' M1=',M1,' F1=',F1);
      VVOD42;
      GB3710(A2,B2,C2,D2,J2,O2,P2,V2,U2,Q2,I2,M2,F2);
      writeln('Коэффициенты ортогонализации');
      writeln(F0,'Коэффициенты ортогонализации');
      writeln('V2=',V2,' U2=',U2,' Q2=',Q2);
      writeln('I2=',I2,' M2=',M2,' F2=',F2);
      writeln(F0,'V2=',V2,' U2=',U2,' Q2=',Q2);
      writeln(F0,'I2=',I2,' M2=',M2,' F2=',F2);
    end;
    20:begin
      VVOD41;
      GB3710(A1,B1,C1,D1,J1,O1,P1,V1,U1,Q1,I1,M1,F1);
      writeln('Коэффициенты ортогонализации');
      writeln(F0,'Коэффициенты ортогонализации');
      writeln('V1=',V1,' U1=',U1,' Q1=',Q1);
      writeln('I1=',I1,' M1=',M1,' F1=',F1);
      writeln(F0,'V1=',V1,' U1=',U1,' Q1=',Q1);
      writeln(F0,'I1=',I1,' M1=',M1,' F1=',F1);
      VVOD52;
      GB3860(A2,C2,E2,D2,B2,J2,O2,P2,T2,V2,U2,
      Q2,I2,M2,F2,G2,H2,K2,L2);
      writeln('Коэффициенты ортогонализации');
      writeln(F0,'Коэффициенты ортогонализации');
      writeln('V2=',V2,' U2=',U2,' Q2=',Q2);
      writeln('I2=',I2,' M2=',M2,' F2=',F2);
      writeln('G2=',G2,' H2=',H2,' K2=',K2);
      writeln('L2=',L2);
      writeln(F0,'V2=',V2,' U2=',U2,' Q2=',Q2);
      writeln(F0,'I2=',I2,' M2=',M2,' F2=',F2);
      writeln(F0,'G2=',G2,' H2=',H2,' K2=',K2);
    end;
  end;
end;

```

```

    writeln(F0,L2=',L2);
end;
5: begin
  VVOD51;
  GB3860(A1,B1,C1,D1,E1,J1,O1,P1,T1,V1,U1,
  Q1,I1,M1,F1,G1,H1,K1,L1);
  writeln('Коэффициенты ортогонализации');
  writeln(F0,'Коэффициенты ортогонализации');
  writeln('V1=',V1,' U1=',U1,' Q1=',Q1);
  writeln('I1=',I1,' M1=',M1,' F1=',F1);
  writeln('G1=',G1,' H1=',H1,' K1=',K1);
  writeln('L1=',L1);
  writeln(F0,'V1=',V1,' U1=',U1,' Q1=',Q1);
  writeln(F0,'I1=',I1,' M1=',M1,' F1=',F1);
  writeln(F0,'G1=',G1,' H1=',H1,' K1=',K1);
  writeln(F0,'L1=',L1);
end;
25:begin
  VVOD51;
  GB3860(A1,B1,C1,D1,E1,J1,O1,P1,T1,V1,U1,
  Q1,I1,M1,F1,G1,H1,K1,L1);
  writeln('Коэффициенты ортогонализации');
  writeln(F0,'Коэффициенты ортогонализации');
  writeln('V1=',V1,' U1=',U1,' Q1=',Q1);
  writeln('I1=',I1,' M1=',M1,' F1=',F1);
  writeln('G1=',G1,' H1=',H1,' K1=',K1);
  writeln('L1=',L1);
  writeln(F0,'V1=',V1,' U1=',U1,' Q1=',Q1);
  writeln(F0,'I1=',I1,' M1=',M1,' F1=',F1);
  writeln(F0,'G1=',G1,' H1=',H1,' K1=',K1);
  writeln(F0,'L1=',L1);
  VVOD52;
  GB3860(A2,C2,E2,D2,B2,J2,O2,P2,T2,V2,U2,
  Q2,I2,M2,F2,G2,H2,K2,L2);
  writeln('Коэффициенты ортогонализации');
  writeln(F0,'Коэффициенты ортогонализации');
  writeln('V2=',V2,' U2=',U2,' Q2=',Q2);
  writeln('I2=',I2,' M2=',M2,' F2=',F2);
  writeln('G2=',G2,' H2=',H2,' K2=',K2);
  writeln('L2=',L2);
  writeln(F0,'V2=',V2,' U2=',U2,' Q2=',Q2);
  writeln(F0,'I2=',I2,' M2=',M2,' F2=',F2);
  writeln(F0,'G2=',G2,' H2=',H2,' K2=',K2);
  writeln(F0,'L2=',L2);
end;
15:begin
  VVOD31;
  GB3660(A1,E1,B1,J1,O1,V1,U1,Q1);
  writeln('Коэффициенты ортогонализации');
  writeln(F0,'Коэффициенты ортогонализации');
  writeln('V1=',V1,' U1=',U1,' Q1=',Q1);
  writeln(F0,'V1=',V1,' U1=',U1,' Q1=',Q1);
  VVOD52;
  GB3860(A2,C2,E2,D2,B2,J2,O2,P2,T2,V2,U2,
  Q2,I2,M2,F2,G2,H2,K2,L2);
  writeln('Коэффициенты ортогонализации');
  writeln(F0,'Коэффициенты ортогонализации');
  writeln('V2=',V2,' U2=',U2,' Q2=',Q2);
  writeln('I2=',I2,' M2=',M2,' F2=',F2);
  writeln('G2=',G2,' H2=',H2,' K2=',K2);
  writeln('L2=',L2);
  writeln(F0,'V2=',V2,' U2=',U2,' Q2=',Q2);
  writeln(F0,'I2=',I2,' M2=',M2,' F2=',F2);

```

```

writeln(F0,'G2=',G2,' H2=',H2,' K2=',K2);
writeln(F0,'L2=',L2);
end;
3: begin
  VVOD31;
  GB3660(A1,E1,B1,J1,O1,V1,U1,Q1);
  writeln('Коэффициенты ортогонализации');
  writeln(F0,'Коэффициенты ортогонализации');
  writeln('V1=',V1,' U1=',U1,' Q1=',Q1);
  writeln(F0,'V1=',V1,' U1=',U1,' Q1=',Q1);
end;
12:begin
  VVOD31;
  GB3660(A1,E1,B1,J1,O1,V1,U1,Q1);
  writeln('Коэффициенты ортогонализации');
  writeln(F0,'Коэффициенты ортогонализации');
  writeln('V1=',V1,' U1=',U1,' Q1=',Q1);
  writeln(F0,'V1=',V1,' U1=',U1,' Q1=',Q1);
  VVOD42;
  GB3710(A2,C2,D2,B2,J2,O2,P2,V2,U2,Q2,I2,M2,F2);
  writeln('Коэффициенты ортогонализации');
  writeln(F0,'Коэффициенты ортогонализации');
  writeln('V2=',V2,' U2=',U2,' Q2=',Q2);
  writeln('I2=',I2,' M2=',M2,' F2=',F2);
  writeln(F0,'V2=',V2,' U2=',U2,' Q2=',Q2);
  writeln(F0,'I2=',I2,' M2=',M2,' F2=',F2);
end;
9: begin
  VVOD31;
  GB3660(A1,E1,B1,J1,O1,V1,U1,Q1);
  writeln('Коэффициенты ортогонализации');
  writeln(F0,'Коэффициенты ортогонализации');
  writeln('V1=',V1,' U1=',U1,' Q1=',Q1);
  writeln(F0,'V1=',V1,' U1=',U1,' Q1=',Q1);
  VVOD32;
  GB3660(A2,E2,B2,J2,O2,V2,U2,Q2);
  writeln('Коэффициенты ортогонализации');
  writeln(F0,'Коэффициенты ортогонализации');
  writeln('V2=',V2,' U2=',U2,' Q2=',Q2);
  writeln(F0,'V2=',V2,' U2=',U2,' Q2=',Q2);
end;
27:begin
  VVOD31;
  GB3660(A1,E1,B1,J1,O1,V1,U1,Q1);
  writeln('Коэффициенты ортогонализации');
  writeln(F0,'Коэффициенты ортогонализации');
  writeln('V1=',V1,' U1=',U1,' Q1=',Q1);
  writeln(F0,'V1=',V1,' U1=',U1,' Q1=',Q1);
  VVOD32;
  GB3660(A2,E2,B2,J2,O2,V2,U2,Q2);
  writeln('Коэффициенты ортогонализации');
  writeln(F0,'Коэффициенты ортогонализации');
  writeln('V2=',V2,' U2=',U2,' Q2=',Q2);
  writeln(F0,'V2=',V2,' U2=',U2,' Q2=',Q2);
  VVOD33;
  GB3660(A3,E3,B3,J3,O3,V3,U3,Q3);
  writeln('Коэффициенты ортогонализации');
  writeln(F0,'Коэффициенты ортогонализации');
  writeln('V3=',V3,' U3=',U3,' Q3=',Q3);
  writeln(F0,'V3=',V3,' U3=',U3,' Q3=',Q3);
end;
end;{case}
end;{VV_DAN}

```

```

procedure OUT_Z_B;
begin
  writeln('Расчетные показатели Z(J) до анализа B(J)');
  writeln(F0,'Расчетные показатели Z(J) до анализа B(J)');
  writeln(F0,'
  |-----|');
  writeln(F0,'
  |   Значение   |');
  writeln(F0,'
  | J |-----|');
  writeln(F0,'
  | Y(J) | Z(J) | Y(J)-Z(J) |');
  for J:=1 to X do
  begin
    writeln(F0,'
    |-----|');
    write(F0,'
    | ',J:2,' | ',Y[J]:10,' | ',Z[J]:10,' | ');
    writeln(F0,(Y[J]-Z[J]):10,' | ');
    write('
    | ',J:2,' | ',Y[J]:10,' | ',Z[J]:10,' | ');
    writeln((Y[J]-Z[J]):10,' | ');
  end;
  writeln(F0,'
  |-----|');
end;
=====
{
  Основная программа
}
=====
begin
1: for J:=1 to 25 do writeln;
  writeln('  Программа на языке Турбо-Паскаль состоит из трех');
  writeln('  файлов tpg3_1,tpg3_2,tpg3_3(w4_1,w4_2,w4_3). Математическое');
  writeln('  моделирование на основе планирования экспериментов. ');
  writeln('  Переложение с языка Бейсик программы GGN3. ');
  writeln('  Разработка Черного А.А. ');
  writeln('  Autor Chernyy Anatoly Alekseevch');
  writeln('Введите имя файла для вывода resultant (без расширения)');
  readln(NAME);
  Assign(F0, (NAME+'.dat'));
  Rewrite(F0);
  writeln(F0,'Выполнение программы математического моделирования');
  writeln(F0,' tpg3_1(w4_1). (Программа на языке Турбо-Паскаль. ');
  writeln(F0,' Autor Chernyy Anatoly Alekseevch');
  writeln('
  _X=4 _X=16 _');
  writeln('X=3, _X=4, X=5,X=9,X=12,X=15, _X=16, X=20,X=25,X=27');
  J:=0;X:=0;F3:=0;F4:=0;H3:=0;H4:=0;
  L3:=0;L4:=0;K3:=0;K4:=0;K5:=0;
  K7:=0;K8:=0;K0:=0;X0:=0;Y0:=0;J5:=0;
  J6:=0;J9:=0;S:=0;
  writeln('Если X=0, то выход из программы');
  write('X=');readln(X);
  writeln(F0,'X=',X);
  for J:=1 to X do
  begin
    F[J]:=0; H[J]:=0; L[J]:=0; Z[J]:=0; KK5[J]:=0;
    KK6[J]:=0; KK7[J]:=0; J7[J]:=0; J8[J]:=0;
    JJ9[J]:=0;
  end;
  VV_DAN;
  if X=0 then goto 2;
  PR_MOD;
3: writeln('-----<Меню 1>-----');
  writeln('Если I0=6, то переход в начало');
  writeln('Если I0=7, то продолжение и');
  writeln('ввод величин показателей Y(J)');
  write('I0='); readln(I0);
  if I0=6 then goto 1;
  writeln('Ввод величин показателей Y(J)');
  writeln(F0,'Показатели Y(J)');
  for J:=1 to X do

```

```

begin
  write('Y(',J,')=');
  readln(Y[J]);
  writeln(F0,'Y(',J,')=',Y[J]);
end;
case X of
3: GB4150;
4: GB4210;
5: GB4290;
9: begin
  GB4150; GB4170;
  end;
12:begin
  GB4150; GB4250;
  end;
15:begin
  GB4150; GB4340;
  end;
16:begin
  GB4210; GB4250;
  end;
20:begin
  GB4210; GB4340;
  end;
25:begin
  GB4290; GB4340;
  end;
27:begin
  GB4150; GB4170; GB4190;
  end;
end; {case}
GB1410;
case X of
4: GB2000;
5: begin GB2000; GB2020; end;
9: begin GB1480; end;
12:begin GB1480; GB1600; end;
15:begin GB1480; GB1600; GB1670; end;
16:begin GB1480; GB1600; GB1930; end;
20:begin GB1480; GB1600; GB1670; GB1730; end;
25:begin GB1480; GB1600; GB1670; GB1730; GB1830; end;
27:begin GB1480; GB2040; end;
end;
writeln('B(J) до анализа');
writeln(F0,'B(J) до анализа');
for J:=1 to X do
  begin
  writeln('B(',J,')=',B[J]);
  writeln(F0,'B(',J,')=',B[J]);
  end;
case X of
3: GB4390;
4: GB4400;
5: GB4420;
9: GB4450;
12:GB4490;
15:GB4530;
16:GB4580;
20:GB4630;
25:GB4690;
27:GB4770;
else goto 2;
end;

```



```

OUT_Z_B;
4:writeln('-----<Меню 2>-----');
writeln('Если I0=3, то ввод результатов опытов');
writeln('на среднем уровне факторов');
writeln('Если I0=4, то ввод дисперсии опытов');
writeln('Если I0=5, то проверка точности и расчеты по модели');
writeln('Если I0=6, то переход в начало программы');
writeln('Если I0=20, то переход в конец программы');
writeln('Если I0=25, то вывод математической модели');
writeln('Если I0=27, то вычисление показателей Z(K5)');
writeln('с использованием циклов');
writeln('Если I0=30, то переход к меню 1');
write('Ввод I0=');
readln(I0);
if I0=6 then goto 1;
if I0=20 then goto 2;
if I0=30 then goto 3;
if I0=25 then begin GB4880; goto 2; end;
if I0=27 then begin GB7000; goto 2; end;
if (I0=3)or(I0=4) then
begin
writeln('Ввод N0-количество опытов на среднем уровне факторов');
readln(N0);
writeln(F0,'Количество опытов на среднем уровне факторов N0=',N0);
writeln('Ввод F8=N0-1');
readln(F8);
writeln(F0,'F8=N0-1=',F8);
end;
if I0=3 then
begin
writeln('Ввод G(J)-результаты опытов');
writeln('на среднем уровне факторов');
writeln(F0,'G(J)-результаты опытов');
writeln(F0,'на среднем уровне факторов');
for J:=1 to N0 do
begin
write('G(',J,')=');
readln(G[J]);
writeln(F0,'G(',J,')=',G[J]);
end;
SS:=0;
for J:=1 to N0 do SS:=SS+G[J];
S0:=SS/N0;
writeln('S0=',S0); SS:=0;
for J:=1 to N0 do SS:=SS+(G[J]-S0)*(G[J]-S0);
U9:=SS/F8;
writeln('Дисперсия опытов U9=',U9);
writeln(F0,'Дисперсия опытов U9=',U9);
end;
if I0=4 then
begin
writeln('Ввод U9-дисперсия опытов');
readln(U9);
writeln(F0,'Дисперсия опытов U9=',U9);
end;
if not(I0=5) then
begin
writeln('Расчетные величины T(J)');
writeln(F0,'Расчетные величины T(J)');
for J:=1 to X do
begin
T[J]:=abs(B[J]/sqrt(U9/O[J]));
writeln('T(',J,')=',T[J]);
end;

```

```

writeln(F0,'T('J,')=',T[J]);
end;
writeln(' ');
writeln('    Для уровня значимости 5%    ');
writeln(' ');
writeln(' F8 | 2 | 3 | 4 | 5 | 6 | ');
writeln(' ');
writeln(' T0 | 4,303 | 3,182 | 2,776 | 2,571 | 2,447 | ');
writeln(' ');
writeln('F8=N0-1=',N0,'-1=',F8:1);
writeln('Ввод T0-табличного T-критерия');
readln(T0);
writeln(F0,'Табличный T-критерий T0=',T0:5:3);
writeln('B(J) после анализа');
writeln(F0,'B(J) после анализа');
for J:=1 to X do
begin
if T[J]<T0 then B[J]:=0;
writeln('B('J,')=',B[J]);
writeln(F0,'B('J,')=',B[J]);
end;
K9:=0;
for J:=1 to X do
begin
if B[J]=0 then K9:=K9+1;
end;
writeln('Количество статистически значимых');
writeln('коэффициентов регрессии');
writeln('K9=',K9);
writeln('F9=X-1');
F9:=X-1;
writeln('F9=',F9);
writeln(F0,'Количество статистически значимых');
writeln(F0,'коэффициентов регрессии');
writeln(F0,'K9=',K9);
writeln(F0,'F9=X-1');
F9:=X-1;
writeln(F0,'F9=',F9);
tablF9;
writeln('F8=',F8:2,' F9=',F9:2);
writeln('Ввод F7-табличного F-критерия');
readln(F7);
writeln(F0,'Табличный F-критерий F7=',F7);
case X of
3: GB4390;
4: GB4400;
5: GB4420;
9: GB4450;
12:GB4490;
15:GB4530;
16:GB4580;
20:GB4630;
25:GB4690;
27:GB4770;
else goto 2;
end;
writeln('Расчетные величины показателя');
writeln('Z(J) после анализа B(J)');
writeln(F0,'Расчетные величины показателя');
writeln(F0,'Z(J) после анализа B(J)');
for J:=1 to X do
begin
writeln('Z('J,')=',Z[J]);

```

```

    writeln(F0,'Z('J,')=',Z[J]);
end;
SS:=0;
for J:=1 to X do
SS:=SS+(Z[J]-Y[J])*(Z[J]-Y[J]);
F6:=SS/(F9*U9);
writeln('Расчетная величина F-критерия F6=',F6);
writeln(F0,'Расчетная величина F-критерия F6=',F6);
if F6<=F7 then
begin
    writeln('Адекватно, так как F6<=F7');
    writeln(F0,'Адекватно, так как F6<=F7');
end;
if F6>F7 then
begin
    writeln('Неадекватно, так как F6>F7');
    writeln(F0,'Неадекватно, так как F6>F7');
end;
if X=4 then
begin
    writeln(F0,'X=4.System, 1.1 - 1.4, finish 1.4. ');
    writeln(F0,'1.1.', 'A1=',A1,'Z(1)=',Z[1]);
    writeln(F0,'1.2.', 'C1=',C1,'Z(3)=',Z[3]);
    writeln(F0,'1.3.', 'D1=',D1,'Z(4)=',Z[4]);
    writeln(F0,'1.4.', 'B1=',B1,'Z(2)=',Z[2]);
end;
if X=16 then
begin
    writeln(F0,'X=16.System, 1.1 - 8.4, finish 8.4. ');
    writeln(F0,'1.1.', 'A2=',A2,'A1=',A1,'Z(1)=',Z[1]);
    writeln(F0,'1.2.', 'A2=',A2,'C1=',C1,'Z(9)=',Z[9]);
    writeln(F0,'1.3.', 'A2=',A2,'D1=',D1,'Z(13)=',Z[13]);
    writeln(F0,'1.4.', 'A2=',A2,'B1=',B1,'Z(2)=',Z[2]);
    writeln(F0,'2.1.', 'C2=',C2,'A1=',A1,'Z(5)=',Z[5]);
    writeln(F0,'2.2.', 'C2=',C2,'C1=',C1,'Z(10)=',Z[10]);
    writeln(F0,'2.3.', 'C2=',C2,'D1=',D1,'Z(14)=',Z[14]);
    writeln(F0,'2.4.', 'C2=',C2,'B1=',B1,'Z(6)=',Z[6]);
    writeln(F0,'3.1.', 'D2=',D2,'A1=',A1,'Z(7)=',Z[7]);
    writeln(F0,'3.2.', 'D2=',D2,'C1=',C1,'Z(11)=',Z[11]);
    writeln(F0,'3.3.', 'D2=',D2,'D1=',D1,'Z(15)=',Z[15]);
    writeln(F0,'3.4.', 'D2=',D2,'B1=',B1,'Z(8)=',Z[8]);
    writeln(F0,'4.1.', 'B2=',B2,'A1=',A1,'Z(3)=',Z[3]);
    writeln(F0,'4.2.', 'B2=',B2,'C1=',C1,'Z(12)=',Z[12]);
    writeln(F0,'4.3.', 'B2=',B2,'D1=',D1,'Z(16)=',Z[16]);
    writeln(F0,'4.4.', 'B2=',B2,'B1=',B1,'Z(4)=',Z[4]);
    writeln(F0,'5.1.', 'A1=',A1,'A2=',A2,'Z(1)=',Z[1]);
    writeln(F0,'5.2.', 'A1=',A1,'C2=',C2,'Z(5)=',Z[5]);
    writeln(F0,'5.3.', 'A1=',A1,'D2=',D2,'Z(7)=',Z[7]);
    writeln(F0,'5.4.', 'A1=',A1,'B2=',B2,'Z(3)=',Z[3]);
    writeln(F0,'6.1.', 'C1=',C1,'A2=',A2,'Z(9)=',Z[9]);
    writeln(F0,'6.2.', 'C1=',C1,'C2=',C2,'Z(10)=',Z[10]);
    writeln(F0,'6.3.', 'C1=',C1,'D2=',D2,'Z(11)=',Z[11]);
    writeln(F0,'6.4.', 'C1=',C1,'B2=',B2,'Z(12)=',Z[12]);
    writeln(F0,'7.1.', 'D1=',D1,'A2=',A2,'Z(13)=',Z[13]);
    writeln(F0,'7.2.', 'D1=',D1,'C2=',C2,'Z(14)=',Z[14]);
    writeln(F0,'7.3.', 'D1=',D1,'D2=',D2,'Z(15)=',Z[15]);
    writeln(F0,'7.4.', 'D1=',D1,'B2=',B2,'Z(16)=',Z[16]);
    writeln(F0,'8.1.', 'B1=',B1,'A2=',A2,'Z(2)=',Z[2]);
    writeln(F0,'8.2.', 'B1=',B1,'C2=',C2,'Z(6)=',Z[6]);
    writeln(F0,'8.3.', 'B1=',B1,'D2=',D2,'Z(8)=',Z[8]);
    writeln(F0,'8.4.', 'B1=',B1,'B2=',B2,'Z(4)=',Z[4]);
end;
case X of

```

```

3: OUT_F_H_L(73);
4: OUT_F_H_L(73);
5: OUT_F_H_L(73);
9: OUT_F_H_L(74);
12:OUT_F_H_L(74);
15:OUT_F_H_L(74);
16:OUT_F_H_L(74);
20:OUT_F_H_L(74);
25:OUT_F_H_L(74);
27:OUT_F_H_L(75);
end;
end;
5:writeln('-----<Меню 3-----');
writeln('Если I0=7, то проверка точности');
writeln('и расчеты по модели');
writeln('Если I0=8, то переход в начало');
writeln('Если I0=17, то математическая модель');
writeln('Если I0=22, то вычисления');
writeln('показателей Z(K5) с использованием');
writeln('циклов');
writeln('Если I0=9, то выход из программы');
writeln('Если I0=25, то переход к меню 1');
writeln('Если I0=27, то переход к меню 2');
write('Ввод I0=');
readln(I0);
if I0=8 then goto 1;
if I0=9 then goto 2;
if I0=25 then goto 3;
if I0=27 then goto 4;
if I0=17 then begin GB4880; goto 2; end;
if I0=22 then begin GB7000; goto 2; end;
writeln('Проверка точности и расчеты по модели');
writeln('F(S), H(S), L(S) - 1,2,3 факторы');
writeln('где S=X=',X,'-количество опытов по плану');
writeln('расчеты по модели');
writeln(F0,'Проверка точности и расчеты по модели');
writeln(F0,'F(S), H(S), L(S) - 1,2,3 факторы');
writeln(F0,'где S=X=',X,'-количество опытов по плану');
writeln(F0,'расчеты по модели');
if (X=3)or(X=4)or(X=5)then
begin
for S:=1 to X do
begin
F[S]:=0; Z[S]:=0;
writeln('Ввод F(',S,')');
readln(F[S]);
case X of
3: begin
GB4150; GB4390;
end;
4: begin
GB4210; GB4400;
end;
5: begin
GB4290; GB4420;
end;
else goto 2;
end;{case}
writeln('Z(',S,')=',Z[S]);
end;{for}
OUT_F_H_L(73);
end;{if}
if (X=9)or(X=12)or(X=15)or(X=16)or(X=20)or(X=25)then

```

```

begin
  for S:=1 to X do
    begin
      F[S]:=0; Z[S]:=0; H[S]:=0;
      writeln('Ввод F(',S,'),' H(',S,')');
      readln(F[S],H[S]);
      case X of
        9: begin
            GB4150; GB4170; GB4450;
          end;
        12:begin
            GB4150; GB4250; GB4490;
          end;
        15:begin
            GB4150; GB4340; GB4530;
          end;
        16:begin
            GB4210; GB4250; GB4580;
          end;
        20:begin
            GB4210; GB4340; GB4630;
          end;
        25:begin
            GB4290; GB4340; GB4690;
          end;
      else goto 2;
      end; {case}
      writeln('Z(',S,)'=',Z[S]);
    end; {for}
    OUT_F_H_L(74);
  end; {if}
  if X=27 then
    begin
      for S:=1 to X do
        begin
          writeln('Ввод F(',S,'),' H(',S,'),' L(',S,')');
          readln(F[S],H[S],L[S]);
          GB4150; GB4170; GB4190; GB4770;
          writeln('Z(',S,)'=',Z[S]);
        end; {for}
      OUT_F_H_L(75);
    end; {if}
  2:writeln('-----<Меню 4-----');
  writeln('Если I0=10, то проверка точности');
  writeln('и расчеты по модели');
  writeln('Если I0=14, то вычисления ');
  writeln('показателей Z(K5) с использованием');
  writeln('циклов');
  writeln('Если I0=35, то переход к меню 1');
  writeln('Если I0=18, то переход к меню 2');
  writeln('Если I0=19, то переход к меню 3');
  writeln('Если I0=50, то переход в начало программы');
  writeln('Если I0=44, то выход из программы');
  write('Ввод I0=');
  readln(I0);
  if I0=11 then goto 1;
  if I0=18 then goto 4;
  if (I0=19)or(I0=10) then goto 5;
  if I0=35 then goto 3;
  if I0=14 then
    begin
      GB7000;
      goto 2;
    end;

```

```

    end;
    writeln('Конец работы');
    close(F0);
end. {Конец программы}
unit TPG3_2;
Interface
type artp=array[1..50] of real;
var
    J6,K5,S,J,I0,X,N0,K9,K4,F8,F9:integer;
    G1,T1,A1,C1,E1,D1,B1,J1,O1,A2,E2,B2,J2,O2,A3:real;
    H1,K1,L1,F1,M1,I1,P1,E3,B3,J3,O3,V1,U1,Q1,V2,U2,Q2:real;
    P2,F2,I2,H2,M2,N3,N4,N5,N6,N7,R3,R4,R0,R5,R6:real;
    S0,J5,S1,V3,U3,Q3,U9,F7,SO,F6:real;
    J9,L3,SS,L4,H4,H3,F3,F4,C2,D2,T2:real;
    T0,G2,K2,L2,K7,K8,X0,Y0,K0,K3:real;
    J7,J8,J9,KK5,KK6,KK7,KK8,F,H,V,L,Y,I,K,M,P,Q,U,O,B,Z,G,T:artp;
    NAME:string[8];
    F0: Text;
procedure GB3660(A,B,E,N,R:real; var V0,U0,Q0:real);
procedure GB3710(A,B,C,D,N,R,S:real; var V0,U0,Q0,I0,M0,F0:real);
procedure GB3860(A,B,C,D,E,N,R,S,W:real;
var V0,U0,Q0,I0,M0,F0,G0,H0,K0,L0:real); function ste(Q,A:real):real;
procedure GB1410; procedure GB1480; procedure GB1600;
procedure GB1670; procedure GB1730; procedure GB1830;
procedure GB1930; procedure GB2000; procedure GB2020;
procedure GB2040; procedure GB4150; procedure GB4210;
procedure GB4290; procedure GB4170; procedure GB4190;
procedure GB4250; procedure GB4340; procedure GB4400;
procedure GB4420; procedure GB4450; procedure GB4490;
procedure GB4530; procedure GB4580; procedure GB4630;
procedure GB4690; procedure GB4390; procedure GB4770;
Implementation
function ste(Q,A:real):real;
{Функция возведения в степень}
begin
    if A<0 then
        begin
            A:=abs(A);
            if Q=0 then Q:=1E-5;
            ste:=1/exp(A*ln(Q));
        end
    else
        begin
            if Q=0 then Q:=1E-5;
            if A=0 then ste:=1
                else ste:=exp(A*ln(Q));
            end;
        end;
end; {ste}
procedure GB1410;
begin
    SS:=0; O[1]:=0;
    for J:=1 to X do
        begin
            SS:=SS+Y[J]; O[1]:=O[1]+1;
        end;
    B[1]:=SS/O[1];
    SS:=0; O[2]:=0;
    for J:=1 to X do
        begin
            SS:=SS+I[J]*Y[J]; O[2]:=O[2]+I[J]*I[J];
        end;
    B[2]:=SS/O[2];
    SS:=0; O[3]:=0;

```

```

for J:=1 to X do
begin
SS:=SS+K[J]*Y[J]; O[3]:=O[3]+K[J]*K[J];
end;
B[3]:=SS/O[3];
end;{GB1410}
procedure GB1480;
begin
SS:=0; O[4]:=0;
for J:=1 to X do
begin
SS:=SS+P[J]*Y[J];
O[4]:=O[4]+P[J]*P[J];
end;
B[4]:=SS/O[4]; SS:=0; O[5]:=0;
for J:=1 to X do
begin
SS:=SS+I[J]*P[J]*Y[J];
O[5]:=O[5]+(I[J]*P[J])*(I[J]*P[J]);
end;
B[5]:=SS/O[5]; SS:=0; O[6]:=0;
for J:=1 to X do
begin
SS:=SS+Q[J]*Y[J];
O[6]:=O[6]+Q[J]*Q[J];
end;
B[6]:=SS/O[6]; SS:=0; O[7]:=0;
for J:=1 to X do
begin
SS:=SS+I[J]*Q[J]*Y[J];
O[7]:=O[7]+(I[J]*Q[J])*(I[J]*Q[J]);
end;
B[7]:=SS/O[7]; SS:=0; O[8]:=0;
for J:=1 to X do
begin
SS:=SS+P[J]*K[J]*Y[J];
O[8]:=O[8]+(P[J]*K[J])*(P[J]*K[J]);
end;
B[8]:=SS/O[8]; SS:=0; O[9]:=0;
for J:=1 to X do
begin
SS:=SS+K[J]*Q[J]*Y[J];
O[9]:=O[9]+(K[J]*Q[J])*(K[J]*Q[J]);
end;
B[9]:=SS/O[9];
end;{GB1480}
procedure GB1600;
begin
SS:=0; O[10]:=0;
for J:=1 to X do
begin
SS:=SS+U[J]*Y[J];
O[10]:=O[10]+U[J]*U[J];
end;
B[10]:=SS/O[10]; SS:=0; O[11]:=0;
for J:=1 to X do
begin
SS:=SS+I[J]*U[J]*Y[J];
O[11]:=O[11]+(I[J]*U[J])*(I[J]*U[J]);
end;
B[11]:=SS/O[11]; SS:=0; O[12]:=0;
for J:=1 to X do
begin

```

```

    SS:=SS+K[J]*U[J]*Y[J];
    O[12]:=O[12]+K[J]*U[J]*K[J]*U[J];
end;
B[12]:=SS/O[12];
end;{GB1600}
procedure GB1670;
begin
    SS:=0; O[13]:=0;
    for J:=1 to X do
        begin
            SS:=SS+V[J]*Y[J];
            O[13]:=O[13]+V[J]*V[J];
        end;
    B[13]:=SS/O[13]; SS:=0; O[14]:=0;
    for J:=1 to X do
        begin
            SS:=SS+I[J]*V[J]*Y[J];
            O[14]:=O[14]+I[J]*V[J]*I[J]*V[J];
        end;
    B[14]:=SS/O[14]; SS:=0; O[15]:=0;
    for J:=1 to X do
        begin
            SS:=SS+K[J]*V[J]*Y[J];
            O[15]:=O[15]+K[J]*V[J]*K[J]*V[J];
        end;
    B[15]:=SS/O[15];
end;{GB1670}
procedure GB1730;
begin
    SS:=0; O[16]:=0;
    for J:=1 to X do
        begin
            SS:=SS+L[J]*Y[J];
            O[16]:=O[16]+L[J]*L[J];
        end;
    B[16]:=SS/O[16]; SS:=0; O[17]:=0;
    for J:=1 to X do
        begin
            SS:=SS+P[J]*L[J]*Y[J];
            O[17]:=O[17]+(P[J]*L[J])*(P[J]*L[J]);
        end;
    B[17]:=SS/O[17]; SS:=0; O[18]:=0;
    for J:=1 to X do
        begin
            SS:=SS+Q[J]*L[J]*Y[J];
            O[18]:=O[18]+(Q[J]*L[J])*(Q[J]*L[J]);
        end;
    B[18]:=SS/O[18]; SS:=0; O[19]:=0;
    for J:=1 to X do
        begin
            SS:=SS+L[J]*U[J]*Y[J];
            O[19]:=O[19]+(L[J]*U[J])*(L[J]*U[J]);
        end;
    B[19]:=SS/O[19]; SS:=0; O[20]:=0;
    for J:=1 to X do
        begin
            SS:=SS+L[J]*V[J]*Y[J];
            O[20]:=O[20]+(L[J]*V[J])*(L[J]*V[J]);
        end;
    B[20]:=SS/O[20];
end;{GB1730}
procedure GB1830;
begin

```



```

SS:=0; O[21]:=0;
for J:=1 to X do
begin
SS:=SS+M[J]*Y[J];
O[21]:=O[21]+M[J]*M[J];
end;
B[21]:=SS/O[21]; SS:=0; O[22]:=0;
for J:=1 to X do
begin
SS:=SS+P[J]*M[J]*Y[J];
O[22]:=O[22]+(P[J]*M[J])*(P[J]*M[J]);
end;
B[22]:=SS/O[22]; SS:=0; O[23]:=0;
for J:=1 to X do
begin
SS:=SS+Q[J]*M[J]*Y[J];
O[23]:=O[23]+(Q[J]*M[J])*(Q[J]*M[J]);
end;
B[23]:=SS/O[23]; SS:=0; O[24]:=0;
for J:=1 to X do
begin
SS:=SS+U[J]*M[J]*Y[J];
O[24]:=O[24]+(U[J]*M[J])*(U[J]*M[J]);
end;
B[24]:=SS/O[24]; SS:=0; O[25]:=0;
for J:=1 to X do
begin
SS:=SS+M[J]*V[J]*Y[J];
O[25]:=O[25]+(M[J]*V[J])*(M[J]*V[J]);
end;
B[25]:=SS/O[25];
end;{GB1830}
procedure GB1930;
begin
SS:=0; O[13]:=0;
for J:=1 to X do
begin
SS:=SS+L[J]*Y[J];
O[13]:=O[13]+L[J]*L[J];
end;
B[13]:=SS/O[13]; SS:=0; O[14]:=0;
for J:=1 to X do
begin
SS:=SS+P[J]*L[J]*Y[J];
O[14]:=O[14]+(P[J]*L[J])*(P[J]*L[J]);
end;
B[14]:=SS/O[14]; SS:=0; O[15]:=0;
for J:=1 to X do
begin
SS:=SS+Q[J]*L[J]*Y[J];
O[15]:=O[15]+(Q[J]*L[J])*(Q[J]*L[J]);
end;
B[15]:=SS/O[15]; SS:=0; O[16]:=0;
for J:=1 to X do
begin
SS:=SS+L[J]*U[J]*Y[J];
O[16]:=O[16]+(L[J]*U[J])*(L[J]*U[J]);
end;
B[16]:=SS/O[16];
end;{GB1930}
procedure GB2000;
begin
SS:=0; O[4]:=0;

```

```

for J:=1 to X do
begin
SS:=SS+L[J]*Y[J];
O[4]:=O[4]+L[J]*L[J];
end;
B[4]:=SS/O[4];
end;{GB2000}
procedure GB2020;
begin
SS:=0; O[5]:=0;
for J:=1 to X do
begin
SS:=SS+M[J]*Y[J];
O[5]:=O[5]+M[J]*M[J];
end;
B[5]:=SS/O[5];
end;{GB2020}
procedure GB2040;
begin
SS:=0; O[10]:=0;
for J:=1 to X do
begin
SS:=SS+U[J]*Y[J];
O[10]:=O[10]+U[J]*U[J];
end;
B[10]:=SS/O[10]; SS:=0; O[11]:=0;
for J:=1 to X do
begin
SS:=SS+I[J]*U[J]*Y[J];
O[11]:=O[11]+(I[J]*U[J])*(I[J]*U[J]);
end;
B[11]:=SS/O[11]; SS:=0; O[12]:=0;
for J:=1 to X do
begin
SS:=SS+P[J]*U[J]*Y[J];
O[12]:=O[12]+(P[J]*U[J])*(P[J]*U[J]);
end;
B[12]:=SS/O[12]; SS:=0; O[13]:=0;
for J:=1 to X do
begin
SS:=SS+I[J]*P[J]*U[J]*Y[J];
O[13]:=O[13]+(I[J]*P[J]*U[J])*(I[J]*P[J]*U[J]);
end;
B[13]:=SS/O[13]; SS:=0; O[14]:=0;
for J:=1 to X do
begin
SS:=SS+V[J]*Y[J];
O[14]:=O[14]+V[J]*V[J];
end;
B[14]:=SS/O[14]; SS:=0; O[15]:=0;
for J:=1 to X do
begin
SS:=SS+I[J]*V[J]*Y[J];
O[15]:=O[15]+(I[J]*V[J])*(I[J]*V[J]);
end;
B[15]:=SS/O[15]; SS:=0; O[16]:=0;
for J:=1 to X do
begin
SS:=SS+P[J]*V[J]*Y[J];
O[16]:=O[16]+(P[J]*V[J])*(P[J]*V[J]);
end;
B[16]:=SS/O[16]; SS:=0; O[17]:=0;
for J:=1 to X do

```

```

begin
  SS:=SS+U[J]*K[J]*Y[J];
  O[17]:=O[17]+(U[J]*K[J])*(U[J]*K[J]);
end;
B[17]:=SS/O[17]; SS:=0; O[18]:=0;
for J:=1 to X do
  begin
    SS:=SS+U[J]*Q[J]*Y[J];
    O[18]:=O[18]+(U[J]*Q[J])*(U[J]*Q[J]);
  end;
B[18]:=SS/O[18]; SS:=0; O[19]:=0;
for J:=1 to X do
  begin
    SS:=SS+I[J]*P[J]*V[J]*Y[J];
    O[19]:=O[19]+(I[J]*P[J]*V[J])*(I[J]*P[J]*V[J]);
  end;
B[19]:=SS/O[19]; SS:=0; O[20]:=0;
for J:=1 to X do
  begin
    SS:=SS+I[J]*U[J]*Q[J]*Y[J];
    O[20]:=O[20]+(I[J]*U[J]*Q[J])*(I[J]*U[J]*Q[J]);
  end;
B[20]:=SS/O[20]; SS:=0; O[21]:=0;
for J:=1 to X do
  begin
    SS:=SS+P[J]*U[J]*K[J]*Y[J];
    O[21]:=O[21]+(P[J]*U[J]*K[J])*(P[J]*U[J]*K[J]);
  end;
B[21]:=SS/O[21]; SS:=0; O[22]:=0;
for J:=1 to X do
  begin
    SS:=SS+K[J]*V[J]*Y[J];
    O[22]:=O[22]+(K[J]*V[J])*(K[J]*V[J]);
  end;
B[22]:=SS/O[22]; SS:=0; O[23]:=0;
for J:=1 to X do
  begin
    SS:=SS+Q[J]*V[J]*Y[J];
    O[23]:=O[23]+(Q[J]*V[J])*(Q[J]*V[J]);
  end;
B[23]:=SS/O[23]; SS:=0; O[24]:=0;
for J:=1 to X do
  begin
    SS:=SS+I[J]*Q[J]*V[J]*Y[J];
    O[24]:=O[24]+(I[J]*Q[J]*V[J])*(I[J]*Q[J]*V[J]);
  end;
B[24]:=SS/O[24]; SS:=0; O[25]:=0;
for J:=1 to X do
  begin
    SS:=SS+P[J]*K[J]*V[J]*Y[J];
    O[25]:=O[25]+(P[J]*K[J]*V[J])*(P[J]*K[J]*V[J]);
  end;
B[25]:=SS/O[25]; SS:=0; O[26]:=0;
for J:=1 to X do
  begin
    SS:=SS+U[J]*K[J]*Q[J]*Y[J];
    O[26]:=O[26]+(U[J]*K[J]*Q[J])*(U[J]*K[J]*Q[J]);
  end;
B[26]:=SS/O[26]; SS:=0; O[27]:=0;
for J:=1 to X do
  begin
    SS:=SS+K[J]*Q[J]*V[J]*Y[J];
    O[27]:=O[27]+(K[J]*Q[J]*V[J])*(K[J]*Q[J]*V[J]);
  end;

```

```

    end;
    B[27]:=SS/O[27];
end;{GB2040}
procedure GB3660(A,B,E,N,R:real; var V0,U0,Q0:real);
var L2,N0,R0,N3,N4,N5:real;
begin
    N0:=(ste(A,N)+ste(B,N)+ste(E,N))/3;
    R0:=(ste(A,R)+ste(B,R)+ste(E,R))/3;
    L2:=2*N;
    N3:=(ste(A,L2)+ste(B,L2)+ste(E,L2))/3;
    N4:=N+R;
    N5:=(ste(A,N4)+ste(B,N4)+ste(E,N4))/3;
    V0:=-N0;
    U0:=(N0*R0-N5)/(N3-N0*N0);
    Q0:=(R0+U0*N0);
end;{GB3660}
procedure GB3710(A,B,C,D,N,R,S:real; var V0,U0,Q0,I0,M0,F0:real);
var
    N0,R0,S0,L2,N3,K2,R3,N4,N5,N6,N7,R4,R5,P0,Z1,Z2,Z3:real;
begin
    N0:=(ste(A,N)+ste(B,N)+ste(C,N)+ste(D,N))/4;
    R0:=(ste(A,R)+ste(B,R)+ste(C,R)+ste(D,R))/4;
    S0:=(ste(A,S)+ste(B,S)+ste(C,S)+ste(D,S))/4;
    L2:=2*N;
    N3:=(ste(A,L2)+ste(B,L2)+ste(C,L2)+ste(D,L2))/4;
    K2:=2*R;
    R3:=(ste(A,K2)+ste(B,K2)+ste(C,K2)+ste(D,K2))/4;
    N4:=N+R;
    N5:=(ste(A,N4)+ste(B,N4)+ste(C,N4)+ste(D,N4))/4;
    N6:=N+S;
    N7:=(ste(A,N6)+ste(B,N6)+ste(C,N6)+ste(D,N6))/4;
    R4:=R+S;
    R5:=(ste(A,R4)+ste(B,R4)+ste(C,R4)+ste(D,R4))/4;
    V0:=-N0;
    U0:=(N0*R0-N5)/(N3-N0*N0);
    Q0:=(R0+U0*N0);
    P0:=(N0*S0-N7)/(N3-N0*N0);
    Z1:=R0*S0-R5+P0*(N0*R0-N5);
    Z2:=U0*(N0*S0-N7)+U0*P0*(N0*N0-N3);
    Z3:=R3-R0*R0+2*U0*(N5-N0*R0);
    I0:=(Z1+Z2)/(Z3+(N3-N0*N0)*U0*U0);
    M0:=I0*U0+P0;
    F0:=(S0+I0*R0+M0*N0);
end;{GB3710}
procedure GB3860(A,B,C,D,E,N,R,S,W:real;
var V0,U0,Q0,I0,M0,F0,G0,H0,K0,L0:real);
var
    N0,R0,S0,W0,L2,N3,K2,R3,M2,S3,N4,N5:real;
    N6,N7,N8,N9,R4,R5,R6,R7,S4,S5,P0:real;
    Z1,Z2,Z3,Z4,Z5,Z6,Z7,Z0,Z8,Z9,T7:real;
    T8,T9,G3,G4,G5,G6,G7:real;
begin
    N0:=(ste(A,N)+ste(B,N)+ste(C,N)+ste(D,N)+ste(E,N))/5;
    R0:=(ste(A,R)+ste(B,R)+ste(C,R)+ste(D,R)+ste(E,R))/5;
    S0:=(ste(A,S)+ste(B,S)+ste(C,S)+ste(D,S)+ste(E,S))/5;
    W0:=(ste(A,W)+ste(B,W)+ste(C,W)+ste(D,W)+ste(E,W))/5;
    L2:=2*N;
    N3:=(ste(A,L2)+ste(B,L2)+ste(C,L2)+ste(D,L2)+ste(E,L2))/5;
    K2:=2*R;
    R3:=(ste(A,K2)+ste(B,K2)+ste(C,K2)+ste(D,K2)+ste(E,K2))/5;
    M2:=2*S;
    S3:=(ste(A,M2)+ste(B,M2)+ste(C,M2)+ste(D,M2)+ste(E,M2))/5;
    N4:=N+R;

```

```

N5:=(ste(A,N4)+ste(B,N4)+ste(C,N4)+ste(D,N4)+ste(E,N4))/5;
N6:=N+S;
N7:=(ste(A,N6)+ste(B,N6)+ste(C,N6)+ste(D,N6)+ste(E,N6))/5;
N8:=N+W;
N9:=(ste(A,N8)+ste(B,N8)+ste(C,N8)+ste(D,N8)+ste(E,N8))/5;
R4:=R+S;
R5:=(ste(A,R4)+ste(B,R4)+ste(C,R4)+ste(D,R4)+ste(E,R4))/5;
R6:=R+W;
R7:=(ste(A,R6)+ste(B,R6)+ste(C,R6)+ste(D,R6)+ste(E,R6))/5;
S4:=S+W;
S5:=(ste(A,S4)+ste(B,S4)+ste(C,S4)+ste(D,S4)+ste(E,S4))/5;
V0:=-N0;
U0:=(N0*R0-N5)/(N3-N0*N0);
Q0:=(R0+U0*N0);
P0:=(N0*S0-N7)/(N3-N0*N0);
Z1:=R0*S0-R5+P0*(N0*R0-N5);
Z2:=U0*(N0*S0-N7)+U0*P0*(N0*N0-N3);
Z3:=R3-R0*R0+2*U0*(N5-N0*R0);
I0:=(Z1+Z2)/(Z3+(N3-N0*N0)*U0*U0);
M0:=I0*U0+P0;
F0:=(S0+I0*R0+M0*N0);
Z4:=R0+U0*N0;
Z5:=Z4*N0-N5-U0*N3;
Z6:=R3+U0*N5-Z4*R0-Z5*U0;
Z7:=Z4*S0+Z5*P0-R5-U0*N7;
Z0:=(N0*W0-N9)/(N3-N0*N0);
Z8:=Z5*Z0+Z4*W0-R7-U0*N9;
Z9:=S3+I0*R5+M0*N7;
T7:=R5+I0*R3+M0*N5;
T8:=N7+I0*N5+M0*N3;
T9:=S0+I0*R0+M0*N0;
G3:=S5+I0*R7+M0*N9;
G4:=T9*N0-T8;
G5:=Z9-T9*S0-G4*P0;
G6:=T9*R0-T7+G4*U0;
G7:=G4*Z0+T9*W0-G3;
G0:=(Z6*G7+Z8*G6)/(Z6*G5-Z7*G6);
H0:=(G0*Z7+Z8)/Z6;
K0:=G0*P0+H0*U0+Z0;
L0:=(W0+G0*S0+H0*R0+K0*N0);
end;{GB3860}
procedure GB4150;
begin
for J:=1 to X do
begin
I[J]:=ste(F[J],J1)+V1;
K[J]:=ste(F[J],O1)+U1*ste(F[J],J1)+Q1;
end;
end;{GB4150}
procedure GB4210;
begin
for J:=1 to X do
begin
I[J]:=ste(F[J],J1)+V1;
K[J]:=ste(F[J],O1)+U1*ste(F[J],J1)+Q1;
L[J]:=ste(F[J],P1)+I1*ste(F[J],O1)+M1*ste(F[J],J1)+F1;
end;
end;{Gb4210}
procedure GB4290;
begin
for J:=1 to X do
begin
I[J]:=ste(F[J],J1)+V1;

```

```

    K[J]:=ste(F[J],O1)+U1*ste(F[J],J1)+Q1;
    L[J]:=ste(F[J],P1)+I1*ste(F[J],O1)+M1*ste(F[J],J1)+F1;
    M[J]:=ste(F[J],T1)+G1*ste(F[J],P1)+H1*ste(F[J],O1)
    +K1*ste(F[J],J1)+L1;
  end;
end;{GB4290}
procedure GB4170;
begin
  for J:=1 to X do
    begin
      P[J]:=ste(H[J],J2)+V2;
      Q[J]:=ste(H[J],O2)+U2*ste(H[J],J2)+Q2;
    end;
  end;{GB4170}
procedure GB4190;
begin
  for J:=1 to X do
    begin
      U[J]:=ste(L[J],J3)+V3;
      V[J]:=ste(L[J],O3)+U3*ste(L[J],J3)+Q3;
    end;
  end;{GB4190}
procedure GB4250;
begin
  for J:=1 to X do
    begin
      P[J]:=ste(H[J],J2)+V2;
      Q[J]:=ste(H[J],O2)+U2*ste(H[J],J2)+Q2;
      U[J]:=ste(H[J],P2)+I2*ste(H[J],O2)+M2*ste(H[J],J2)+F2;
    end;
  end;{GB4250}
procedure GB4340;
begin
  for J:=1 to X do
    begin
      P[J]:=ste(H[J],J2)+V2;
      Q[J]:=ste(H[J],O2)+U2*ste(H[J],J2)+Q2;
      U[J]:=ste(H[J],P2)+I2*ste(H[J],O2)+M2*ste(H[J],J2)+F2;
      V[J]:=ste(H[J],T2)+G2*ste(H[J],P2)+H2*ste(H[J],O2)
      +K2*ste(H[J],J2)+L2;
    end;
  end;{GB4340}
procedure GB4400;
begin
  for J:=1 to X do
    Z[J]:=B[1]+B[2]*I[J]+B[3]*K[J]+B[4]*L[J];
  end;{GB4400}
procedure GB4420;
begin
  for J:=1 to X do
    Z[J]:=B[1]+B[2]*I[J]+B[3]*K[J]+B[4]*L[J]+B[5]*M[J];
  end;{GB4420}
procedure GB4450;
begin
  for J:=1 to X do
    begin
      N3:=B[1]+B[2]*I[J]+B[3]*K[J]+B[4]*P[J];
      N4:=B[5]*I[J]*P[J]+B[6]*Q[J]+B[7]*I[J]*Q[J];
      N5:=B[8]*P[J]*K[J]+B[9]*K[J]*Q[J];
      Z[J]:=N3+N4+N5;
    end;
  end;{GB4450}
procedure GB4490;

```

```

begin
  for J:=1 to X do
    begin
      N3:=B[1]+B[2]*I[J]+B[3]*K[J]+B[4]*P[J];
      N4:=B[5]*I[J]*P[J]+B[6]*Q[J]+B[7]*I[J]*Q[J]+B[8]*P[J]*K[J];
      N5:=B[9]*K[J]*Q[J]+B[10]*U[J]+B[11]*I[J]*U[J];
      Z[J]:=N3+N4+N5+B[12]*K[J]*U[J];
    end;
  end;{GB4490}
procedure GB4530;
begin
  for J:=1 to X do
    begin
      N3:=B[1]+B[2]*I[J]+B[3]*K[J]+B[4]*P[J];
      N4:=B[5]*I[J]*P[J]+B[6]*Q[J]+B[7]*I[J]*Q[J]+B[8]*P[J]*K[J];
      N5:=B[9]*K[J]*Q[J]+B[10]*U[J]+B[11]*I[J]*U[J];
      N6:=B[12]*K[J]*U[J]+B[13]*V[J]+B[14]*I[J]*V[J];
      Z[J]:=N3+N4+N5+N6+B[15]*K[J]*V[J];
    end;
  end;{GB4530}
procedure GB4580;
begin
  for J:=1 to X do
    begin
      N3:=B[1]+B[2]*I[J]+B[3]*K[J]+B[4]*P[J];
      N4:=B[5]*I[J]*P[J]+B[6]*Q[J]+B[7]*I[J]*Q[J]+B[8]*P[J]*K[J];
      N5:=B[9]*K[J]*Q[J]+B[10]*U[J]+B[11]*I[J]*U[J]+B[12]*K[J]*U[J];
      N6:=B[13]*L[J]+B[14]*P[J]*L[J]+B[15]*Q[J]*L[J];
      Z[J]:=N3+N4+N5+N6+B[16]*L[J]*U[J];
    end;
  end;{GB4580}
procedure GB4630;
begin
  for J:=1 to X do
    begin
      N3:=B[1]+B[2]*I[J]+B[3]*K[J]+B[4]*P[J];
      N4:=B[5]*I[J]*P[J]+B[6]*Q[J]+B[7]*I[J]*Q[J]+B[8]*P[J]*K[J];
      N5:=B[9]*K[J]*Q[J]+B[10]*U[J]+B[11]*I[J]*U[J];
      N6:=B[12]*K[J]*U[J]+B[13]*V[J]+B[14]*I[J]*V[J]+B[15]*K[J]*V[J];
      N7:=B[16]*L[J]+B[17]*P[J]*L[J]+B[18]*Q[J]*L[J]+B[19]*L[J]*U[J];
      Z[J]:=N3+N4+N5+N6+N7+B[20]*L[J]*V[J];
    end;
  end;{GB4630}
procedure GB4690;
begin
  for J:=1 to X do
    begin
      N3:=B[1]+B[2]*I[J]+B[3]*K[J]+B[4]*P[J];
      N4:=B[5]*I[J]*P[J]+B[6]*Q[J]+B[7]*I[J]*Q[J]+B[8]*P[J]*K[J];
      N5:=B[9]*K[J]*Q[J]+B[10]*U[J]+B[11]*I[J]*U[J];
      N6:=B[12]*K[J]*U[J]+B[13]*V[J]+B[14]*I[J]*V[J]+B[15]*K[J]*V[J];
      N7:=B[16]*L[J]+B[17]*P[J]*L[J]+B[18]*Q[J]*L[J]+B[19]*L[J]*U[J];
      R3:=B[20]*L[J]*V[J]+B[21]*M[J]+B[22]*P[J]*M[J];
      R4:=B[23]*Q[J]*M[J]+B[24]*U[J]*M[J]+B[25]*M[J]*V[J];
      Z[J]:=N3+N4+N5+N6+N7+R3+R4;
    end;
  end;{GB4690}
procedure GB4390;
begin
  for J:=1 to X do
    Z[J]:=B[1]+B[2]*I[J]+B[3]*K[J];
  end;{GB4390}
procedure GB4770;

```

```

begin
  for J:=1 to X do
    begin
      N3:=B[1]+B[2]*I[J]+B[3]*K[J]+B[4]*P[J];
      N4:=B[5]*I[J]*P[J]+B[6]*Q[J]+B[7]*I[J]*Q[J]+B[8]*P[J]*K[J];
      N5:=B[9]*K[J]*Q[J]+B[10]*U[J]+B[11]*I[J]*U[J]+B[12]*P[J]*U[J];
      N6:=B[13]*I[J]*P[J]*U[J]+B[14]*V[J]+B[15]*I[J]*V[J];
      N7:=B[16]*P[J]*V[J]+B[17]*U[J]*K[J]+B[18]*U[J]*Q[J];
      R0:=B[19]*I[J]*P[J]*V[J]+B[20]*I[J]*U[J]*Q[J];
      R4:=B[21]*P[J]*U[J]*K[J]+B[22]*K[J]*V[J]+B[23]*Q[J]*V[J];
      R5:=B[24]*I[J]*Q[J]*V[J]+B[25]*P[J]*K[J]*V[J];
      R6:=B[26]*U[J]*K[J]*Q[J]+B[27]*K[J]*Q[J]*V[J];
      Z[J]:=N3+N4+N5+N6+N7+R0+R4+R5+R6;
    end;
  end;{GB4770}
end.
unit TPG3_3;
Interface
uses tpg3_2;
var
  grDriver: Integer;
  grMode: Integer;
procedure PR_MOD; procedure GB4880;
procedure GB7000; procedure VVOD31;
procedure VVOD32; procedure VVOD33;
procedure VVOD41; procedure VVOD42;
procedure VVOD51; procedure VVOD52;
procedure tablF9; procedure OUT_F_H_L(PR:integer);
Implementation
procedure VVOD31;
begin
  writeln('Ввод A1 E1 B1 J1 O1');
  readln(A1,E1,B1,J1,O1);
  writeln(F0,'A1=',A1,' E1=',E1,' B1=',B1);
  writeln(F0,'J1=',J1,' O1=',O1);
end;{VVOD31}
procedure VVOD32;
begin
  writeln('Ввод A2 E2 B2 J2 O2');
  readln(A2,E2,B2,J2,O2);
  writeln(F0,'A2=',A2,' E2=',E2,' B2=',B2);
  writeln(F0,'J2=',J2,' O2=',O2);
end;{VVOD32}
procedure VVOD33;
begin
  writeln('Ввод A3 E3 B3 J3 O3');
  readln(A3,E3,B3,J3,O3);
  writeln(F0,'A3=',A3,' E3=',E3,' B3=',B3);
  writeln(F0,'J3=',J3,' O3=',O3);
end;{VVOD33}
procedure VVOD41;
begin
  writeln('Ввод A1 C1 D1 B1 J1 O1 P1');
  readln(A1,C1,D1,B1,J1,O1,P1);
  writeln(F0,'A1=',A1,' C1=',C1,' D1=',D1);
  writeln(F0,'B1=',B1,' J1=',J1,' O1=',O1);
  writeln(F0,'P1=',P1);
end;{VVOD41}
procedure VVOD42;
begin
  writeln('Ввод A2 C2 D2 B2 J2 O2 P2');
  readln(A2,C2,D2,B2,J2,O2,P2);
  writeln(F0,'A2=',A2,' C2=',C2,' D2=',D2);

```



```

        writeln(F0,'B2=',B2,' J2=',J2,' O2=',O2);
        writeln(F0,'P2=',P2);
    end;{VVD42}
procedure VVD51;
begin
    writeln('Ввод A1 C1 E1 D1 B1 J1 O1 P1 T1');
    readln(A1,C1,E1,D1,B1,J1,O1,P1,T1);
    writeln(F0,'A1=',A1,' C1=',C1,' E1=',E1);
    writeln(F0,'D1=',D1,' B1=',B1,' J1=',J1);
    writeln(F0,'O1=',O1,' P1=',P1,' T1=',T1);
end;{VVD51}
procedure VVD52;
begin
    writeln('Ввод A2 C2 E2 D2 B2 J2 O2 P2 T2');
    readln(A2,C2,E2,D2,B2,J2,O2,P2,T2);
    writeln(F0,'A2=',A2,' C2=',C2,' E2=',E2);
    writeln(F0,'D2=',D2,' B2=',B2,' J2=',J2);
    writeln(F0,'O2=',O2,' P2=',P2,' T2=',T2);
end;{VVD52}
procedure PR_MOD;
begin
    case X of
    3: begin
        F[1]:=A1; F[2]:=B1; F[3]:=E1;
        end;
    4: begin
        F[1]:=A1; F[2]:=B1; F[3]:=C1; F[4]:=D1;
        end;
    5: begin
        F[1]:=A1; F[2]:=B1; F[3]:=C1; F[4]:=D1; F[5]:=E1;
        end;
    9: begin
        F[1]:=A1; H[1]:=A2; F[2]:=B1; H[2]:=A2; F[3]:=A1;
        H[3]:=B2; F[4]:=B1; H[4]:=B2; F[5]:=A1; H[5]:=E2;
        F[6]:=B1; H[6]:=E2; F[7]:=E1; H[7]:=A2; F[8]:=E1;
        H[8]:=B2; F[9]:=E1; H[9]:=E2;
        end;
    12:begin
        F[1]:=A1; H[1]:=A2; F[2]:=B1; H[2]:=A2; F[3]:=A1;
        H[3]:=B2; F[4]:=B1; H[4]:=B2; F[5]:=E1; H[5]:=A2;
        F[6]:=E1; H[6]:=B2; F[7]:=A1; H[7]:=C2; F[8]:=B1;
        H[8]:=D2; F[9]:=A1; H[9]:=D2; F[10]:=B1; H[10]:=C2;
        F[11]:=E1; H[11]:=C2; F[12]:=E1; H[12]:=D2;
        end;
    15:begin
        F[1]:=A1; H[1]:=A2; F[2]:=B1; H[2]:=A2; F[3]:=A1;
        H[3]:=B2; F[4]:=B1; H[4]:=B2; F[5]:=A1; H[5]:=E2;
        F[6]:=B1; H[6]:=E2; F[7]:=E1; H[7]:=A2; F[8]:=E1;
        H[8]:=B2; F[9]:=E1; H[9]:=E2; F[10]:=A1; H[10]:=C2;
        F[11]:=B1; H[11]:=D2; F[12]:=A1; H[12]:=D2; F[13]:=B1;
        H[13]:=C2; F[14]:=E1; H[14]:=C2; F[15]:=E1; H[15]:=D2;
        end;
    16:begin
        F[1]:=A1; H[1]:=A2; F[2]:=B1; H[2]:=A2; F[3]:=A1;
        H[3]:=B2; F[4]:=B1; H[4]:=B2; F[5]:=A1; H[5]:=C2;
        F[6]:=B1; H[6]:=C2; F[7]:=A1; H[7]:=D2; F[8]:=B1;
        H[8]:=D2; F[9]:=C1; H[9]:=A2; F[10]:=C1; H[10]:=C2;
        F[11]:=C1; H[11]:=D2; F[12]:=C1; H[12]:=B2; F[13]:=D1;
        H[13]:=A2; F[14]:=D1; H[14]:=C2; F[15]:=D1; H[15]:=D2;
        F[16]:=D1; H[16]:=B2;
        end;
    20:begin
        F[1]:=A1; H[1]:=A2; F[2]:=B1; H[2]:=A2; F[3]:=A1;

```

```

H[3]:=B2; F[4]:=B1; H[4]:=B2; F[5]:=A1; H[5]:=E2;
F[6]:=B1; H[6]:=E2; F[7]:=A1; H[7]:=C2; F[8]:=B1;
H[8]:=D2; F[9]:=A1; H[9]:=D2; F[10]:=B1; H[10]:=C2;
F[11]:=C1; H[11]:=A2; F[12]:=C1; H[12]:=C2; F[13]:=C1;
H[13]:=E2; F[14]:=C1; H[14]:=D2; F[15]:=C1; H[15]:=B2;
F[16]:=D1; H[16]:=A2; F[17]:=D1; H[17]:=C2; F[18]:=D1;
H[18]:=E2; F[19]:=D1; H[19]:=D2; F[20]:=D1; H[20]:=B2;
end;
25:begin
F[1]:=A1; H[1]:=A2; F[2]:=B1; H[2]:=A2; F[3]:=A1;
H[3]:=B2; F[4]:=B1; H[4]:=B2; F[5]:=A1; H[5]:=E2;
F[6]:=B1; H[6]:=E2; F[7]:=E1; H[7]:=A2; F[8]:=E1;
H[8]:=B2; F[9]:=E1; H[9]:=E2; F[10]:=A1; H[10]:=C2;
F[11]:=B1; H[11]:=D2; F[12]:=A1; H[12]:=D2; F[13]:=B1;
H[13]:=C2; F[14]:=E1; H[14]:=C2; F[15]:=E1; H[15]:=D2;
F[16]:=C1; H[16]:=A2; F[17]:=C1; H[17]:=C2; F[18]:=C1;
H[18]:=E2; F[19]:=C1; H[19]:=D2; F[20]:=C1; H[20]:=B2;
F[21]:=D1; H[21]:=A2; F[22]:=D1; H[22]:=C2; F[23]:=D1;
H[23]:=E2; F[24]:=D1; H[24]:=D2; F[25]:=D1; H[25]:=B2;
end;
27:begin
F[1]:=A1; H[1]:=A2; L[1]:=A3; F[2]:=B1; H[2]:=A2;
L[2]:=A3; F[3]:=A1; H[3]:=B2; L[3]:=A3; F[4]:=B1;
H[4]:=B2; L[4]:=A3; F[5]:=A1; H[5]:=A2; L[5]:=B3;
F[6]:=B1; H[6]:=A2; L[6]:=B3; F[7]:=A1; H[7]:=B2;
L[7]:=B3; F[8]:=B1; H[8]:=B2; L[8]:=B3; F[9]:=A1;
H[9]:=E2; L[9]:=E3; F[10]:=B1; H[10]:=E2; L[10]:=E3;
F[11]:=E1; H[11]:=A2; L[11]:=E3; F[12]:=E1; H[12]:=B2;
L[12]:=E3; F[13]:=E1; H[13]:=E2; L[13]:=A3; F[14]:=E1;
H[14]:=E2; L[14]:=B3; F[15]:=A1; H[15]:=A2; L[15]:=E3;
F[16]:=B1; H[16]:=A2; L[16]:=E3; F[17]:=A1; H[17]:=B2;
L[17]:=E3; F[18]:=B1; H[18]:=B2; L[18]:=E3; F[19]:=A1;
H[19]:=E2; L[19]:=A3; F[20]:=B1; H[20]:=E2; L[20]:=A3;
F[21]:=A1; H[21]:=E2; L[21]:=B3; F[22]:=B1; H[22]:=E2;
L[22]:=B3; F[23]:=E1; H[23]:=A2; L[23]:=A3; F[24]:=E1;
H[24]:=B2; L[24]:=A3; F[25]:=E1; H[25]:=A2; L[25]:=B3;
F[26]:=E1; H[26]:=B2; L[26]:=B3; F[27]:=E1; H[27]:=E2;
L[27]:=E3;
end;
end;
end;
procedure GB4880;
begin
writeln(F0,'Математическая модель');
if X=3 then
writeln(F0,'Z(J)=' ,B[1]:10,'+',B[2]:10,'*I(J)+' ,B[3]:10,'*K(J),');
if X=9 then
begin
writeln(F0,'Z(J)=' ,B[1]:10,'+',B[2]:10,'*I(J)+' ,B[3]:10,'*K(J)+');
writeln(F0,'+',B[4]:10,'*P(J)+' ,B[5]:10,'*I(J)*P(J)+');
writeln(F0,'+',B[6]:10,'*Q(J)+' ,B[7]:10,'*I(J)*Q(J)+');
writeln(F0,'+',B[8]:10,'*P(J)*K(J)+' ,B[9]:10,'*K(J)*Q(J),');
end;
if X=27 then
begin
writeln(F0,'Z(J)=' ,B[1]:10,'+',B[2]:10,'*I(J)+' ,B[3]:10,'*K(J)+');
writeln(F0,'+',B[4]:10,'*P(J)+' ,B[5]:10,'*I(J)*P(J)+');
writeln(F0,'+',B[6]:10,'*Q(J)+' ,B[7]:10,'*I(J)*Q(J)+');
writeln(F0,'+',B[8]:10,'*P(J)*K(J)+' ,B[9]:10,'*K(J)*Q(J)+');
writeln(F0,'+',B[10]:10,'*U(J)+' ,B[11]:10,'*I(J)*U(J)+');
writeln(F0,'+',B[12]:10,'*P(J)*U(J)+' ,B[13]:10,'*I(J)*P(J)*U(J)+');
writeln(F0,'+',B[14]:10,'*V(J)+' ,B[15]:10,'*I(J)*V(J)+');
writeln(F0,'+',B[16]:10,'*P(J)*V(J)+' ,B[17]:10,'*U(J)*K(J)+');

```

```

writeln(F0,'+',B[18]:10,'*U(J)*Q(J)+' ,B[19]:10,'*I(J)*P(J)*V(J)+' );
writeln(F0,'+',B[20]:10,'*I(J)*U(J)*Q(J)+' ,B[21]:10,'*P(J)*U(J)*K(J)+' );
writeln(F0,'+',B[22]:10,'*K(J)*V(J)+' ,B[23]:10,'*Q(J)*V(J)+' );
writeln(F0,'+',B[24]:10,'*I(J)*Q(J)*V(J)+' ,B[25]:10,'*P(J)*K(J)*V(J)+' );
writeln(F0,'+',B[26]:10,'*U(J)*K(J)*Q(J)+' ,B[27]:10,'*K(J)*Q(J)*V(J),');
end;
if (X=3) or (X=27) or (X=9) then
begin
writeln(F0,'ГДЕ');
writeln(F0,'I(J)=F(J)^(J1:10,'+',V1:10,',');
writeln(F0,'K(J)=F(J)^(O1:10,'+',U1:10,'*F(J)^(J1:10,'+',Q1:10);
end;
if (X=9) or (X=27) then
begin
writeln(F0,'P(J)=H(J)^(J2:10,'+',V2:10,',');
writeln(F0,'Q(J)=H(J)^(O2:10,'+',U2:10,'*H(J)^(J2:10,'+',Q2:10);
end;
if X=27 then
begin
writeln(F0,'U(J)=L(J)^(J3:10,'+',V3:10,',');
write(F0,'V(J)=L(J)^(O3:10,'+',U3:10,'*L(J)^(J3:10,'+',Q3:10);
end;
if X=4 then
begin
writeln(F0,'Z(J)=' ,B[1]:10,'+',B[2]:10,'*I(J)+' ,B[3]:10,'*K(J)+' );
writeln(F0,'+',B[4]:10,'*L(J),');
end;
if X=5 then
begin
writeln(F0,'Z(J)=' ,B[1]:10,'+',B[2]:10,'*I(J)+' ,B[3]:10,'*K(J)+' );
writeln(F0,'+',B[4]:10,'*L(J)+' ,B[5]:10,'*M(J),');
end;
if X=12 then
begin
writeln(F0,'Z(J)=' ,B[1]:10,'+',B[2]:10,'*I(J)+' ,B[3]:10,'*K(J)+' );
writeln(F0,'+',B[4]:10,'*P(J)+' ,B[5]:10,'*I(J)*P(J)+' );
writeln(F0,'+',B[6]:10,'*Q(J)+' ,B[7]:10,'*I(J)*Q(J)+' );
writeln(F0,'+',B[8]:10,'*P(J)*K(J)+' ,B[9]:10,'*K(J)*Q(J)+' );
writeln(F0,'+',B[10]:10,'*U(J)+' ,B[11]:10,'*I(J)*U(J)+' );
writeln(F0,'+',B[12]:10,'*K(J)*U(J),');
end;
if X=15 then
begin
writeln(F0,'Z(J)=' ,B[1]:10,'+',B[2]:10,'*I(J)+' ,B[3]:10,'*K(J)+' );
writeln(F0,'+',B[4]:10,'*P(J)+' ,B[5]:10,'*I(J)*P(J)+' );
writeln(F0,'+',B[6]:10,'*Q(J)+' ,B[7]:10,'*I(J)*Q(J)+' );
writeln(F0,'+',B[8]:10,'*P(J)*K(J)+' ,B[9]:10,'*K(J)*Q(J)+' );
writeln(F0,'+',B[10]:10,'*U(J)+' ,B[11]:10,'*I(J)*U(J)+' );
writeln(F0,'+',B[12]:10,'*K(J)*U(J)+' ,B[13]:10,'*V(J)+' );
writeln(F0,'+',B[14]:10,'*I(J)*V(J)+' ,B[15]:10,'*K(J)*V(J),');
end;
if X=16 then
begin
writeln(F0,'Z(J)=' ,B[1]:10,'+',B[2]:10,'*I(J)+' ,B[3]:10,'*K(J)+' );
writeln(F0,'+',B[4]:10,'*P(J)+' ,B[5]:10,'I(J)*P(J)+' );
writeln(F0,'+',B[6]:10,'*Q(J)+' ,B[7]:10,'*I(J)*Q(J)+' );
writeln(F0,'+',B[8]:10,'*P(J)*K(J)+' ,B[9]:10,'*K(J)*Q(J)+' );
writeln(F0,'+',B[10]:10,'*U(J)+' ,B[11]:10,'*I(J)*U(J)+' );
writeln(F0,'+',B[12]:10,'*K(J)*U(J)+' ,B[13]:10,'*L(J)+' );
writeln(F0,'+',B[14]:10,'*P(J)*L(J)+' ,B[15]:10,'*Q(J)*L(J)+' );
writeln(F0,'+',B[16]:10,'*L(J)*U(J),');
end;
end;

```

```

if X=20 then
begin
writeln(F0,'Z(J)=' ,B[1]:10,'+',B[2]:10,'*I(J)+' ,B[3]:10,'*K(J)+' );
writeln(F0,'+',B[4]:10,'*P(J)+' ,B[5]:10,'*I(J)*P(J)+' );
writeln(F0,'+',B[6]:10,'*Q(J)+' ,B[7]:10,'*I(J)+Q(J)+' );
writeln(F0,'+',B[8]:10,'*P(J)*K(J)+' ,B[9]:10,'*K(J)*Q(J)+' );
writeln(F0,'+',B[10]:10,'*U(J)+' ,B[11]:10,'*I(J)*U(J)+' );
writeln(F0,'+',B[12]:10,'*K(J)*U(J)+' ,B[13]:10,'*V(J)+' );
writeln(F0,'+',B[14]:10,'*I(J)*V(J)+' ,B[15]:10,'*I(J)*V(J)+' );
writeln(F0,'+',B[16]:10,'*L(J)+' ,B[17]:10,'*P(J)*L(J)+' );
writeln(F0,'+',B[18]:10,'*Q(J)*L(J)+' ,B[19]:10,'*L(J)*U(J)+' );
writeln(F0,'+',B[20]:10,'*L(J)*V(J),');
end;
if X=25 then
begin
writeln(F0,'Z(J)=' ,B[1]:10,'+',B[2]:10,'*I(J)+' ,B[3]:10,'*K(J)+' );
writeln(F0,'+',B[4]:10,'*P(J)+' ,B[5]:10,'*I(J)*P(J)+' );
writeln(F0,'+',B[6]:10,'*Q(J)+' ,B[7]:10,'*I(J)*Q(J)+' );
writeln(F0,'+',B[8]:10,'*P(J)*Q(J)+' ,B[9]:10,'*K(J)*Q(J)+' );
writeln(F0,'+',B[10]:10,'*U(J)+' ,B[11]:10,'*I(J)*U(J)+' );
writeln(F0,'+',B[12]:10,'*K(J)*U(J)+' ,B[13]:10,'*V(J)+' );
writeln(F0,'+',B[14]:10,'*I(J)*V(J)+' ,B[15]:10,'*I(J)*V(J)+' );
writeln(F0,'+',B[16]:10,'*L(J)+' ,B[17]:10,'*P(J)*L(J)+' );
writeln(F0,'+',B[18]:10,'*Q(J)*L(J)+' ,B[19]:10,'*L(J)*U(J)+' );
writeln(F0,'+',B[20]:10,'*L(J)*V(J)+' ,B[21]:10,'*M(J)+' );
writeln(F0,'+',B[22]:10,'*P(J)*M(J)+' ,B[23]:10,'*Q(J)*M(J)+' );
writeln(F0,'+',B[24]:10,'*U(J)*M(J)+' ,B[25]:10,'*M(J)*V(J),');
end;
if (X=4)or(X=5)or(X=12)or(X=15)or(X=16)or
(X=20)or(X=25) then
begin
writeln(F0,'ГДЕ');
writeln(F0,'I(J)=F(J)^( ,J1:10,'+',V1:10,',');
writeln(F0,'K(J)=F(J)^( ,O1:10,'+',U1:10,'*F(J)^( ,J1:10,'+',Q1:10);
end;
if (X=4)or(X=5)or(X=16)or(X=20)or(X=25)then
begin
writeln(F0,'L(J)=F(J)^( ,P1:10,'+',I1:10,'*F(J)^( ,O1:10,'+',');
writeln(F0,'+',M1:10,'F(J)^( ,J1:10,'+',F1:10);
end;
if (X=5)or(X=25) then
begin
writeln(F0,'M(J)=F(J)^( ,T1:10,'+',G1:10,'*F(J)^( ,P1:10,'+',');
writeln(F0,'+',H1:10,'*F(J)^( ,O1:10,'+',K1:10,'*F(J)^( ,I1:10,'+',L1:10);
end;
if (X=12)or(X=15) or (X=16) or (X=20) or (X=25) then
begin
writeln(F0,'P(J)=H(J)^( ,J2:10,'+',V2:10,',');
writeln(F0,'Q(J)=H(J)^( ,O2:10,'+',U2:10,'*H(J)^( ,J2:10,'+',Q2:10,',');
writeln(F0,'U(J)=H(J)^( ,P2:10,'+',I2:10,'*H(J)^( ,O2:10,'+',');
writeln(F0,'+',M2:10,'*H(J)^( ,J2:10,'+',F2:10);
end;
if (X=12) or (X=15) or (X=20) or (X=25) then
begin
writeln(F0,'V(J)=H(J)^( ,T2:10,'+',G2:10,'*H(J)^( ,P2:10,'+',');
writeln(F0,'+',H2:10,'*H(J)^( ,O2:10,'+',K2:10,'*H(J)^( ,J2:10,'+',');
writeln(F0,'+',L2:10);
end;
end;{GB4880}
procedure GB7000;
label 1;
begin
repeat

```

```

if (X=3)or(X=4)or(X=5) then I0:=61;
if (X=9)or(X=12)or(X=15)or(X=16)or(X=20)or(X=25) then I0:=62;
if X=27 then I0:=63;
if I0=61 then
  begin
    case X of
      3: I0:=73;
      4: I0:=74;
      5: I0:=75;
    end;
  F3:=0; F4:=0; K5:=0;
  writeln('Фактор F(1)=F3+F4');
  for J:=1 to X do
    begin
      F[J]:=0; Z[J]:=0;
    end;
  writeln('F4-шаг приращения фактора');
  writeln('X-количество значений фактора');
  writeln('Ввод принятых величин X F3 F4');
  readln(X,F3,F4);
  writeln(F0,'F4-шаг приращения фактора');
  writeln(F0,'X-количество значений фактора');
  writeln(F0,'X=',X,' F3=',F3,' F4=',F4);
  for K5:=1 to X do
    begin
      F[K5]:=F3+K5*F4;
      writeln('F(',K5,')=',F[K5]);
    end; {for}
  case I0 of
    73:begin
      GB4150; GB4390;
    end;
    74:begin
      GB4210; GB4400;
    end;
    75:begin
      GB4290; GB4420;
    end;
  end; {case}
  for K5:=1 to X do
    writeln('Z(',K5,')=',Z[K5]);
  OUT_F_H_L(73);
  end; {if}
if I0=62 then
  begin
    case X of
      9: I0:=76;
      12:I0:=77;
      15:I0:=78;
      16:I0:=79;
      20:I0:=80;
      25:I0:=81;
    end;
  F3:=0; F4:=0; H3:=0;
  H4:=0; K5:=0;
  writeln('Фактор F(1)=F3+F4');
  writeln(F0,'Фактор F(1)=F3+F4');
  for J:=1 to X do
    begin
      F[J]:=0; H[J]:=0; Z[J]:=0;
    end;
  writeln('F4-шаг приращения 1-го фактора');
  writeln('Фактор H(1)=H3+H4');

```

```

writeln('H4-шаг приращения 2-го фактора');
writeln('X-количество значений 1,2-го фактора');
writeln('Ввод принятых величин X F3 F4 H3 H4');
writeln(F0,'F4-шаг приращения 1-го фактора');
writeln(F0,'Фактор H(1)=H3+H4');
writeln(F0,'H4-шаг приращения 2-го фактора');
writeln(F0,'X-количество значений 1,2-го фактора');
readln(X,F3,F4,H3,H4);
writeln(F0,'X=',X,' F3=',F3,' F4=',F4);
writeln(F0,'H3=',H3,' H4=',H4);
for K5:=1 to X do
begin
  F[K5]:=F3+K5*F4;
  writeln('F(',K5,')=',F[K5]);
  H[K5]:=H3+K5*H4;
  writeln('H(',K5,')=',H[K5]);
  case I0 of
    76:begin
      GB4150; GB4170; GB4450;
    end;
    77:begin
      GB4150; GB4250; GB4490;
    end;
    78:begin
      GB4150; GB4340; GB4530;
    end;
    79:begin
      GB4210; GB4250; GB4580;
    end;
    80:begin
      GB4210; GB4340; GB4630;
    end;
    81:begin
      GB4290; GB4340; GB4690;
    end;
  end; {case}
  writeln('Z(',K5,')=',Z[K5]);
end; {for}
for K5:=1 to X do
  writeln('Z(',K5,')=',Z[K5]);
OUT_F_H_L(74);
end; {if}
if I0=63 then
begin
  K5:=0; F3:=0; F4:=0; H3:=0;
  H4:=0; L3:=0; L4:=0;
  writeln('Факторы F(1)=F3+F4, H(1)=H3+H4, L(1)=L3+L4');
  writeln(F0,'Факторы F(1)=F3+F4, H(1)=H3+H4, L(1)=L3+L4');
  for J:=1 to X do
    begin
      F[J]:=0; H[J]:=0; L[J]:=0; Z[J]:=0;
    end;
  X:=0;
  writeln('F4,H4,L4-шаг приращения 1,2,3-го фактора');
  writeln('X-количество значений 1,2,3-го фактора');
  writeln(F0,'F4,H4,L4-шаг приращения 1,2,3-го фактора');
  writeln(F0,'X-количество значений 1,2,3-го фактора');
  writeln('Ввод принятых величин');
  writeln('X,F3,F4,H3,H4,L3,L4');
  readln(X,F3,F4,H3,H4,L3,L4);
  writeln(F0,'X=',X,' F3=',F3,' F4=',F4);
  writeln(F0,'H3=',H3,' H4=',H4);
  writeln(F0,'L3=',L3,' L4=',L4);

```

```

for K5:=1 to X do
begin
F[K5]:=F3+K5*F4;
writeln('F(',K5,')=',F[K5]);
H[K5]:=H3+K5*H4;
writeln('H(',K5,')=',H[K5]);
L[K5]:=L3+K5*L4;
writeln('L(',K5,')=',L[K5]);
GB4150; GB4170;
GB4190; GB4770;
end; {for}
for K5:=1 to X do
writeln('Z(',K5,')=',Z[K5]);
OUT_F_H_L(75);
end; {if}
writeln('Выявление MAX Z(K5) и MIN Z(K5)');
writeln(F0,'Выявление MAX Z(K5) и MIN Z(K5)');
writeln('Ввод I0=90-продолжение');
K8:=0; K8:=Z[1];
readln(I0);
for K5:=1 to X do
if Z[K5]>=K8 then K8:=Z[K5];
writeln('MAX Z(K5)=',K8);
writeln(F0,'MAX Z(K5)=',K8);
for K5:=1 to X do
if Z[K5]=K8 then
begin
writeln('MAX Z(',K5,')=',Z[K5]);
writeln(F0,'MAX Z(',K5,')=',Z[K5]);
end;
K7:=0; K7:=Z[1];
for K5:=1 to X do
if Z[K5]<=K7 then K7:=Z[K5];
writeln('MIN Z(K5)=',K7);
writeln(F0,'MIN Z(K5)=',K7);
for K5:=1 to X do
if Z[K5]=K7 then
begin
writeln('MIN Z(',K5,')=',Z[K5]);
writeln(F0,'MIN Z(',K5,')=',Z[K5]);
end;
writeln('MIN Z(K5)=K7, MAX Z(K5)=K8');
writeln('K6(K5)=(Z(K5)+abs(K7))/(abs(K7)+abs(K8))');
writeln(F0,'MIN Z(K5)=K7, MAX Z(K5)=K8');
writeln(F0,'K6(K5)=(Z(K5)+abs(K7))/(abs(K7)+abs(K8))');
for K5:=1 to X do
begin
KK6[K5]:=(Z[K5]+abs(K7))/(abs(K7)+abs(K8));
writeln('K6(',K5,')=',KK6[K5]);
writeln(F0,'K6(',K5,')=',KK6[K5]);
end;
J5:=0;
J5:=abs(K7)/(abs(K7)+abs(K8));
writeln('J5:=abs(K7)/(abs(K7)+abs(K8))');
writeln('J5=',J5);
writeln(F0,'J5:=abs(K7)/(abs(K7)+abs(K8))');
writeln(F0,'J5=',J5);
writeln('Если I0=70, то повторение');
writeln('вычисления показателей');
writeln('и построение графиков');
writeln('Если I0=80, то построение графика');
readln(I0);
if I0=70 then GB7000;

```

```

1: K0:=0; K3:=0; K4:=0; K4:=X;
   K7:=0; K8:=0; X0:=0; Y0:=0;
   writeln('Построение графика');
   writeln('Зависимость КК6(К5) от фактора');
   writeln('КК6(К5)-относительная величина показателя');
   writeln('К5-номер величины фактора и показателя');
   writeln('Величина фактора заданы');
   writeln('Ввод X0-отступ вправо по оси X');
   writeln('(Предпочтительно X=5)');
   writeln('Y0-отступ вниз по оси Y');
   writeln('(Предпочтительно Y0=180)');
   writeln('K0-длина графика по оси X');
   writeln('K3-высота графика по оси Y');
   writeln('Было X0=',X0,' Y0=',Y0);
   writeln('Было K0=',K0,' K3=',K3);
   writeln('Введите X0 Y0 K0 K3');
   readln(X0,Y0,K0,K3);
   writeln('Ввод I0=75, то повторение построения графика');
writeln('Ввод I0=85, то полное повторение построения графика');
writeln('Ввод I0=95, то выход из программы');
write('I0=');
readln(I0);
if I0=75 then goto 1;
until not(I0=85);
writeln(F0,'Построение графика');
writeln(F0,'Зависимость КК6(К5) от фактора');
writeln(F0,'КК6(К5)-относительная величина показателя');
writeln(F0,'К5-номер величины фактора и показателя');
writeln(F0,'Величина фактора задана');
writeln(F0,'Ввод X0-отступ вправо по оси X');
writeln(F0,'(Предпочтительно X=5)');
writeln(F0,'Y0-отступ вниз по оси Y');
writeln(F0,'(Предпочтительно Y0=180)');
writeln(F0,'K0-длина графика по оси X');
writeln(F0,'K3-высота графика по оси Y');
writeln(F0,'X0=',X0,' Y0=',Y0);
writeln(F0,'K0=',K0,' K3=',K3);
writeln(F0,'где');
writeln(F0,'X0-отступ вправо по оси X');
writeln(F0,'Y0-отступ вниз по оси Y');
writeln(F0,'K0-длина графика по оси X');
writeln(F0,'K3-длина графика по оси Y');
end; {GB7000}
procedure tablF9;
begin
  write('          Значения F7 для 5% уровня');
  writeln('значимости');
  write(' _____');
  writeln(' _____');
  write(' |           F9           |');
  writeln(' |           |');
  write(' | F8 _____');
  writeln(' | _____');
  write(' | 2 | 3 | 4 | 8 | 11 | 14 |');
  writeln(' |15-16 | 19-20 | 24 | 26-30 |');
  write(' | _____');
  writeln(' | _____');
  write(' | 2 | 19.0 | 19.16 | 19.25 | 19.37 | 19.4 | 19.42 |');
  writeln(' |19.43 | 19.44 | 19.45 | 19.46 |');
  write(' | _____');
  writeln(' | _____');
  write(' | 3 | 9.55 | 9.28 | 9.12 | 8.84 | 8.76 | 8.71 |');
  writeln(' | 8.69 | 8.66 | 8.64 | 8.62 |');

```



```

write(' | | | | | | | | | | ');
writeln(' | | | | | | | | | | ');
write(' | 4 | 6.94 | 6.59 | 6.39 | 6.04 | 5.93 | 5.87 | ');
writeln(' | 5.84 | 5.8 | 5.77 | 5.74 | ');
write(' | | | | | | | | | | ');
writeln(' | | | | | | | | | | ');
write(' | 5 | 5.79 | 5.41 | 5.19 | 4.82 | 4.7 | 4.64 | ');
writeln(' | 4.6 | 4.56 | 4.53 | 4.5 | ');
write(' | | | | | | | | | | ');
writeln(' | | | | | | | | | | ');
write(' | 6 | 5.14 | 4.76 | 4.53 | 4.15 | 4.03 | 3.96 | ');
writeln(' | 3.92 | 3.87 | 3.84 | 3.81 | ');
write(' | | | | | | | | | | ');
writeln(' | | | | | | | | | | ');
end;
procedure OUT_F_H_L(PR:integer);
begin
if PR=73 then
begin
writeln(F0,' | | | | | | | | | | ');
writeln(F0,' | | | | | | | | | | ');
writeln(F0,' | J | | | | | | | | | | ');
writeln(F0,' | | | | | | | | | | ');
end;
if PR=74 then
begin
writeln(F0,' | | | | | | | | | | ');
writeln(F0,' | | | | | | | | | | ');
writeln(F0,' | J | | | | | | | | | | ');
writeln(F0,' | | | | | | | | | | ');
end;
if PR=75 then
begin
write(F0,' | | | | | | | | | | ');
writeln(F0,' | | | | | | | | | | ');
write(F0,' | | | | | | | | | | ');
writeln(F0,' | | | | | | | | | | ');
write(F0,' | J | | | | | | | | | | ');
writeln(F0,' | | | | | | | | | | ');
write(F0,' | | | | | | | | | | ');
writeln(F0,' | L(J) | | | | | | | | | | ');
end;
if PR=73 then
begin
for J:=1 to X do
begin
writeln(F0,' | | | | | | | | | | ');
writeln(F0,' | ',J:2,' | ',F[J]:10,' | ',Z[J]:10,' | ');
end;
writeln(F0,' | | | | | | | | | | ');
end;
if PR=74 then
begin
for J:=1 to X do
begin
write(F0,' | | | | | | | | | | ');
writeln(F0,' | | | | | | | | | | ');
write(F0,' | ',J:2,' | ',F[J]:10,' | ',H[J]:10);
writeln(F0,' | ',Z[J]:10,' | ');
end;
writeln(F0,' | | | | | | | | | | ');
end;
if PR=75 then

```

```

begin
  for J:=1 to X do
    begin
      write(F0, '|-----|-----|-----|');
      writeln(F0, '|-----|-----|-----|');
      write(F0, '|,J:2,|',F[J]:10,'|',H[J]:10,'|');
      writeln(F0,L[J]:10,'|',Z[J]:10,'|');
    end;
    write(F0, '|-----|-----|-----|');
    writeln(F0, '|-----|-----|-----|');
  end;
end;
end.

```

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Почему исходное уравнение для выявления математической модели выбрано в виде ряда (многочлена), почему оно называется уравнением регрессии, а его коэффициенты – коэффициентами регрессии?
2. В каких случаях факторы, влияющие на показатель процесса, считаются существенными, как производится выбор интервалов варьирования факторов?
3. Зачем выполняется регрессионный анализ?
4. Почему показатели степени факторов надо принимать буквенными?
5. В каких случаях матрица становится ортогональной, зачем надо делать матрицу ортогональной, от чего зависит количество коэффициентов ортогональности?
6. На основе чего и как выявляются коэффициенты ортогональности?
7. Можно ли определять коэффициенты регрессии независимо друг от друга, если матрица не будет ортогональной?
8. Почему рационально выполнять параллельные опыты на среднем уровне факторов, сколько надо проводить таких опытов, как определяется дисперсия опытов?
9. В чем преимущества независимого определения коэффициентов регрессии?
10. Почему дисперсия в определении коэффициентов регрессии рассчитываются независимо друг от друга, и как это делается?
11. Как определяют расчетные t-критерии, с чем их сравнивают, в каких случаях коэффициенты регрессии – значимые, а в каких – незначимые?
12. Зачем сравнивают введенные величины показателей с рассчитанными (по разностям и в процентах)?
13. О чем свидетельствует незначимость коэффициентов регрессии?
14. Как определяется адекватность и точность математической модели?
15. Как выявляются уравнения регрессии двухфакторного, трехфакторного, многофакторного процесса?
16. Почему совпадает количество опытов в плане и количество членов в уравнении регрессии?
17. Почему для каждого фактора отдельно выявляются коэффициенты ортогонализации?
18. Почему надо выполнять расчеты на ЭВМ с такой точностью, какую может обеспечить вычислительная машина?
19. В каких случаях рационально применять язык программирования Бейсик, а в каких – Турбо Паскаль?
20. Каков алгоритм математического моделирования, почему надо до рассмотрения компьютерных программ изучить язык программирования, можно ли не зная операторов языка программирования рассматривать и анализировать программы на этом языке?
21. Из каких частей состоят программы математического моделирования?

22. Почему расчеты по математическим моделям надо выполнять, используя общую программу математического моделирования?
23. Как выполняются расчеты по математическим моделям и графические построения?
24. Каковы преимущества представления результатов расчетов в абсолютных и относительных величинах, как выявляются максимальные и минимальные величины?
25. Почему выполнение программ надо заносить в файлы?
26. Можно ли оптимизировать, прогнозировать процессы, изобретать на основе моделирования?
27. Как выявляются факторы, существенно влияющие на показатели процесса, как можно уменьшить количество факторов, что дает применение комплексных факторов?
28. Почему надо изменять масштабы при графических построениях и что при этом достигается?
29. В каких случаях следует применять разные методы моделирования?
30. Какова эффективность моделирования, в чем заключаются преимущества изложенных выше методик математического моделирования?
31. Зачем в компьютерных программах предусмотрены различные переходы и можно ли их применять, если использовать не язык Бейсик, а другие языки программирования?
32. Что дает применение в компьютерных программах управляющей величины X ?
33. Чем отличается аппроксимация от математического моделирования, в каких случаях надо применять многократно аппроксимацию?
34. Какие части компьютерных программ относятся к аппроксимации, выявлению математической модели, выполнению расчетов по математической модели, поиску максимальных и минимальных величин показателей, графическому построению зависимости показателя от фактора?
35. Почему по программе строятся графики и как это выполняется?
36. Можно ли многократно изменять масштабы графических построений и если можно, то зачем это надо делать?
37. Почему для выбора показателей степени фактора в исходном уравнении надо несколько раз использовать часть компьютерной программы, которая предусматривает аппроксимацию и в каких случаях после рассмотрения результатов аппроксимации можно переходить к математическому моделированию?
38. Что дает использование аппроксимации в комплексных компьютерных программах, как проверяется точность полученных результатов аппроксимации, а затем и математических моделей?
39. Почему использование файлов упрощает компьютерные программы, как выполняется анализ результатов выполнения программ при рассмотрении файлов, можно ли из файлов исключить ненужные сведения и добавлять необходимые для разъяснения полученных данных?

40. Как достигается универсальность компьютерных программ?
41. Почему математическое моделирование позволяет выполнять фундаментальные научные исследования, какие результаты моделирования рационально вносить в научные отчеты и использовать при разработке изобретений?

ВОПРОСЫ ДЛЯ САМОПРОВЕРКИ

1. Какие известны по литературным источникам методы математического моделирования, их недостатки?
2. Как устранены недостатки существующих методов математического моделирования в изложенной выше разработке?
3. Что такое математическая модель, зачем надо ее выявлять и как ее анализировать?
4. Как производится выбор показателей процесса, существенных факторов, планов проведения экспериментов, как выполняются эксперименты для математического моделирования?
5. Почему для выявления математических моделей выбраны уравнения в виде рядов (многочленов), как называются эти уравнения и коэффициенты при каждом члене многочлена?
6. Как объяснить применение при математическом моделировании понятия регрессии?
7. Соответствует ли количество коэффициентов регрессии в уравнении регрессии количеству уровней фактора (для однофакторного процесса)?
8. В каких случаях матрицы определения коэффициентов регрессии становятся ортогональными и зачем надо добиваться ортогональности матриц?
9. Сколько надо определить коэффициентов ортогонализации, если принять два, три, четыре, пять уровней фактора (для однофакторного процесса)?
10. Почему нерационально применять больше пяти уровней фактора?
11. Равно ли количество членов многочлена и коэффициентов регрессии количеству опытов по плану проведения экспериментов (при полном факторном эксперименте)?
12. Почему показатели степени фактора в уравнении регрессии приняты буквенными?
13. Можно ли изменять величины показателей степени фактора при выявлении математических моделей и если можно, то в каких случаях, сколько раз, какие величины показателей степени рационально принимать первоначально и в последующем, что является критерием правильности выбора показателей степени фактора?
14. Как определяются коэффициенты регрессии при ортогональности матрицы?
15. Какие преимущества достигаются при определении коэффициентов регрессии независимо друг от друга?
16. По какому критерию выявляется статистическая значимость коэффициентов регрессии?
17. Как выявляется дисперсия опытов, почему лучше проводить серию параллельных одинаковых опытов на среднем уровне факторов, как определить средние уровни факторов, сколько надо выполнять одинаковых опытов на среднем уровне факторов?

18. По какой формуле выполняется расчет дисперсии опытов?
19. Почему дисперсии в определении коэффициентов регрессии рассчитываются независимо друг от друга, является ли это следствием ортогональности матриц?
20. По какому критерию проверяется адекватность математической модели?
21. Как оценивается фактическая точность математической модели?
22. Можно ли использовать для выявления математической модели комплексные факторы и факторы в виде зависимости одного фактора от другого или ряда других факторов, каковы особенности анализа математической модели при комплексных факторах?
23. Как выявляются уравнения регрессии при влиянии на показатель двух и трех факторов?
24. Почему рационально применять различные методы моделирования (моделирование на основе теории подобия, теории размерностей, математическое моделирование) и как следует выполнять в этом случае анализ результатов моделирования?
25. Что является критерием истины и как подтвердить истинность данных, рассчитанных по математическим моделям?
26. Какие особенности моделирования многофакторного процесса?
27. Каков алгоритм математического моделирования для программирования применительно к использованию ЭВМ?
28. В чем заключаются преимущества языка программирования Бейсик, какие операторы языка Бейсик использованы в разработанных программах?
29. Можно ли совершенствовать, оптимизировать, прогнозировать, автоматизировать процессы, разрабатывать изобретения на основе математических моделей?
30. Как достигается экономичность исследовательской работы при последующем математическом моделировании?
31. В чем заключается фундаментальность исследований и какое значение имеет математическое моделирование при выполнении таких исследований?
32. Необходимо ли применять математическое моделирование при выполнении научно-исследовательских, диссертационных работ, каковы могут быть направления дальнейшего совершенствования методики математического моделирования?
33. В каких случаях рационально разрабатывать программы математического моделирования на языке Турбо Паскаль?
34. О чем свидетельствует сравнение программ математического моделирования на языке Бейсик и Турбо Паскаль?
35. Можно ли программы математического моделирования, разработанные на языке Турбо Паскаль, применить для использования в среде визуального программирования под Windows-Delphi?

Л И Т Е Р А Т У Р А

1. Черный А.А. Математическое моделирование применительно к литейному производству: Учеб. пособие. – Пенза: Изд-во Пенз. гос. ун-та, 1998. – 121 с.
2. Черный А.А. Планирование экспериментов и математическое моделирование процессов. – Саратов: Изд-во Сарат. ун-та, 1977. – 80 с.
3. Смирнов Н.В. Курс теории вероятностей и математической статистики для технических приложений/ Н.В. Смирнов, И.В. Дунин-Барковский. – 2-е изд., испр. и доп. – М.: Наука. Гл.ред. физ.-мат. лит., 1965. – 512 с.
4. Математическое моделирование литейных процессов: Методические указания/ Сост. А.А.Черный. – Пенза: Подразделение оперативной полиграфии Пензенского ЦНТИ, 1992. – 36 с.
5. Моделирование сложных процессов по результатам экспериментов: Методические указания/ Сост. А.А.Черный. – Пенза: Пензенский политехнический институт, 1990. – 37 с.
6. Математическое моделирование процессов литейного производства и применение ЭВМ для их расчетов: Методические указания/ Сост. А.А.Черный. – Пенза: Пензенский политехнический институт, 1990. – 36 с.
7. Разработка новых сплавов с использованием ЭВМ: Методические указания/ Сост. А.А.Черный. – Пенза: Пензенский политехнический институт, 1990. – 28 с.
8. Черный А.А. Методика и программы математического моделирования: Учеб. пособие. – Пенза: Подразделение оперативной полиграфии Пензенского ЦНТИ, 1994. – 38 с.
9. Черный А.А. Практика планирования экспериментов и математического моделирования процессов. – Саратов: Изд-во Сарат. ун-та, 1984. – 103 с.
10. Новик Ф.С. Оптимизация процессов технологии металлов методом планирования экспериментов/ Ф.С. Новик, А.Б. Арсов. – М.: Машиностроение; София: техника, 1980. – 304 с.
11. Математическое моделирование в литейном производстве: рабочая программа и метод. указ. к практическим работам. / Сост. А.А. Черный. – Пенза: Изд-во Пенз.гос.ун-та, 2005. – 20 с.
12. Вычислительная техника в инженерных расчетах: рабочая программа и метод.указ. к лабораторным, практическим и курсовым работам. / Сост. А.А. Черный. – Пенза: Изд-во Пенз.гос.ун-та, 2005. – 39 с.
13. Задания по математическому моделированию в литейном производстве: метод.указ./ Сост. А.А. Черный. – Пенза: Изд-во Пенз.гос.ун-та, 2005. – 27 с.
14. Принципы инженерного творчества: рабочая программа и метод.указ. к практическим работам./ Сост. А.А. Черный. – Пенза: Изд-во Пенз.гос.ун-та, 2005. – 16 с.

15. Черный А.А. Математическое моделирование в литейном производстве: Учеб.пособие/А.А.Черный.-Пенза: Информационно-издательский центр ПГУ, 2007.-192с.

16. Алексеев Е.Р. Турбо Паскаль 7.0/Е.Р. Алексеев, О.В. Чеснокова-М.: НТ Пресс, 2009.-320с.

17. Столяров А.М. Microsoft Excel 2003/А.М. Столяров, Е.С. Столярова-М.: НТ Пресс, 2007.-191с.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
АЛГОРИТМ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ И РАСЧЕТОВ ПО МАТЕМАТИЧЕСКИМ МОДЕЛЯМ.....	4
ПЛАН ПРОВЕДЕНИЯ ЭКСПЕРИМЕНТОВ.....	5
ТАБЛИЦА СИСТЕМНОГО ПРЕДСТАВЛЕНИЯ РЕЗУЛЬТАТОВ РАСЧЕТОВ.....	6
ПРОГРАММА НА ЯЗЫКЕ БЕЙСИК NV6.....	7
ОПЕРАТОРЫ ЯЗЫКА ПРОГРАММИРОВАНИЯ БЕЙСИК.....	15
ОСОБЕННОСТИ ПРИМЕНЕНИЯ WINDOWS	30
ЗАДАНИЕ ДЛЯ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ.....	32
ВЫПОЛНЕНИЕ ПРОГРАММЫ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ.....	33
РАСПЕЧАТКА РЕЗУЛЬТАТОВ ВЫПОЛНЕНИЯ ПРОГРАММЫ VN6 МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ И РАСЧЕТОВ ПО МАТЕМАТИЧЕСКОЙ МОДЕЛИ.....	37
РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ПРОГРАММЫ VN6.....	38
АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ РАСЧЕТОВ.....	45
ПОСТРОЕНИЕ ГРАФИКОВ.....	48
РЕКОМЕНДАЦИИ ПО УЛУЧШЕНИЮ ПРОЦЕССА ПЛАВКИ В ГАЗОВОЙ ВАГРАНКЕ ПО РЕЗУЛЬТАТАМ МОДЕЛИРОВАНИЯ.....	51
ПРОГРАММА W4 на языке Турбо Паскаль (три модуля tpg3, X=3, X=9, X=27, _X=4_, X=5, X=12, X=15, _X=16_, X=20, X=25, система после «адекватно»).....	52
КОНТРОЛЬНЫЕ ВОПРОСЫ.....	83
ВОПРОСЫ ДЛЯ САМОПРОВЕРКИ.....	86
ЛИТЕРАТУРА.....	88

Черный Анатолий Алексеевич

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ С
ПРИМЕНЕНИЕМ ГРАФИЧЕСКИХ ПОСТРОЕНИЙ В EXCEL

Учебное пособие

Издательство Пензенского государственного университета.
440026, Пенза, Красная, 40.