

**О. Г. КРУПЕННИКОВ, С. И. РЯЗАНОВ,
Ю. В. ПСИГИН, Д. В. КРАВЧЕНКО**

**СБОРНИК ЗАДАНИЙ
ДЛЯ ПРАКТИЧЕСКИХ ЗАНЯТИЙ
ПО ИНФОРМАТИКЕ**

Ульяновск 2010

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Государственное образовательное учреждение высшего профессионального образования
УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

СБОРНИК ЗАДАНИЙ ДЛЯ ПРАКТИЧЕСКИХ ЗАНЯТИЙ ПО ИНФОРМАТИКЕ

Учебное пособие

Под редакцией профессора, кандидата технических наук
Е. А. Карева

Ульяновск
2010

УДК 681.3:519.68 (075)

ББК 22.18 я7

С 23

Рецензенты:

кафедра «Математическое моделирование технических систем» Ульяновского государственного университета;

д-р техн. наук, профессор, генеральный директор ООО «УНИТЭК» В. В. Елифанов

Авторы – О. Г. Крупенников, С. И. Рязанов, Ю. В. Псигин, Д. В. Кравченко

*Утверждено редакционно-издательским советом УлГТУ
в качестве учебного пособия*

Сборник заданий для практических занятий по информатике : учебное пособие / О. Г. Крупенников, С. И. Рязанов, Ю. В. Псигин, Д. В. Кравченко ; под ред. Е. А. Карева. – Ульяновск : УлГТУ, 2010. – 157 с.

ISBN 978-5-9795-0664-7

В настоящем сборнике содержатся основные теоретические положения и задания для выполнения двенадцати практических занятий в интегрированной среде FREE PASCAL по дисциплине «Информатика».

Содержание учебного пособия соответствует требованиям государственного образовательного стандарта и рабочим программам дисциплины «Информатика» для студентов, обучающихся по направлениям подготовки бакалавров (150900 – Технология, оборудование, и автоматизация машиностроительных производств; 15100 – Технологические машины и оборудование; 190100 – Наземные транспортные системы) и дипломированных специалистов по машиностроительным специальностям дневной формы обучения.

УДК 621.9.048.4(075)

ББК 22.18 я7

ISBN 978-5-9795-0664-7

© О. Г. Крупенников, С. И. Рязанов,
Ю. В. Псигин, Д. В. Кравченко, 2010

© Оформление. УлГТУ, 2010

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ.....	4
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №1. Расчет конструктивных параметров деталей машин	6
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №2. Расчет функциональных показателей технических систем и процессов по условиям	13
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №3. Решение проектных задач с инвариантным выбором	33
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №4. Аналитический расчет параметров технических систем при изменении одного фактора..	38
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №5. Аналитический расчет параметров технических систем при изменении нескольких факторов.....	47
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №6. Обработка массивов экспериментальных данных при исследовании технических систем...	54
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №7. Создание и работа с базами данных элементов технических систем	62
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №8. Создание и работа со множественными базами данных элементов технических систем...	73
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №9. Сохранение результатов расчетов параметров технических систем	79
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №10. Применение процедур и функций при расчетах параметров технических систем	86
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №11. Усовершенствованные варианты отображения результатов проектирования технических систем	93
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №12. Создание графических модулей элементов технических систем	98
ЗАКЛЮЧЕНИЕ	121
ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ	122
ОСНОВНЫЕ ИСПОЛЬЗОВАННЫЕ ПОНЯТИЯ	123
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	127
ПРИЛОЖЕНИЕ А. Работа в среде FREE PASCAL	128
ПРИЛОЖЕНИЕ Б. Создание программ на языке FREE PASCAL	134
...	
ПРИЛОЖЕНИЕ В. Общие указания к выполнению практических занятий	142
ПРИЛОЖЕНИЕ Г. Примеры разработки алгоритмов программ.....	144

ПРЕДИСЛОВИЕ

Настоящее учебное пособие предназначено для методического обеспечения выполнения государственного образовательного стандарта высшего профессионального образования (ГОС ВПО) по дисциплине «Информатика» при подготовке бакалавров – по направлению 150900 – Технология, оборудование и автоматизация машиностроительных производств; 15100 – Технологические машины и оборудование; 190100 – Наземные транспортные системы – и дипломированных специалистов по машиностроительным специальностям дневной формы обучения.

Выписка из ГОС ВПО по направлению 150900

Индекс	Наименование дисциплины и ее основные разделы
ЕН.Ф.02.	ИНФОРМАТИКА Понятие информации и информатики. Общая характеристика процессов сбора, передачи, обработки и накопления информации (данных). Технические и программные средства реализации информационных процессов. Модели решения функциональных и вычислительных задач. Алгоритмизация и программирование. Языки программирования высокого уровня. Базы данных. Программное обеспечение и технологии программирования. Локальные и глобальные сети ЭВМ. Основы защиты информации и сведений, составляющих государственную тайну. Методы защиты информации. Компьютерный практикум.

В учебном пособии, помимо общих указаний по работе в интегрированной среде FREE PASCAL, рассматриваются вопросы программирования простых и структурированных типов данных, процедур и функций, а также организации элементов линейных структур, ветвления и циклов.

В пособии изложены основные теоретические положения и задания по выполнению двенадцати практических занятий по дисциплине «Информатика», содержание которых обобщает многолетний опыт вычислительного практикума на машиностроительном факультете УлГТУ.

В соответствии с действующими учебными планами практикум по дисциплине «Информатика» выполняется в первом и втором семестрах и рассчитан соответственно на 16 часов и 32 часа (см. ниже).

Распределение видов и часов занятий по семестрам

Вид занятий	Количество часов в семестр			Всего
	1	2	3	
Лекции	20	16		36
Лабораторные работы	–	16		16
Практические занятия	16	16		32
Самостоятельная работа	60	60		120
Зачет (сем.)	1	–		–
Экзамен (сем.)	–	2		–
Итого часов:				204

Целью практикума по информатике является закрепление знаний, полученных студентами при изучении лекционного курса и привитие студентам умений и навыков автоматизации решения функциональных и вычислительных инженерных задач.

Каждый студент до выполнения практического занятия должен изучить настоящее учебное пособие, а также повторить материал, изложенный в соответствующих разделах (темах) лекционного курса «Информатика», по конспекту лекций и рекомендованной литературе. При изучении следует обратить внимание на особенности разработки программ по соответствующим темам (разделам) FREE PASCAL, а также на рекомендации по оформлению отчета и формулированию выводов по работе.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №1.

РАСЧЕТ КОНСТРУКТИВНЫХ ПАРАМЕТРОВ ДЕТАЛЕЙ МАШИН

1.1. Разработка программ линейной структуры

Программы, разрабатываемые студентами на данном практическом занятии, должны быть реализованы на основе алгоритмов линейной структуры. Такие программы представляют собой совокупность следующих друг за другом операторов. Определенность расположения операторов зависит от условий конкретной задачи и от субъективных предпочтений программиста.

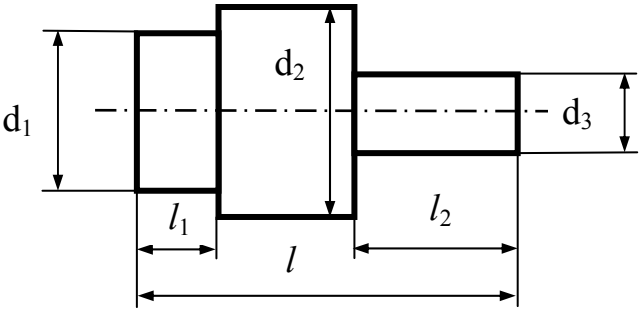
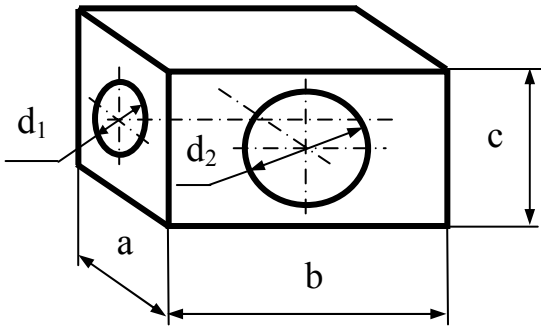
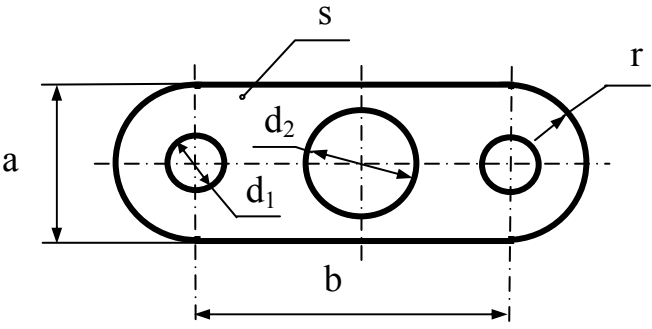
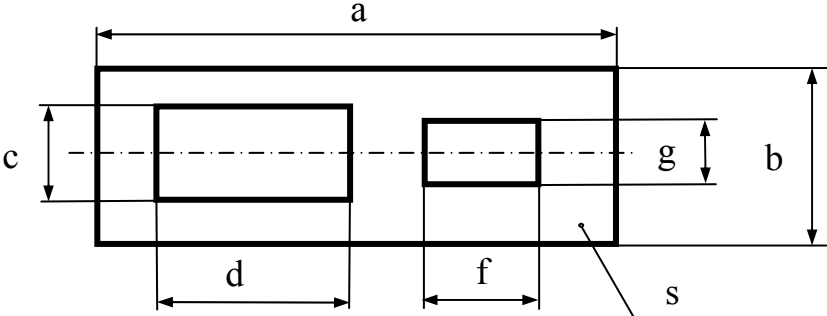
Пример 1.1. Составить программу для вычисления массы стальной заготовки длиной A , шириной B и высотой H . Алгоритм программы представлен в приложении Г.1.

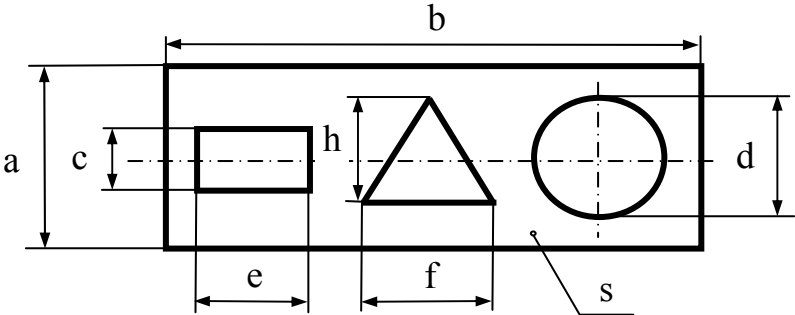
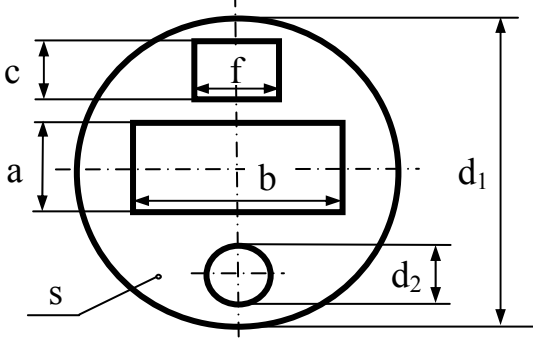
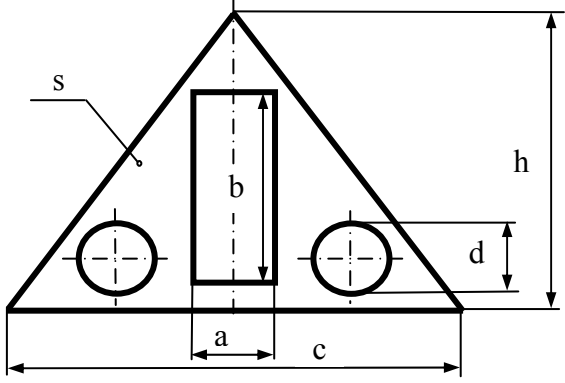
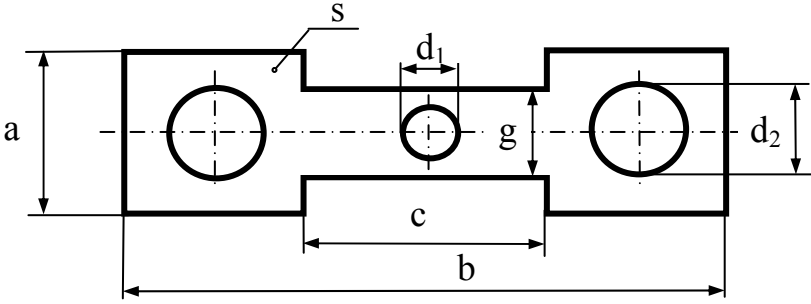
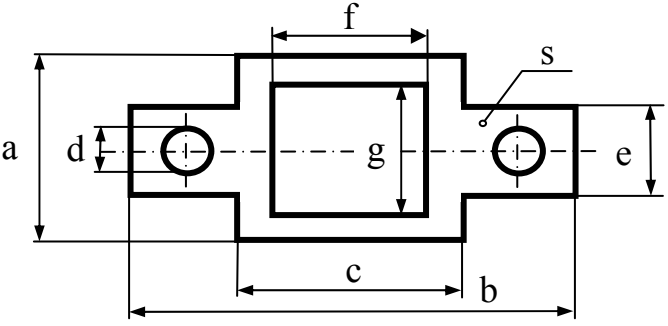
```
PROGRAM MASSA;
USES CRT;
    VAR MAS, A, B, H, G: REAL;
BEGIN
    CLRSCR;
    WRITE (' ВВЕДИТЕ ДЛИНУ ЗАГОТОВКИ В М : ');
    READLN (A);
    WRITE (' ВВЕДИТЕ ШИРИНУ ЗАГОТОВКИ В М : ');
    READLN (B);
    WRITE (' ВВЕДИТЕ ВЫСОТУ ЗАГОТОВКИ В М : ');
    READLN (H);
    WRITE (' ВВЕДИТЕ ПЛОТНОСТЬ ЗАГОТОВКИ В КГ/М^3 : ');
    READLN (G);
    MAS:= G*A*B*H/1000;
    WRITELN (' МАССА ЗАГОТОВКИ РАВНА ',MAS:4:2,' КГ ');
    READKEY;
END.
```

1.2. Задания к практическому занятию

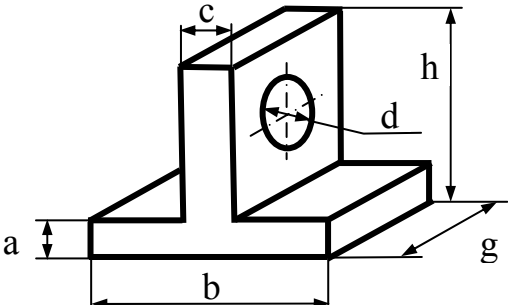
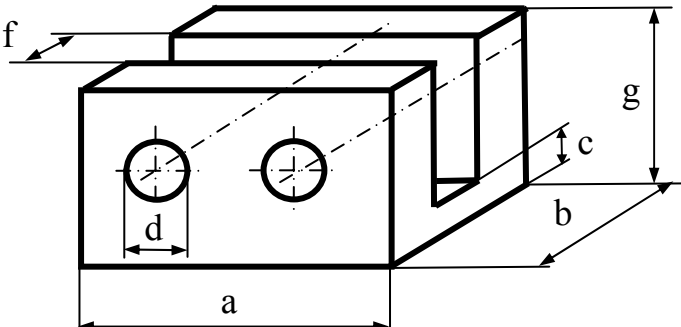
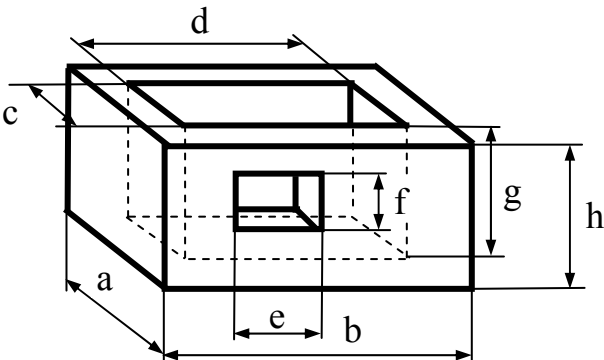
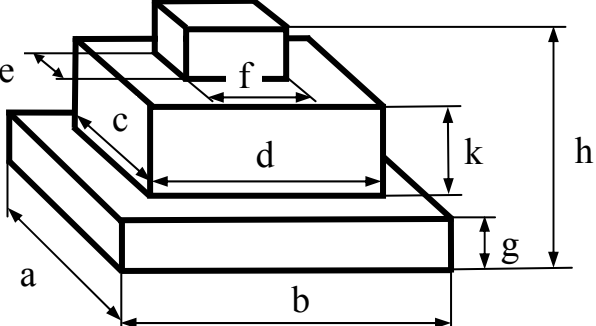
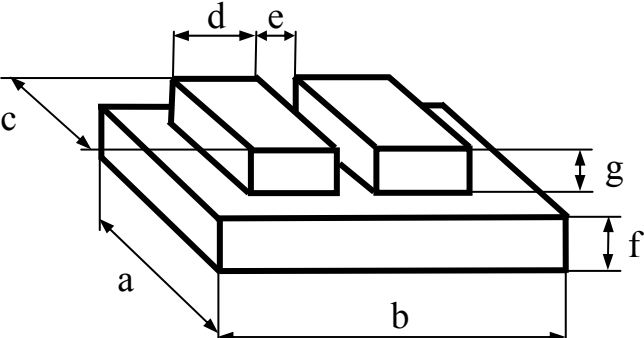
Согласно выданному преподавателем варианту задания составить программу для определения полной площади, объема и массы детали.

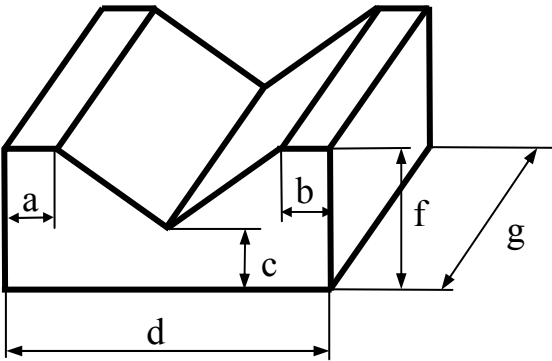
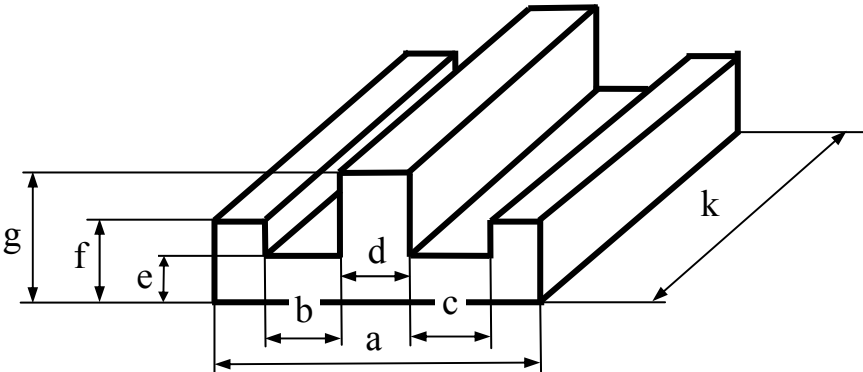
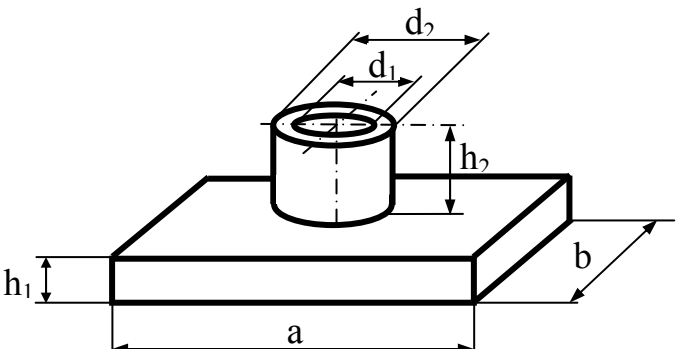
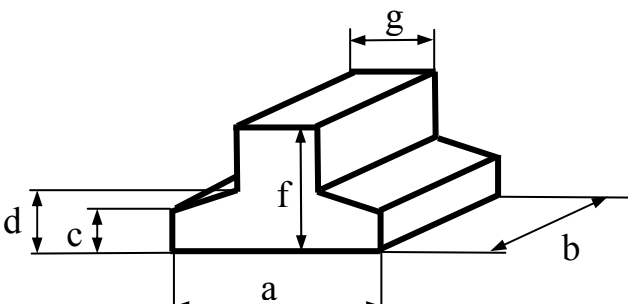
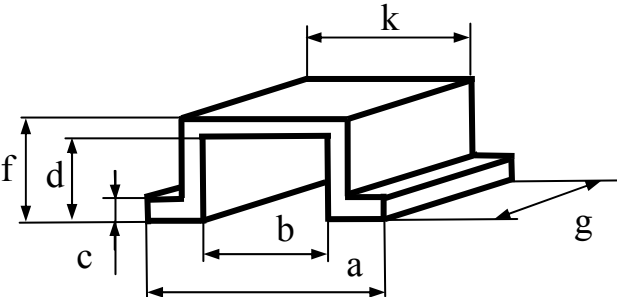
Таблица 1.1

№ варианта	Эскиз детали
1	2
1	
2	
3	
4	

1	2
5	
6	
7	
8	
9	

1	2
10	Technical drawing of a mechanical part (10). The part has a conical nose of length a and diameter d. The main body has a total length l. A central slot of length b and width c is located at a distance a from the nose. The slot has a rounded end with radius $r = c/2$. The right end of the part is rounded with a radius $R = d/2$.
11	Technical drawing of a mechanical part (11). The part has a total length l and an outer diameter d_1. It features a central hole with diameter d_2. On either side of the hole are rectangular slots of length a and width b. The distance between the inner edges of the slots is c, and the height of the slots is e. The ends of the part are rounded with a radius $r = d_1/2$.
12	Technical drawing of a mechanical part (12). The part consists of a hexagonal head of length a and diameter d_1, a central shaft of length b and diameter d_2, and a rectangular base of length c and diameter d_3. The total length of the part is l. The base has a width f and a height d_4.
13	Technical drawing of a mechanical part (13). The part has a total length l and an outer diameter d. It features a stepped profile with diameters d_1, d_2, d_3, and d_4. The lengths of the steps are a, b, and c.
14	Technical drawing of a mechanical part (14). The part has a total length l and an outer diameter d. It features a central rectangular block of length c and diameter d_2. On either side of the central block are side blocks of length a and b and diameter d_1. The distance between the inner edges of the side blocks is d_3, and the height of the side blocks is d_4.

1	2
15	
16	
17	
18	
19	

20	
21	
22	
23	
24	

1	2
25	
26	
27	
28	
29	
30	

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №2.

РАСЧЕТ ФУНКЦИОНАЛЬНЫХ ПОКАЗАТЕЛЕЙ ТЕХНИЧЕСКИХ СИСТЕМ И ПРОЦЕССОВ ПО УСЛОВИЯМ

2.1. Конструкции оператора IF

Условный оператор IF реализует алгоритмическую конструкцию «развилка» и изменяет порядок выполнения операторов в зависимости от истинности или ложности некоторого логического условия.

Существует два варианта оператора:

1. Полная развилка:

```
IF <ЛОГИЧ. УСЛОВИЕ> THEN <ОПЕРАТОР 1>  
                        ELSE <ОПЕРАТОР 2>;
```

2. Укороченная развилка:

```
IF <ЛОГИЧ. УСЛОВИЕ> THEN <ОПЕРАТОР 1>;
```

Так как условный оператор IF является единым предложением, то ни перед THEN, ни перед ELSE точку с запятой ставить нельзя.

Если в одной из ветвей развилки необходимо выполнить несколько операторов, то их следует заключать в операторные скобки BEGIN – END.

Пример 2.1. Составить программу, которая по содержанию углерода в железе выдавала бы название сплава (Fe – C).

```
PROGRAM Fe_C;  
USES CRT;  
VAR X:REAL;  
BEGIN  
  CLRSCR;  
  WRITELN ('ВВЕДИТЕ СОДЕРЖАНИЕ УГЛЕРОДА В %:');  
  READLN (X);  
  IF (X<=2.14) THEN WRITELN ('ЭТО – СТАЛЬ')  
                  ELSE WRITELN ('ЭТО – ЧУГУН');  
  READKEY  
END.
```

Многоальтернативный выбор можно осуществить также при помощи нескольких операторов IF:

```
IF <УСЛОВИЕ 1> THEN <ОПЕРАТОР 1>;  
IF <УСЛОВИЕ 2> THEN <ОПЕРАТОР 2>;  
.....  
IF <УСЛОВИЕ N THEN <ОПЕРАТОР N>;
```

Пример 2.2. Реализовать программу из примера 2.1 при помощи укороченных развилок IF. Алгоритм программы представлен в приложении Г.2.

```
PROGRAM Fe_C;  
USES CRT;  
VAR X:REAL;  
BEGIN  
  CLRSCR;  
  WRITELN ('ВВЕДИТЕ СОДЕРЖАНИЕ УГЛЕРОДА В %:');  
  READLN (X);  
  IF (X<=2.14) THEN WRITELN ('ЭТО – СТАЛЬ');  
  IF (X>2.14) THEN WRITELN ('ЭТО – ЧУГУН');  
  READKEY  
END.
```

Для существенного уменьшения числа проверок логических условий, более точной записи алгоритма программы, а иногда и упрощения условий, подлежащих проверке, применяют *вложенные конструкции оператора IF*:

```
IF <УСЛОВИЕ 1> THEN <ОПЕРАТОР 1>  
    ELSE IF <УСЛОВИЕ 2> THEN <ОПЕРАТОР 2>  
        ELSE <ОПЕРАТОР 3>; .
```

2.2. Задания к практическому занятию

1. Разработать программу для вычисления коэффициента Z_ε , учитывающего влияние перекрытия зубьев в сопряжении зубчатых колес:

$$Z_\varepsilon = \sqrt{(4 - \varepsilon_\alpha) / 3}$$

при $\beta = 0$;

$$Z_\varepsilon = \varepsilon_\alpha^{-0,5}$$

при $\beta = 10$ и $\varepsilon_\alpha = 2$;

$$Z_\varepsilon = \sqrt{(4 - \varepsilon_\alpha) \cdot (1 - \varepsilon_\beta) / 3 + \varepsilon_\beta / \varepsilon_\alpha}$$

при $\beta = 10$ и $\varepsilon_\alpha = 0,5$,

где β – угол наклона зубьев, град; ε_α – коэффициент торцового перекрытия; $\varepsilon_\beta = 0,8$ – коэффициент осевого перекрытия.

2. Разработать программу для вычисления диаметров начальных окружностей d_{w1} шестерни и d_{w2} колеса для случаев внешнего и внутреннего зацепления шестерни и колеса по приведенным ниже формулам. В зависимости от задач проектирования необходимо рассчитать диаметры только для случая внешнего зацепления (1 вариант) или внутреннего зацепления (2 вариант), что должно быть учтено в программе.

$$d_{w1} = 2a_w / (u \pm 1);$$

$$d_{w2} = 2a_w \cdot u / (u \pm 1),$$

где a_w – межосевое расстояние, мм; $u = z_2 / z_1$, где z_2 и z_1 – соответственно число зубьев колеса и шестерни; знаки $+$ и $-$ относятся соответственно к внешнему и внутреннему зацеплению.

3. Разработать программу для вычисления коэффициента Z_V , учитывающего влияние окружной скорости V шестерни на ее износ в зубчатой передаче, как показано в приведенных ниже формулах для трех значений твердости H_0 зубьев колес:

$$Z_V = 0,85V^{0,1}$$

при $H_0 = 320$;

$$Z_V = 0,85V^{0,1}$$

при $H_0 = 350$;

$$Z_V = 0,925V^{0,05}$$

при $H_0 = 370$,

где V – окружная скорость, м/мин; H_0 – твердость поверхности зубьев зубчатых колес.

4. Разработать программу расчета податливости λ_p витков резьбы коротких болтов, если материал болта – сталь 40Х (модуль упругости $E_B = 2,05 \times 10^5$ МПа), материал гайки – сталь 40ХН2МА (модуль упругости $E_{\Gamma} = 2,08 \times 10^5$ МПа), при этом соотношение наружного диаметра болта d и шага резьбы P может изменяться в диапазоне от 6 до 20. Возможность выбора вариантов варьирования указанного соотношения должна быть предусмотрена в программе.

$$\lambda_p = 0,95 * \frac{1}{Ed} \quad \text{при } \frac{d}{P} < 10;$$

$$\lambda_p = 0,8 * \frac{1}{Ed} \quad \text{при } \frac{d}{P} > 10,$$

где d – наружный диаметр болта, мм; P – шаг резьбы, мм;

E – модуль упругости материала: $\frac{1}{E} = \frac{1}{2} \cdot \left(\frac{1}{E_B} + \frac{1}{E_{\Gamma}} \right)$.

5. Разработать программу расчета податливости λ_p витков резьбы коротких болтов с наружным диаметром d по формуле: $\lambda_p = 0,80 \cdot \frac{1}{Ed}$, где E – модуль упругости материала, МПа.

В программе должна быть предусмотрена возможность расчета λ_p для различных значений модулей упругости E ($E = 2,05 \times 10^5$ МПа – для стали 40Х; $E = 1,95 \times 10^5$ МПа – для стали 40ХГСНА; $E = 1,7 \times 10^5$ МПа – для стали 12Х18Н9Т). Выбор значения E в программе должен происходить автоматически на основе введенных пользователем исходных данных о марке стали болта из приведенного выше перечня.

6. Разработать программу расчета поперечных колебаний f валов с различными вариантами размещения сосредоточенных масс m (т. е. деталей, размещенных на валу) и закрепления валов: консольное закрепление (1), на двух жестких опорах по концам вала (2) и «глухая» заделка с

одного конца и жесткая опора с другого конца (3). В зависимости от задач проектирования предполагается выбор из приведенного списка какого-либо одного из приведенных вариантов. В программе необходимо «заложить» расчеты по приведенным ниже формулам:

$$f = \frac{1}{2\pi} \sqrt{\frac{k}{m}},$$

где m – масса валов и насаженных на них деталей, кг; $k = \frac{3EJ}{\ell^3}$ – 1 вариант

крепления вала и установки деталей на валу; $k = \frac{48EJ}{\ell^3}$ – 2 вариант

крепления вала и установки деталей на валу; $k = \frac{12EJ}{3,5\ell^3}$ – 3 вариант кре-

пления вала и установки деталей на валу; E – модуль упругости материала, Па (принять $E=2,05 \times 10^5$ Па) ; J – осевой момент инерции сечения вала, м⁴; ℓ – длина вала, м.

7. Разработать программу для расчета статической грузоподъемности C_0 (допустимой нагрузки) радиальных шарикоподшипников (1), роликоподшипников (2) или упорно-радиальных шарикоподшипников (3), если для поименованных подшипников расчет C_0 ведут соответственно по формулам:

$$C_0 = 0,1 \cdot f_0 \cdot i \cdot z \cdot D^2 \cos \alpha; \quad (1)$$

$$C_0 = 0,22 \cdot i \cdot z \cdot \lambda \cdot D \cdot \cos \alpha; \quad (2)$$

$$C_0 = 0,55 \cdot z \cdot D^2 \cdot \sin \alpha, \quad (3)$$

где C_0 – статическая грузоподъемность, Н; f_0 – коэффициент (в расчетах принять $f_0 = 1,25$); i и z – соответственно число рядов и число тел качения в ряду; D – диаметр тел качения, мм; ℓ – эффективная длина роликов, мм; α – номинальный (начальный) угол контакта (в расчетах принять любое значение α из ряда $12^\circ, 26^\circ, 36^\circ$).

8. Разработать программу для расчета момента трения M_m на торце гайки для первой и десятой ее затяжки с учетом изменения среднего значения коэффициента трения на торце гайки f_t от затяжки к затяжке, а также от наличия и отсутствия смазки, если для материала 30ХГСА гайки (резьба М12×1.5) и шайбы коэффициент принимает следующие значения:

- первая затяжка: $f_t = 0,16$, без смазки;

$$f_t = 0,13, \text{ смазка – масло МК-8};$$

- десятая затяжка: $f_t = 0,22$, без смазки;

$$f_t = 0,11, \text{ смазка – масло МК-8}.$$

Расчет M_t выполнять по формуле

$$M_m = Q_0 \cdot f_t \cdot \frac{a + d_0}{2};$$

где Q_0 – усилие затяжки, Н; a – внешний диаметр опорного торца; d_0 – диаметр отверстия в корпусе под болт.

В зависимости от задач проектирования предполагается выбор вариантов затяжки из приведенного выше списка, что должно быть учтено в программе расчета M_m .

9. Разработать программу для расчета момента трения M_m на торце гайки для первой и десятой ее затяжки с учетом изменения среднего значения коэффициента трения на торце гайки f_t от затяжки к затяжке, а также от наличия или отсутствия покрытия гайки, если известно, что для материала ВТ6 гайки (резьба М16×1.5) и шайбы коэффициент f_t принимает следующие значения:

- первая затяжка: $f_t = 0,16$ (гайка без покрытия);

$$f_t = 0,11 \text{ (кадмиевое покрытие гайки);}$$

- десятая затяжка: $f_t = 0,17$ (гайка без покрытия);

$$f_t = 0,2 \text{ (кадмиевое покрытие гайки).}$$

Расчет M_m выполнять по формуле

$$M_m = \frac{1}{3} Q_0 \cdot f_t \cdot \frac{a^3 - d_0^3}{a^2 - d_0^2},$$

где Q_0 – усилие затяжки, Н; a – внешний диаметр опорного торца, мм; d_0 – диаметр отверстия в корпусе под болт, мм.

В зависимости от задач проектирования предполагается выбор вариантов покрытия из приведенного выше списка, что должно быть учтено в программе расчета M_m .

10. Разработать программу для расчета диаметра d болтов, изготовленных из различных материалов, по заданному внешнему усилию N , воздействию на болт, если известно, что допустимое напряжение при растяжении $[\sigma_p]$ у болтов из различных материалов принимает следующие значения: для стали 45 – 7,2 кгс/мм²; для стали 20 – 4,8 кгс/мм²; для стали 30ХГСА – 20,8 кгс/мм².

Расчет d выполнять по формуле

$$d = \sqrt{\frac{4N}{\pi \cdot [\sigma_p]}},$$

где d – внутренний диаметр резьбы болта, мм; N – усилие растяжения болта, кгс; $[\sigma_p]$ – допустимое напряжение при растяжении, кгс/мм².

В зависимости от задач проектирования предполагается выбор материалов из приведенного выше списка, что должно быть учтено в программе расчета d .

11. Разработать программу для расчета момента на ключе $M_{кл}$ для создания силы затяжки резьбовых соединений, если болты изготовлены из материалов с различным пределом текучести σ_T , принимающим следующие значения: для стали 20 – $\sigma_T = 24$ кгс/мм²; для стали 45 – $\sigma_T = 45$ кгс/мм²; для стали 30ХГСА – $\sigma_T = 108$ кгс/мм².

Расчет $M_{кл}$ выполнять по формуле

$$M_{кл} = 7 \cdot 10^{-4} \cdot \sigma_m \cdot d^3,$$

где $M_{кл}$ – момент на ключе, Н·м; d – диаметр резьбы, мм.

В зависимости от задач проектирования предполагается выбор материалов из приведенного выше списка, что должно быть учтено в программе.

12. Разработать программу для расчета момента на ключе $M_{кл}$ для создания силы затяжки Q_0 резьбовых соединений, если приведенный коэффициент трения f принимает следующие значения:

- без покрытия болтов (гаек): $f = 0,2$ без смазки;
 $f = 0,16$ со смазкой;
- цинковое покрытие болтов (гаек): $f = 0,22$ без смазки;
 $f = 0,18$ со смазкой;
- медное покрытие болтов (гаек): $f = 0,18$ без смазки;
 $f = 0,14$ со смазкой.

Для расчета $M_{кл}$ использовать формулу

$$M_{кл} = f \cdot Q_0 \cdot d \cdot 10^{-2},$$

где $M_{кл}$ – момент на ключе, Н·м; Q_0 – сила затяжки, кгс; d – наружный диаметр резьбы, мм.

В зависимости от задач проектирования предполагается выбор вариантов покрытия болтов (гаек) из приведенного выше списка, а также предполагается учет наличия или отсутствия смазки, что должно быть учтено в программе расчета $M_{кл}$.

13. Разработать программу для расчета предельного крутящего момента M_k для шпоночного соединения между валом и ступицей, если для шпонок из стали 45 приняты следующие значения допустимых напряжений на смятие:

- при непрерывном использовании редукторов с полной нагрузкой $[\sigma]_{см} = 5 \text{ кгс/мм}^2$;

- при среднем режиме использования редукторов $[\sigma]_{\text{см}} = 14 \text{ кгс/мм}^2$;
- при предельных статических перегрузках $[\sigma]_{\text{см}} = 26 \text{ кгс/мм}^2$.

Для расчета M_K использовать формулу

$$M_K = \frac{[\sigma]_{\text{см}} \cdot d \cdot \ell_p \cdot t_2}{2} \cdot 10^{-2},$$

где M_K – момент крутящий, Н·м; $[\sigma]_{\text{см}}$ – допустимое напряжение на смятие, кгс/мм²; d – диаметр вала, мм; ℓ_p – рабочая длина шпонки, мм; t_2 – глубина заделки шпонки в ступицу вала, мм; $t_2 = 0,4h$, где h – высота шпонки, мм.

В зависимости от задач проектирования предполагается выбор вариантов режима использования редукторов из приведенного выше списка, что должно быть учтено в программе расчета M_K .

14. Разработать программу для расчета предельного крутящего момента M_K , передаваемого шлицевым соединением в изделиях подъемно-транспортного машиностроения при тяжелых (1), средних (2) или хороших (3) условиях их эксплуатации.

Допускаемые напряжения смятия $[\sigma]_{\text{см}}$ для шлицевых соединений, зубья которых не подвергаются термообработке, принимают следующие значения в зависимости от условий эксплуатации: для тяжелых условий $[\sigma]_{\text{см}} = 400 \text{ кгс/мм}^2$; для средних условий $[\sigma]_{\text{см}} = 700 \text{ кгс/мм}^2$ (2); для хороших условий $[\sigma]_{\text{см}} = 1000 \text{ кгс/мм}^2$ (3).

Для расчета M_K (в Н·м) использовать формулу

$$M_K = \frac{[\sigma]_{\text{см}} \cdot d_{\text{ср}} \cdot z \cdot h \cdot \ell \cdot \psi}{2} \cdot 10^{-2},$$

где $d_{\text{ср}} = \frac{D + d}{2}$, D и d – соответственно наружный и внутренний диаметры, мм; z – число зубьев; h и ℓ – соответственно высота и длина

поверхности контакта зубьев, мм, причем $h = \frac{D-d}{2}$; ψ – коэффициент, значение которого обычно принимают из диапазона 0,7 ... 0,8.

В зависимости от задач проектирования предполагается выбор условий эксплуатации оборудования из приведенного выше списка, что должно быть учтено в программе расчета M_k .

15. Разработать программу для расчета контактного давления q_m в сопряжении с натягом деталей машин, которое обеспечивает неподвижность соединения деталей при нагружении осевой силой A , если охватываемые детали (втулки) выполняют из стали (1), чугуна (2) и латуни (3), при этом охватываемая деталь (вал) в вариантах (1) ... (3) изготовлена из стали.

Расчет q_m выполнять по формуле

$$q_m = \frac{A \cdot k}{\mu \cdot \pi \cdot d \cdot \ell},$$

где A – осевая сила, Н; k – коэффициент запаса сцепления ($k = 1,5 \dots 2$); μ – коэффициент трения, принимающий различные значения для вариантов (1) ... (3) сопряжения деталей: $\mu = 0,06$ для варианта (1) – «сталь-сталь»; $\mu = 0,7$ для варианта (2) – «сталь-чугун»; $\mu = 0,1$ для варианта (3) – «сталь-латунь»; d и ℓ – соответственно диаметр и длина посадочной поверхности, мм.

В зависимости от задач проектирования предполагается выбор варианта сочетания материалов из приведенного выше списка, что должно быть учтено в программе расчета q_m .

16. Разработать программу для расчета давления q_m в сопряжении с натягом деталей машин. Такой характер сопряжения деталей обеспечивает неподвижность соединения деталей при нагружении осевой силой A , причем способы сборки соединений стальных вала и втулки пред-

ставлены операциями механической запрессовки (1) или тепловой сборки (2).

Расчет q_m выполнять по формуле

$$q_m = \frac{A \cdot k}{\mu \cdot \pi \cdot d \cdot \ell},$$

где A – осевая сила, Н; k – коэффициент запаса ($k = 1,5 \dots 2$); μ – коэффициент трения, принимающий различные значения для вариантов (1) и (2) сопряжения деталей: $\mu = 0,06$ – для способа сборки (1); $\mu = 0,14$ – для способа сборки (2); d и ℓ – соответственно диаметр и длина посадочной поверхности, мм.

В зависимости от задач проектирования предполагается выбор одного из способов сборки из приведенного выше списка, что должно быть учтено в программе расчета q_m .

17. Разработать программу для расчета контактного давления q_m в сопряжении с натягом вала и втулки, которое обеспечивает неподвижность соединения деталей при нагружении крутящим моментом M_k , если втулки изготовлены из алюминиевого сплава (1), латуни (2) и пластмассы (3), вал изготовлен из стали.

Расчет q_m выполнять по формуле

$$q_m = \frac{2 \cdot M_k \cdot K}{\mu \cdot \pi \cdot d^2 \cdot \ell},$$

где M_k – момент крутящий, Н·м; k – коэффициент запаса сцепления ($k = 1,5 \dots 2$); d и ℓ – соответственно диаметр и длина посадочной поверхности, мм; μ – коэффициент трения, принимающий различные значения: $\mu = 0,02$ – для варианта (1) – «сталь-алюминий»; $\mu = 0,05$ – для варианта (2) – «сталь-латунь»; $\mu = 0,4$ – для варианта (3) – «сталь-пластмасса».

В зависимости от задач проектирования предполагается выбор варианта сочетания материалов из приведенного выше списка, что должно быть учтено в программе расчета q_m .

18. Разработать программу для расчета контактного давления q_m в сопряжении с натягом валов и втулок, которое обеспечивает неподвижность соединения деталей при нагружении крутящим моментом M_k , если способы сборки вала и втулки представлены операциями механической запрессовки (1) и тепловой сборки (2).

Расчет q_m выполнять по формуле

$$q_m = \frac{2 \cdot M_k \cdot K}{\mu \cdot \pi \cdot d^2 \cdot \ell},$$

где M_k – момент крутящий, Н·м; k – коэффициент запаса сцепления ($k = 1,5 \dots 2$); d и ℓ – соответственно диаметр и длина посадочной поверхности, мм; μ – коэффициент трения: $\mu = 0,02$ – для способа сборки (1); $\mu = 0,05$ – для способа сборки (2).

В зависимости от задач проектирования предполагается выбор одного из способов сборки из приведенного выше списка, что должно быть учтено в программе расчета q_m .

19. Разработать программу для расчета контактного давления q_m в сопряжении с натягом валов и втулок, которое обеспечивает неподвижность соединения деталей при нагружении осевой силой A и крутящим моментом M_k , если втулки выполняют из стали (1), чугуна (2) и латуни (3), при этом вал в вариантах (1), (2) и (3) изготавливают из стали.

Расчет q_m выполнять по формуле

$$q_m = \frac{K \cdot \sqrt{\left(\frac{2 \cdot M_k}{d}\right)^2 + A^2}}{\mu \cdot \pi \cdot d \cdot \ell},$$

где A – осевая сила, Н; k – коэффициент запаса ($k = 1,5 \dots 2$); μ – коэффициент трения, принимающий различные значения для вариантов (1) ... (3) сопряжения деталей: $\mu = 0,06$ для варианта (1) – «сталь-сталь»; $\mu = 0,7$ для варианта (2) – «сталь-чугун»; $\mu = 0,1$ для варианта (3) – «сталь-латунь»; d и ℓ – соответственно диаметр и длина посадочной поверхности, мм.

В зависимости от задач проектирования предполагается выбор варианта сочетания материалов из приведенного выше списка, что должно быть учтено в программе расчета q_m .

20. Разработать программу для расчета срока службы T ремней ременных передач, если ремни передач выполняют плоскими прорезиненными (1), клиновыми кордотканевыми (2), клиновыми кордошнуровыми (3). Расчеты T выполнять по формуле

$$T = \frac{1}{3600 \cdot v \cdot z_{ш}} \cdot \left(\frac{C}{\sigma_{\max}} \right)^m,$$

где T – срок службы ремня, ч; v – число пробегов ремня в секунду:

$$v = \frac{V}{L},$$

где V – скорость ремня, м/с; L – длина ремня, м; $z_{ш}$ – число шкивов; C – номинальное напряжение в ремне: $C = 600 \dots 700$ кгс/мм² – для ремней (1); $C = 210$ кгс/мм² – для ремней (2); $C = 300$ кгс/мм² – для ремней (3); m – показатель степени, значения которого различны для вариантов (1) ... (3) используемых ремней: $m = 5$ – для ремней (1); $m = 9$ – для ремней (2); $m = 11$ – для ремней (3).

В зависимости от задач проектирования предполагается выбор варианта ременных передач (ремней) из приведенного выше списка, что должно быть учтено в программе расчета срока службы T ремней.

21. Разработать программу для расчета тяговой способности клиноременных передач по показателю общей передаваемой мощности W (в кВт),

если условия нагрузки передач варьируются в диапазоне: спокойная нагрузка (1), умеренные колебания нагрузки (2), значительные колебания нагрузки (3). При этом число Z одновременно работающих ремней равно трем.

Расчет W выполнять по формуле

$$W = W_0 \cdot \frac{K_z}{K_g},$$

где W_0 – мощность, передаваемая одним ремнем, кВт; K_z – коэффициент, учитывающий неравномерное распределение нагрузки между одновременно работающими ремнями (при $Z = 3$ $K_z = 0,95$); K_g – коэффициент динамичности, учитывающий условия (1) ... (3) нагрузки передачи: $K_g = 1$ в условиях (1); $K_g = 1,1$ в условиях (2); $K_g = 1,25$ в условиях (3) нагрузки передачи.

В зависимости от задач проектирования предполагается выбор варианта колебания нагрузки передачи из приведенного выше списка, что должно быть учтено в программе расчета W .

22. Разработать программу для расчета тяговой способности клиноременных передач по показателю общей передаваемой мощности W (в кВт), если число Z одновременно работающих ремней варьируется от 3 до 8, при этом передача испытывает спокойную нагрузку.

Расчет W выполнять по формуле

$$W = W_0 \cdot \frac{K_z}{K_g},$$

где W_0 – мощность, передаваемая одним ремнем, кВт; K_z – коэффициент, учитывающий неравномерное распределение нагрузки между одновременно работающими ремнями (при $Z = 3$ $K_z = 0,95$); K_g – коэффициент динамичности, учитывающий условия (1) ... (3) нагрузки передачи: $K_g = 1$ в условиях (1); $K_g = 1,1$ в условиях (2); $K_g = 1,25$ в условиях (3) нагрузки передачи.

Коэффициент K_z принимать в зависимости от числа Z работающих ремней: $K_z = 0,95$ при $Z = 2 \dots 3$; $K_z = 0,9$ при $Z = 4 \dots 6$; $K_z = 0,85$ при Z свыше 6.

В зависимости от задач проектирования предполагается выбор числа одновременно работающих ремней, что должно быть учтено в программе расчета W .

23. Разработать программу для расчета необходимого (потребного) числа ремней Z ременной передачи, если угол обхвата α принимает значения от 70° до 150° с шагом 20° .

Расчет Z выполнять по формуле

$$Z = \frac{W \cdot K_g}{W_0 \cdot K},$$

где W – общая передаваемая передачей мощность, кВт; K_g – коэффициент динамичности (принять $K_g = 1,1$); K – корректирующий коэффициент:

$$K = K_\alpha \cdot K_L \cdot K_z,$$

где K_α – коэффициент угла обхвата:

$\alpha, ^\circ$	70	90	110	130	150
K_α	0,56	0,68	0,78	0,88	0,92

K_L – коэффициент, учитывающий влияние длины ремня на его радиус (принять $K_\alpha = 0,99$); K_z – коэффициент, учитывающий неравномерное распределение нагрузки между одновременно работающими ремнями (см. задание 22).

В зависимости от задач проектирования предполагается выбор того или другого угла обхвата из приведенной выше таблицы, что должно быть учтено в программе расчета необходимого (потребного) числа ремней Z ременной передачи.

24. Разработать программу, которая по величине среднего арифметического отклонения профиля R_a микронеровности поверхности позволяет определить высоту микронеровностей (по 10 точкам) R_z (в мкм) поверхности деталей

$$R_z = 4R_a, \text{ если } R_a = (1,25 \dots 2,5) \text{ мкм};$$

$$R_z = 5R_a, \text{ если } R_a = (0,32 \dots 0,63) \text{ мкм};$$

$$R_z = 0,1R_a, \text{ если } R_a < 0,04 \text{ мкм}.$$

25. Разработать программу для расчета среднего арифметического отклонения профиля R_a (в мкм) микронеровностей поверхностей отверстий, если методы обработки отверстий представлены следующим набором операций: сверление (1), зенкерование (2), развертывание (3).

Расчет R_a выполнять по формулам

$$R_a = 6,36 \cdot d^{0,25} \cdot V^{0,12} \cdot S^{0,41}, \quad (1)$$

$$R_a = 5,6 \cdot \frac{V^{0,16} \cdot S^{0,76}}{d^{0,23}}, \quad (2)$$

$$R_a = 0,1 \cdot \frac{d^{0,26} \cdot V^{0,84}}{d^{0,27}}, \quad (3)$$

где d – диаметр инструмента, мм ($d = 12 \dots 25$); V – скорость резания, м/мин; S – подача, мм/об.

Значения V и S выбирать из следующих рядов:

$V = 20$ м/мин, $S = 0,06$ мм/об при сверлении;

$V = 40$ м/мин, $S = 0,3$ мм/об при зенкеровании;

$V = 10$ м/мин, $S = 1,6$ мм/об при развертывании.

В зависимости от задач проектирования предполагается назначение различных операций (1) ... (3) и вариантов их сочетания для обработки отверстий, что должно быть учтено в программе расчета R_a .

26. Разработать программу для расчета высоты микронеровностей (по 10 точкам) R_z (в мкм) поверхностей отверстий, если обработку отверстий выполняют следующими способами: сверлением (1), зенкерованием (2), развертыванием (3), причем способ (1) может быть использован исключительно, а также в сочетании с (2) и (3).

Расчет R_z производить по формулам

$$R_z = 48,7 \cdot \frac{d^{0,17} \cdot S^{0,46}}{V^{0,04}}, \quad (1)$$

$$R_z = 458 \cdot \frac{S^{0,41}}{d^{0,64} \cdot V^{0,5}}, \quad (2)$$

$$R_z = 1,82 \cdot \frac{d^{0,04} \cdot V^{0,4}}{S^{0,1}}, \quad (3)$$

где d – диаметр инструмента, мм ($d = 12 \dots 25$); V – скорость резания, м/мин; S – подача, мм/об.

Значения V и S выбирать из следующих рядов:

$V = 20$ м/мин, $S = 0,06$ мм/об при сверлении;

$V = 40$ м/мин, $S = 0,3$ мм/об при зенкеровании;

$V = 10$ м/мин, $S = 1,6$ мм/об при развертывании.

В зависимости от задач проектирования предполагается выбор способа (способов) обработки отверстий из приведенного выше списка, что должно быть учтено в программе расчета R_z .

27. Разработать программу для расчета среднего арифметического отклонения профиля R_a (в мкм) микронеровностей поверхностей, если обработку поверхностей выполняют следующими способами: фрезерование торцовое (1), строгание (2), шлифование (3), причем способ (3) может быть использован в сочетании с (1) и (2).

Расчет R_a выполнять по формулам:

$$R_a = 4,83 \cdot \frac{S_z^{1,69} \cdot t^{0,15}}{V^{1,23} \cdot \rho^{0,14} \cdot \gamma^{0,46}}, \quad (1)$$

где S_z – подача на зуб ($S_z = 0,08$ мм/зуб); t – глубина резания ($t = 0,3$ мм); V – скорость резания ($V = 44$ м/мин); ρ – радиус при вершине резца ($\rho = 0,5$ мм); γ – передний угол ($\gamma = 20^\circ$).

$$R_a = 37,3 \cdot \frac{S^{1,93} \cdot \gamma^{0,27}}{V^{0,11} \cdot t^{0,12} \cdot \rho^{0,8}}, \quad (2)$$

причем S – подача на двойной ход ($S = 0,25 \dots 0,5$ мм/дв. ход); V – скорость резания ($V = 5,6 \dots 22$ м/мин).

Остальные параметры (t, ρ, γ) – как при фрезеровании.

$$R_a = 0,27 \cdot \frac{k^{0,48} \cdot V^{0,27} \cdot S^{0,8} \cdot t^{0,13}}{i^{0,11}}, \quad (3)$$

причем $V = 0,21$ м/с; $S = 3$ мм/ход; $t = 0,01$ мм; i – число выхаживаний ($i = 5$); k – зернистость круга (46 ... 100 мкм).

В зависимости от задач проектирования предполагается выбор способа (способов) обработки поверхностей из приведенного выше списка, что должно быть учтено в программе расчета R_a .

28. Разработать программу для расчета среднего арифметического отклонения профиля R_a (в мкм) микронеровностей наружных поверхностей деталей, изготовленных точением, если в зависимости от потребности, детали изготавливают из стали 3 (1), стали 20 (2), стали 45 (3), стали 70 (4).

Расчет R_a выполнять по формуле

$$R_a = k_0 \cdot \frac{S^{k_1} \cdot (90 + \gamma)^{k_4}}{\rho^{k_2} \cdot V^{k_3}},$$

где S – подача ($S = 0,43$ мм/об); V – скорость резания ($V = 282$ м/мин); γ – передний угол ($\gamma = +4^\circ$); ρ – радиус при вершине резца.

Коэффициенты $k_0 \dots k_4$ выбирать из таблицы.

Сталь	k_0	k_1	k_2	k_3	k_4
Сталь 3	0,01	0,65	0,6	0,5	1,9
Сталь 20	41,8	0,75	0,75	1,38	0,25
Сталь 45	7,0	0,85	0,85	0,36	0,15
Сталь 70	5,8	1,1	1,1	0,15	0,45

В зависимости от задач проектирования предполагается выбор материала обрабатываемой заготовки из приведенного выше списка, что должно быть учтено в программе расчета R_a .

29. Разработать программу для расчета среднего арифметического отклонения профиля R_a (в мкм) микронеровностей наружных и внутренних поверхностей деталей в результате шлифования заготовок деталей из стали 40Х, если в зависимости от потребности шлифуют либо наружные (1), либо внутренние поверхности (2).

Расчет R_a выполнять по формулам:

$$R_a = 0,47 \cdot \frac{S_{np}^{0,35} \cdot t^{0,29} \cdot k^{0,08} \cdot j_{cm}^{0,34}}{V_{\partial}^{0,06} \cdot S_{pad}^{0,003} \cdot i^{0,024}}; \quad (1)$$

$$R_a = 92,5 \cdot \frac{S_{np}^{0,39} \cdot S_{pad}^{0,05} \cdot i^{0,012} \cdot k^{0,06}}{V^{0,23} \cdot H^{0,04} \cdot j_{cn}^{0,98}}, \quad (2)$$

где S_{np} – продольная подача, мм; $S_{np} = 0,5 \cdot B$, где B – ширина круга, мм; t – глубина резания, мм; k – зернистость круга ($k = 16$ мкм); j_{cm} – жесткость станка ($j_{cm} = 30000$ Н/мм); V_{∂} – скорость вращения заготовки ($V_{\partial} = 50$ м/мин); S_{pad} – радиальная подача ($S_{pad} = 0,5$ мм/дв. ход); i – число выхаживаний; V – скорость резания ($V = 50$ м/мин); H – твердость круга ($H = 3$).

В зависимости от задач проектирования предполагается выбор указанных выше вариантов шлифования заготовок, что должно быть учтено в программе расчета R_a .

30. Разработать программу для расчета среднего отклонения профиля R_a (в мкм) микронеровностей наружных поверхностей деталей в результате полирования поверхностей лентами, если в зависимости от условий полирование выполняют или лентами конечной длины (1), или лентами «бесконечной» длины (2) с различной зернистостью K , и в зависимости от потребности полируют заготовки из стали 45 или из стали 30ХГСА.

Расчет R_a вести по формулам:

$$R_a = k_0 \cdot \frac{R_{a_{исх}}^{k_1} \cdot K^{k_2} \cdot (10^5)^{k_3}}{p^{k_3} \cdot V_{\partial}^{k_4} \cdot \tau^{k_5}}; \quad (1)$$

$$R_a = 0,004 \cdot \left(\frac{P}{10}\right)^{0,46} \cdot \frac{K^{1,26}}{\tau^{0,61}}, \quad (2)$$

где $R_{a_{исх}}$ – исходная шероховатость поверхности заготовки (до полирования) по параметру R_a ($R_a = 0,32$ мкм); K – зернистость ленты ($K = 40$ мкм); p – давление ленты ($p = 5 \cdot 10^5$ Па); V_{∂} – скорость вращения детали (принять $V_{\partial} = 10$ м/мин); τ – время обработки ($\tau = 25$ с); P – сила прижима ленты ($P = 70$ Н).

Значения коэффициентов $k_0 \dots k_5$ приведены в таблице:

Материал заготовки	k_0	k_1	k_2	k_3	k_4	k_5
Сталь 45 (HRC 56...62)	1,01	0,81	0,01	0,33	0,1	0,2
Сталь 30ХГСА (HRC 30...35)	0,49	1,17	0,47	0	0,14	0,24

В зависимости от задач проектирования предполагается выбор указанных выше вариантов полирования заготовок, что должно быть учтено в программе.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №3. РЕШЕНИЕ ПРОЕКТНЫХ ЗАДАЧ С ИНВАРИАНТНЫМ ВЫБОРОМ

3.1. Конструкция оператора выбора CASE

С помощью этого оператора можно выбрать вариант выполнения в программе каких-либо операторов из любого количества вариантов.

В общем виде *структура оператора CASE* такова:

```
CASE < СЕЛЕКТОР > OF
    < ВАРИАНТ 1 > : < ОПЕРАТОР 1 > ;
    < ВАРИАНТ 2 > : < ОПЕРАТОР 2 > ;
    -----
    < ВАРИАНТ N-1 > : < ОПЕРАТОР N-1 > ;
ELSE < ВАРИАНТ N >
    END ;
```

Здесь < СЕЛЕКТОР > – это выражение какого-либо простого порядкового типа (например, INTEGER или CHAR), значение которого вычисляется в программе заранее. < ВАРИАНТ ...> – некоторая константа, представляющая собой одно из возможных значений селектора.

В конструкции оператора CASE строку ELSE можно опускать. Если же ELSE присутствует в операторе выбора, то перед этой строкой в отличие от IF точку с запятой можно ставить.

Каждый вариант в CASE может быть представлен более чем одной константой. В этом случае возможные значения селектора данного варианта отделяются друг от друга запятыми, образуя список. При этом элементом такого списка может быть не только константа, но и отрезок значений (тип – диапазон).

В любом случае в конструкции оператора CASE должны быть заданы все возможные значения селектора – либо явным их упоминанием, либо отнесением к пункту ELSE.

Пример 3.1. Составить тест–программу по выбору компонентов латунного сплава. Алгоритм программы представлен в приложении Г.3.

```
PROGRAM TEST;
USES CRT;
  VAR X: BYTE;
  BEGIN
CLRSCR;
  WRITELN ('ЛАТУНЬ – ЭТО:');
  WRITELN ('1. СПЛАВ МЕДИ С ЖЕЛЕЗОМ');
  WRITELN ('2. СПЛАВ МЕДИ С ОЛОВОМ');
  WRITELN ('3. СПЛАВ МЕДИ С ЦИНКОМ');
  WRITELN;
    WRITE ('ВЫБЕРИТЕ ОДИН ПРАВИЛЬНЫЙ ОТВЕТ:');
    READLN (X);
  CASE X OF
    1: WRITELN ('ОТВЕТ НЕПРАВИЛЬНЫЙ!');
    2: WRITELN ('ОТВЕТ НЕВЕРНЫЙ!');
    3: WRITELN ('ПРАВИЛЬНЫЙ ОТВЕТ!!!');
  ELSE WRITELN ('ТАКОГО НОМЕРА ОТВЕТА НЕТ !');
    END;
  READKEY
  END.
```

3.2. Задания к практическому занятию

1. В меню записан список деталей: болт, винт, гайка, шайба, ось, вал. Составить программу, обеспечивающую распечатку названия детали по указанному номеру.
2. Используя исходные данные к задаче 1, составить программу, обеспечивающую распечатку названия детали по первой букве.
3. Составить программу, которая по названию месяца выдавала бы количество рабочих дней при пятидневной рабочей неделе с учетом государственных праздников.

4. Составить программу, которая по первой букве названия месяца выдавала бы количество рабочих дней при пятидневной рабочей неделе с учетом государственных праздников.
5. Составить программу, которая по порядковому номеру месяца выдавала бы количество рабочих дней при пятидневной рабочей неделе с учетом государственных праздников.
6. Составить программу, которая по названию средства технологического оснащения (сверло, фреза, метчик, патрон, кондуктор, тиски) сообщала бы сведения о группе, к которой оно принадлежит (инструмент, приспособление).
7. Составить программу, которая по первой букве названия средства технологического оснащения (сверло, фреза, метчик, патрон, кондуктор, тиски) сообщала бы сведения о группе, к которой оно принадлежит (инструмент, приспособление).
8. Составить программу, которая по номеру квартала сообщала бы рабочему, какие месяцы он содержит.
9. Составить программу, которая по названию месяца выдавала бы его порядковый номер и название времени года.
10. Дан список дисциплин, изучаемых в УлГТУ, и отчетность по ним. Составить программу, которая по названию дисциплины выдавала бы отчетность по нему: станки (экзамен, зачет); детали машин (экзамен); режущий инструмент (экзамен, зачет); гидропривод (экзамен, зачет); теоретическая механика (экзамен).
11. Дан список дисциплин, изучаемых в УлГТУ, и номер семестра, когда они изучаются. Составить программу, которая по номеру семестра выдавала бы список изучаемых дисциплин: станки – 5, 6; детали машин – 3, 4; режущий инструмент – 7, 8; теоретическая механика – 3, 4; гидропривод – 7, 8.
12. Составить программу, которая по обозначению резьбы гаек (М4, М5, М6, М8, М10) выдавала бы соответствующие размеры «под ключ» (6, 7, 8, 13, 17 мм).

13. Составить программу, которая по названию типа производства (единичное, серийное, массовое) выдавала бы сведения о соответствующем виде описания технологического процесса (ТП): маршрутный, маршрутно–операционный, операционный ТП.
14. Составить программу, которая по названию материала заготовки (сталь, чугун, бронза, медь, алюминий) выдавала бы к какой группе (черных или цветных) металлов и сплавов он относится.
15. Составить программу, которая по первой букве названия материала заготовки (сталь, чугун, бронза, медь, алюминий) выдавала бы к какой группе (черных или цветных) металлов и сплавов он относится.
16. Составить программу, которая по марке твердого сплава выдавала бы сообщение о его химическом составе (в процентах): ВК4 – 4% Со, 96% WC; ВК6 – 6% Со, 94% WC; ВК15 – 15% Со, 85% WC; Т15К6 – 6% Со, 15% TiC, 79% WC; ТТ8К6 – 6% Со, 2% TaC, 8% TiC, 84% WC.
17. Составить программу, которая по марке автомобиля (ВАЗ, УАЗ, МАЗ, ГАЗ, АЗЛК, КАМАЗ) выдавала бы название города, где выпускаются данные автомобили (Тольятти, Ульяновск, Минск, Нижний Новгород, Москва, Набережные Челны).
18. Составить программу, которая по названию детали (вал, шар, конус, планка, угольник, плита, корпус) выдавала бы сообщение о том, к какой группе (тела вращения или корпусные детали) относится данная деталь.
19. Составить программу, которая по значению длины детали в миллиметрах выдавала бы эту длину в метрах, дециметрах или сантиметрах.
20. Составить программу, которая по номеру штрих – кода на изделии выдавала бы название страны – изготовителя: 00 – США; 46 – Россия; 50 – Великобритания, 45 – Япония; 76 – Швейцария; 73 – Швеция.
21. Составить программу, которая по номеру рабочей смены распечатывала бы ее время проведения.
22. Составить программу, которая по названию инструмента (резец, фреза, хон, шлифовальный круг, напильник) выдавала бы наименование рабо-

чей специальности (токарь, фрезеровщик, хонинговщик, шлифовщик, слесарь).

23. Составить программу, которая по первой букве названия инструмента (резец, фреза, хон, шлифовальный круг, напильник) выдавала бы наименование рабочей специальности (токарь, фрезеровщик, хонинговщик, шлифовщик, слесарь).
24. Составить программу, которая по буквам номерного знака выдавала бы название страны, где зарегистрирован автомобиль: RUS – Россия, DE – Германия, US – США, SWE – Швеция, POL – Польша.
25. Составить программу, которая по номеру пары выдавала бы время ее проведения на машиностроительном факультете.
26. Составить программу, которая по аббревиатуре выдавала бы название стандарта: ГОСТ – межгосударственный стандарт, ГОСТ Р – российский стандарт; СТП – стандарт предприятия; ТУ – технические условия.
27. Составить программу, которая по названию инструмента выдавала бы его принадлежность к определенному виду обработки: резец, фреза, метчик, сверло – лезвийная обработка; шлифовальный круг, притир, хон, лента – абразивная обработка.
28. Составить программу, которая по первой букве названия инструмента выдавала бы его принадлежность к определенному виду обработки: резец, фреза, метчик, сверло – лезвийная обработка; шлифовальный круг, притир, хон, лента – абразивная обработка.
29. Составить программу, которая по названию технологической операции выдавала бы сведения о том, к какому виду технологического воздействия она относится: точение, фрезерование, шлифование, хонингование – обработка резанием; сварка, пайка, клепка, запрессовка – сборочные операции.
30. Составить программу, которая по первой букве названия технологической операции выдавала бы сведения о том, к какому виду технологического воздействия она относится: точение, фрезерование, шлифование, хонингование – обработка резанием; сварка, пайка, клепка, запрессовка – сборочные операции.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №4.

АНАЛИТИЧЕСКИЙ РАСЧЕТ ПАРАМЕТРОВ ТЕХНИЧЕСКИХ СИСТЕМ ПРИ ИЗМЕНЕНИИ ОДНОГО ФАКТОРА

4.1. Организация циклов при разработке алгоритмов решения технических задач

Иногда при решении технических задач необходимо находить значение целевой функции при неоднократном изменении какого-либо параметра. Алгоритм решения подобного класса задач, в котором предусмотрено неоднократное выполнение одной и той же последовательности действий, называется **циклом**. Циклическая структура позволяет существенно сократить размер записи алгоритма, представить его компактно путем соответствующей организации повторения предписываемых действий. Естественно, что повторять какие-либо действия имеет смысл при каждом новом выходе на повторение. Это возможно, если в цикле меняются определенные параметры. Такие изменяемые параметры называют **параметрами цикла**.

Циклические структуры могут быть организованы, например, с помощью комбинации условного оператора IF и безусловного оператора GOTO.

Однако в языке FREE PASCAL имеются специальные операторы циклов (FOR, WHILE и REPEAT), позволяющие организовать циклы с известным или неизвестным числом повторений.

Оператор цикла с параметром FOR. Организует выполнение одного оператора с заранее известным числом повторений.

Существует два варианта FOR:

1. С увеличивающимся параметром

FOR <I>:= <START> TO <FINISCH> DO <ОПЕРАТОР>;

2. С уменьшающимся параметром

FOR <I>:= <START> DOWNT0 <FINISCH> DO < ОПЕРАТОР >; .

Здесь <I> – параметр (индекс) цикла, являющийся переменной порядкового типа; <START> – начальное значение параметра цикла; <FINISCH> – конечное значение параметра цикла; < ОПЕРАТОР > – выполняемый оператор.

Если в первом варианте значение <START> больше значения <FINISCH>, а во втором варианте – <START> больше <FINISCH>, то оператор цикла FOR не выполняется ни разу.

В цикле FOR нельзя задавать шаг, отличный от 1.

После выхода из цикла его параметр становится неопределенным, кроме случая, когда выход из него был осуществлен с помощью оператора безусловного перехода GOTO или стандартной процедуры BREAK.

Пример 4.1. Составить программу для вычисления среднеарифметического размера деталей в партии. Алгоритм программы представлен в приложении Г.4.

```
PROGRAM SRED_RAZM;  
USES CRT;  
    VAR I, N: INTEGER;  
    SUM, SR, R: REAL;  
BEGIN  
    CLRSCR;  
    WRITE ('ВВЕДИТЕ КОЛИЧЕСТВО ДЕТАЛЕЙ В ПАРТИИ:');  
    READLN (N);  
    SUM:= 0;  
    FOR I:= 1 TO N DO  
        BEGIN  
            WRITE ('ВВЕДИТЕ РАЗМЕР',I,'-ОЙ ДЕТАЛИ В ММ:');  
            READLN (R);  
            SUM:= SUM+R  
        END;  
    SR:=SUM/N;  
    WRITE('СРЕДНЕАРИФМЕТИЧЕСКИЙ РАЗМЕР ДЕТАЛЕЙ');
```



```
WRITELN('В ПАРТИИ РАВЕН', SR:4:3, 'ММ');  
READKEY;  
END.
```

Оператор цикла с предусловием WHILE. Организует выполнение одного оператора с заранее неизвестным числом повторений.

Его структура имеет вид:

```
WHILE <УСЛОВИЕ> DO <ОПЕРАТОР >; ,
```

где <УСЛОВИЕ> – логическое выражение, истинность которого проверяется в начале каждой итерации; <ОПЕРАТОР > – выполняемый оператор.

Так как истинность логического выражения (условия) проверяется в самом начале цикла, то его тело может не выполниться ни разу.

Выход из цикла WHILE осуществляется, если проверяемое условие окажется ложным. Если данное логическое условие никогда не становится ложным, то цикл будет выполняться бесконечное число раз (т. е. программа заикнется). Для выхода из состояния заикливания необходимо нажать комбинацию клавиш «CTRL + BREAK». При необходимости многократного повторения в цикле с предусловием нескольких операторов их следует заключить в операторные скобки «BEGIN – END».

Пример 4.2. Модернизировать программу из примера 4.1 с использованием цикла WHILE. Алгоритм программы представлен в приложении Г.5.

```
PROGRAM SRED_RAZM_1;  
USES CRT;  
    VAR I, N: INTEGER;  
        SUM, SR, R: REAL;  
BEGIN  
    CLRSCR;  
    WRITE ('ВВЕДИТЕ КОЛИЧЕСТВО ДЕТАЛЕЙ В ПАРТИИ:');  
    READLN (N);  
    SUM:=0;  
    I:=1;
```

```

WHILE (I<= N) DO
    BEGIN
    WRITE ('ВВЕДИТЕ РАЗМЕР',I,'-ОЙ ДЕТАЛИ В ММ: ');
    READLN (R);
    SUM:= SUM+R;
    I:=I+1;
    END;
    SR:= SUM/N;
    WRITE ('СРЕДНЕАРИФМЕТИЧЕСКИЙ РАЗМЕР ДЕТАЛЕЙ');
    WRITELN ('В ПАРТИИ РАВЕН ', SR:4:3,' ММ');
    READKEY;
END.

```

Оператор цикла с постусловием REPEAT. Организует выполнение цикла, состоящего из любого числа операторов, с заранее неизвестным числом повторений.

Структура оператора REPEAT такова:

```

REPEAT
    < ОПЕРАТОР 1>;
    < ОПЕРАТОР 2>;
    .....
    < ОПЕРАТОР N>
UNTIL <УСЛОВИЕ>; ,

```

где < ОПЕРАТОРЫ> – выполняемые операторы, составляющие тело цикла; <УСЛОВИЕ> – логическое выражение, истинность которого проверяется в конце каждой итерации.

Служебные слова «REPEAT – UNTIL» выполняют роль ограничителей тела цикла, что позволяет обходиться без операторных скобок «BEGIN – END» при наличии нескольких операторов в теле цикла.

Так как истинность логического выражения проверяется в конце цикла REPEAT, то тело цикла выполняется, по крайней мере, один раз, даже если условие с самого начала оказалось истинным.

Выход из цикла REPEAT осуществляется, если проверяемое условие становится истинным. Если же оно ложно всегда, то цикл будет выполняться бесконечное число раз (т. е. программа заикнется). Для выхода из состояния заикливания необходимо нажать комбинацию клавиш «CTRL + BREAK».

Пример 4.3. Модернизировать программу из примера 4.2 с использованием цикла REPEAT. Алгоритм программы представлен в приложении Г.6.

```
PROGRAM SRED_RAZM_2;
USES CRT;
    VAR I, N: INTEGER;
        SUM, SR, R: REAL;
BEGIN
    CLRSCR;
    WRITE ('ВВЕДИТЕ КОЛИЧЕСТВО ДЕТАЛЕЙ В ПАРТИИ:');
    READLN (N);
    SUM:= 0;
    I:= 1;
    REPEAT
        WRITE ('ВВЕДИТЕ РАЗМЕР',I,'-ОЙ ДЕТАЛИ В ММ: ');
        READLN (R);
        SUM:= SUM+R;
        I:=I+1;
    UNTIL (I>N);
    SR:= SUM/N;
    WRITE ('СРЕДНЕАРИФМЕТИЧЕСКИЙ РАЗМЕР ДЕТАЛЕЙ');
    WRITELN('В ПАРТИИ РАВЕН ', SR:4:3,' ММ');
    READKEY;
END.
```

4.2. Задания к практическому занятию

1. Составить программу для определения скорости резания V при сверлении отверстия сверлом диаметром D , изменяющимся от 1 до 4 мм с шагом 1 мм: $V = C_V \cdot D^q \cdot K_V / T^m \cdot S^y$, где $C_V = 400$; $q = m = y = 0,5$; $K_V = 1$; $T = 60$ мин – период стойкости и $S = 0,2$ мм/об.
2. Составить программу для определения крутящего момента $M_{кр}$ при сверлении отверстия сверлом диаметром D , изменяющимся от 12 до 16 мм с шагом 1 мм: $M_{кр} = 10 \cdot C_M \cdot D^q \cdot S^y \cdot K_p$, где $C_M = 0,1$; $q = 2$; $S = 0,4$ мм/об; $y = 0,8$ и $K_p = 1$.
3. Составить программу для определения усилия P , изгибающего оправку с торцевой фрезой под действием составляющих сил резания – $P_z = 20$ Н и P_y , изменяющуюся от 10 до 30 Н с шагом 5 Н: $P = \sqrt{P_y^2 + P_z^2}$.
4. Составить программу для определения крутящего момента $M_{кр}$ на шпинделе фрезерного станка при изменении силы резания P_z от 100 до 102,5 Н с шагом $h_p = 0,5$ Н, диаметр фрезы D равен 20 мм:
$$M_{кр} = P_z \cdot D / 200.$$
5. Составить программу для определения эффективной мощности при фрезеровании N_e , если сила P_z изменяется от 40 до 41 Н с шагом $h_p = 0,2$ Н: $N_e = P_z \cdot V / 61200$, где $V = 200$ м/мин – скорость резания.
6. Составить программу для определения силы затяжки болтов P_3 при изменении длины болта L от 50 до 70 мм с шагом $h_l = 5$ мм:
 $P_3 = 4E \cdot \lambda \cdot D^2 / L\pi$, где $\lambda = 0,001$ м – удлинение, $E = 10^6$ Н/м² – модуль упругости, $D = 15$ мм – диаметр болтов.
7. Составить программу для определения усилия запрессовки P , если диаметр сопрягаемых деталей d изменяется от 10 до 30 мм с шагом $hd = 5$ мм: $P = f \cdot \pi \cdot d \cdot Lp$, где $f = 0,1$ – коэффициент трения; $p = 1000$ МПа – давление в контакте, $L = 50$ мм – длина сопряжения.
8. Составить программу для определения радиального зазора e_R в шарикоподшипниках, если диаметр шариков $d_{ш}$ изменяется от 8 до 10 мм с

шагом $h_d = 0,5$ мм: $e_R = e_0^2 / 4 \cdot (2R - d_{ш})$, где $R = 12$ мм – радиус желоба, $e_0 = 0,003$ мм – осевой зазор в шарикоподшипниках.

9. Составить программу для определения угла поворота φ стопорной гайки подшипника, если осевой зазор в подшипнике l_0 изменяется от 0,004 до 0,01 мм с шагом $h_l = 0,002$ мм: $\varphi = (l_0 / P) \cdot 360^\circ$, где $P = 1$ мм – шаг резьбы гайки.
10. Составить программу для нахождения усилия при раскатывании поверхности P_H , если глубина наклепанного слоя h_n изменяется от 0,01 до 0,04 мм с шагом 0,005 мм: $P_H = 2h_H^2 \cdot \delta_T \cdot m^2$, где $m = 1,5$ – поправочный коэффициент, $\delta_T = 600$ МПа – предел текучести материала детали.
11. Определить, насколько уменьшится толщина стенки калибруемого цилиндра Δh , если начальная толщина h_0 изменяется от 1 до 5 мм с шагом $h_h = 1$ мм: $\Delta h = -2h_0 z / (2d_0 + h_0)$, где $z = 0,05$ мм – величина припуска на калибрование, $d_0 = 60$ мм – начальный диаметр цилиндра.
12. Определить, насколько изменится цена заготовки Π_3 , если оптовая цена единицы массы материала Π_M увеличивается от 100 до 500 руб/кг с шагом $h_{\Pi} = 100$ руб/кг: $\Pi_3 = q_m \cdot \Pi_M \cdot K_T$, где $K_m = 1,05$ – коэффициент транспортных расходов при приобретении материалов, $q_m = 0,8$ кг/шт. – норма расхода материала.
13. Определить, как изменится прогиб вала Y , установленного в центрах, если сила P_y , действующая на него, увеличивается от 200 до 1000 Н с шагом $h_p = 200$ Н: $Y = P_y \cdot L^3 / 48EJ$, где $E = 10^6$ Н/м² – модуль упругости материала, $J = 0,05 \cdot d^4$ – момент инерции поперечного сечения, $d = 40$ мм – диаметр, а $L = 60$ мм – длина вала.
14. Определить интенсивность изнашивания фрезы U , если ширина фрезерования B изменяется от 40 до 60 мм с шагом $h_B = 5$ мм:
 $U = (1 + 100 / B) U_0$, где $U_0 = 16$ мкм/км – начальный износ фрезы.
15. Определить, насколько изменится диаметр пузырьков пара d_0 , если поверхностное натяжение смазочно–охлаждающей жидкости σ возрастает от 0,02 до 0,08 г/см с шагом $h_\sigma = 0,02$ г/см: $d_0 = 0,2\psi \sqrt{\sigma / \rho_c}$, где $\psi = 50^\circ$ – угол смачивания, $\rho_{ж} = 1$ г/см³ – плотность СОЖ.

16. Определить удлинение заготовки ΔL при ее нагревании, если температура нагрева ΔT изменяется от 40 до 100 °С с шагом 10 °С, коэффициент линейного расширения α равен $16 \cdot 10^{-6} \text{ } ^\circ\text{C}^{-1}$, а начальная длина заготовки L составляет 100 мм: $\Delta L = L \cdot \alpha \cdot \Delta T$.
17. Определить, как будет изменяться объем цилиндра $V = \pi \cdot D^2 \cdot H$, если его высота H изменяется от 10 до 50 мм с шагом $h = 10$ мм при $D = 8$ мм.
18. Шероховатость обработанной шлифованием поверхности можно рассчитать по формуле $R_a = c \cdot t^z$. Определить значения R_a при изменении показателя степени z от 0,2 до 0,5 с шагом 0,05. Глубина резания $t = 0,02$ мм, коэффициент $c = 2$.
19. Плотность газов из трубы заводской котельной убывает с высотой по закону $\rho = \rho_0 \cdot e^{-hz}$. Определить плотность газа на высотах $h = 100, 200, 300, 400, 500, 600$ м, считая, что $\rho_0 = 1,29 \text{ кг/м}^3$, $z = 1,23 \cdot 10^{-4} \text{ м}^{-1}$.
20. Шероховатость поверхности заготовки после электроэрозионного вырезания R_z (мкм) рассчитывают по формуле $R_z = 3\sqrt{\frac{Q}{f}}$, где Q – производительность обработки, мм/с; f – частота импульсов, Гц. Определить значение R_z , если Q изменяется от 0,5 до 1 мм/с с шагом 0,1 мм/с, а $f = 20 \text{ кГц}$.
21. Жесткость токарного резца определяется по формуле $j = P_y / y$. Рассчитать жесткость испытываемых резцов, если P_y изменялась от 5 000 до 10 000 Н с шагом 1 000 Н, а величина смещения $y = 0,3$ мм.
22. Определить время фрезерования заготовок, если длина обработки составляет ряд 160; 180; 200; 220; 240 мм, а подача фрезы равна 0,14 м/мин.
23. Электрическое сопротивление силового кабеля станка определяется по зависимости $R = \rho \cdot \ell / s$, где ρ – удельное сопротивление, Ом·мм²/м; ℓ – длина, м; s – площадь сечения кабеля в мм². Рассчитать сопротивление кусков кабеля при $\ell = 5; 10; 15; 20; 25$ м; $\rho = 0,017 \text{ Ом·мм}^2/\text{м}$; $d = 3,5 \text{ мм}^2$.

24. Общее сопротивление двух параллельно соединенных проводников в цепи станка рассчитывается по формуле $1 / R = 1 / R_1 + 1 / R_2$, определить общее сопротивление R , если сопротивление одного из проводников изменялось от 20 до 100 Ом с шагом 20 Ом, а сопротивление другого равно 60 Ом.
25. Шероховатость обработанной шлифованием поверхности можно рассчитать по формуле $R_a = c \cdot t^z$. Определить значения R_a при изменении глубины резания t от 0,01 до 0,04 мм с шагом 0,005 мм; показателе степени z , равном 0,5, и коэффициенте $c = 2$.
26. Согласно закону Ома $I = U / R$. Определить возможную силу тока в электрической цепи станка, если напряжение U изменяется от 220 до 225 В с шагом 1 В, а $R = 800$ Ом.
27. Мощность электрической цепи станка можно определить по формуле $P = I \cdot U$. Рассчитать ее для силы тока $I = 0,1; 0,2; 0,3; 0,4; 0,5$ А и $U = 120$ В.
28. Технологический процесс состоит из операций лезвийной и абразивной обработки. Определить возможное время технологического процесса, если продолжительность лезвийных операций колеблется от 2 до 18 мин с шагом 4 мин, а продолжительность абразивных операций равна 25 мин.
29. Общее сопротивление двух последовательно соединенных проводников в цепи станка рассчитывается по формуле ($R = R_1 + R_2$), определить общее сопротивление R , если сопротивление одного из проводников изменялось от 10 до 50 Ом с шагом 10 Ом, а сопротивление другого равно 40 Ом.
30. Определить, как будет изменяться объем цилиндра $V = \pi \cdot D^2 \cdot H$, если его диаметр D изменяется от 4 до 12 мм с шагом $d = 2$ мм при $H = 20$ мм.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №5.

АНАЛИТИЧЕСКИЙ РАСЧЕТ ПАРАМЕТРОВ ТЕХНИЧЕСКИХ СИСТЕМ ПРИ ИЗМЕНЕНИИ НЕСКОЛЬКИХ ФАКТОРОВ

5.1. Основные понятия о вложенных циклах

Если телом цикла является циклическая структура, то такие циклы называют *вложенными* (или *сложными*). Цикл, содержащий в себе другой цикл, называется *внешним*. Цикл, содержащийся в теле другого цикла, называется *внутренним*.

Внутренний и внешний циклы могут быть любыми из трех рассмотренных в практическом занятии №4 циклов (FOR, WHILE и REPEAT). При этом правила организации (построения) как внешнего, так и внутреннего циклов остаются такими же, как и для простого цикла каждого из этих видов.

Однако при построении вложенных циклов необходимо соблюдать дополнительное условие – все операторы внутреннего цикла должны полностью лежать в теле внешнего цикла. Возможная глубина вложенности циклов (количество уровней) ограничивается только памятью ПЭВМ.

Параметры циклов, расположенных на разных уровнях вложенности, изменяются не одновременно. Вначале все свои значения изменит параметр цикла самого «глубокого» уровня при фиксированных (начальных) значениях параметров циклов с менее «глубоким» уровнем. Затем изменяется на один шаг значение параметра цикла следующего уровня вложенности и снова полностью выполняется самый внутренний цикл и т. д. до тех пор, пока параметры циклов всех уровней не примут все требуемые значения.

При этом, если в сложном цикле с глубиной вложенности k число повторений циклов на каждом уровне равно $N_0, N_1, \dots, N_k$ соответственно, то общее количество повторений тела самого внутреннего цикла равно $N = N_0 \cdot N_1 \cdot \dots \cdot N_k$.

Пример 5.1. Предприятие выпускает металлопрокат из четырех различных материалов с плотностью 7,0; 7,2; 7,4; 7,6 г/см³ диаметром 10, 20, 30 и 40 мм. Составить программу для расчета массы погонного метра каждого вида проката. Алгоритм программы представлен в приложении Г.6.

```
PROGRAM PROKAT;
USES CRT;
    VAR DN, DK, HD, PN, PK, HP, D, P, V, M: REAL;
BEGIN
    CLRSCR;
    WRITE ('ВВЕДИТЕ ШАГ ИЗМЕНЕНИЯ ПЛОТНОСТИ HP В КГ/М^3: ');
    READLN (HP);
    WRITE ('ВВЕДИТЕ ШАГ ИЗМЕНЕНИЯ ДИАМЕТРА HD В М: ');
    READLN (HD);
    WRITE ('ВВЕДИТЕ МИНИМАЛЬНОЕ (НАЧАЛЬНОЕ) ЗНАЧЕНИЕ');
    WRITE ('ПЛОТНОСТИ PN В КГ/М^3: ');
    READLN (PN);
    WRITE ('ВВЕДИТЕ МАКСИМАЛЬНОЕ (КОНЕЧНОЕ) ЗНАЧЕНИЕ');
    WRITE ('ПЛОТНОСТИ PK В КГ/М^3: ');
    READLN (PK);
    WRITE ('ВВЕДИТЕ МИНИМАЛЬНОЕ (НАЧАЛЬНОЕ) ЗНАЧЕНИЕ');
    WRITE ('ДИАМЕТРА ПРОКАТА DN В М: ');
    READLN (DN);
    WRITE ('ВВЕДИТЕ МАКСИМАЛЬНОЕ (КОНЕЧНОЕ) ЗНАЧЕНИЕ');
    WRITE ('ДИАМЕТРА ПРОКАТА DK В М: ');
    READLN (DK);
    P:= PN;
    REPEAT
        D:= DN;
        REPEAT
            V:= PI*SQR(D)/4;
            M:= P*V;
```

```

WRITELN ('ПРИ P= ', P:4:2, ' КГ/М^3 И D=', D:4:2, ' M=', M:4:2, ' КГ');
D:= D+HD;
UNTIL (D>DK);
P:= P+HP;
READLN;
UNTIL (P>PK);
READKEY;
END.

```

5.2. Задания к практическому занятию

1. Составить программу для определения скорости резания V при сверлении отверстия сверлом диаметром D , изменяющимся от 1 до 4 мм с шагом 1 мм, и подачей S , изменяющейся от 0,1 до 0,4 мм/об с шагом 0,1 мм:

$V = C_V \cdot D^q \cdot K_V / T^m \cdot S^y$, где $C_V = 400$; $q = m = y = 0,5$; $K_V = 1$; $T = 60$ мин – период стойкости.

2. Составить программу для определения крутящего момента при сверлении $M_{кр}$ отверстия сверлом диаметром D , изменяющимся от 12 до 15 мм с шагом 1 мм, и подаче S , изменяющейся от 0,2 до 0,8 мм/об с шагом 0,2 мм:

$M_{кр} = 10C_M \cdot D^q \cdot S^y \cdot K_p$, где $C_M = 0,1$; $q = 2$; $y = 0,8$ и $K_p = 1$.

3. Составить программу для определения усилия P , изгибающего оправку с торцевой фрезой под действием составляющих сил резания P_z и P_y ,

изменяющихся от 10 до 25 Н с шагом 5 Н: $P = \sqrt{P_y^2 + P_z^2}$.

4. Составить программу для определения крутящего момента $M_{кр}$ на шпинделе фрезерного станка при изменении силы резания P_z от 100 до 102 Н с шагом $h_p = 0,5$ Н и диаметра фрезы D от 10 до 20 мм с шагом $h_d = 5$ мм: $M_{кр} = P_z \cdot D / 200$.

5. Составить программу для определения эффективной мощности при фрезеровании N_e , если сила P_z изменяется от 40 до 41 Н с шагом

$h_p = 0,25$ Н, а скорость резания V изменяется от 200 до 204 м/мин с шагом $h_v = 1$ м/мин: $N_e = P_z \cdot V / 61200$.

6. Составить программу для определения силы затяжки болтов P_3 при изменении длины болта L от 50 до 65 мм с шагом $h_l = 5$ мм и его диаметра D от 10 до 18 мм с шагом $h_d = 2$ мм: $P_3 = 4E\lambda \cdot D^2 / (L \cdot \pi)$, где $\lambda = 0,001$ м – удлинение болтов, $E = 10^6$ Н / м² – модуль упругости.
7. Составить программу для определения усилия запрессовки P , если диаметр сопрягаемых деталей d изменяется от 10 до 30 мм с шагом $h_d = 5$ мм, а их длина L – от 50 до 70 мм с шагом $h_l = 5$ мм: $P = f \cdot \pi \cdot d \cdot L \cdot p$, где $f = 0,1$ – коэффициент трения; $p = 1\,000$ МПа – давление в контакте.
8. Составить программу для определения радиального зазора e_R в шарикоподшипниках, если диаметр шариков $d_{ш}$ изменяется от 8 до 10 мм с шагом $h_d = 0,5$ мм, а осевой зазор e_0 – от 0,001 до 0,005 мм с шагом $h_e = 0,001$ мм: $e_R = e_0^2 / 4 \cdot (2R - d_{ш})$, где $R = 12$ мм – радиус желоба.
9. Составить программу для определения угла поворота φ стопорной гайки подшипника, если осевой зазор в подшипнике ℓ_0 изменяется от 0,004 до 0,01 мм с шагом $h_l = 0,002$ мм, а шаг резьбы гайки P – от 0,5 до 2 с шагом $h_p = 0,5$ мм: $\varphi = (\ell_0 / P) \cdot 360^\circ$.
10. Составить программу для нахождения усилия при раскатывании поверхности P_H , если глубина наклепанного слоя h_n изменяется от 0,2 до 0,4 мм с шагом 0,05 мм, а предел текучести материала детали δ_T – от 400 до 800 МПа с шагом $h_d = 100$ МПа: $P_H = 2h_H^2 \cdot \delta_T \cdot m^2$, где $m = 1,5$ – поправочный коэффициент.
11. Определить, насколько уменьшится толщина стенки калибруемого цилиндра Δh , если начальная толщина h_0 изменяется от 1 до 5 мм с шагом $h_h = 1$ мм, а начальный диаметр d_0 – от 50 до 70 мм с шагом $h_d = 5$ мм: $\Delta h = -2h_0 \cdot z / (2d_0 + h_0)$, где $z = 0,05$ мм – величина припуска на калибрование.
12. Определить, насколько изменится цена заготовки Π_3 , если оптовая цена единицы массы материала Π_M увеличивается от 100 до 500 руб./кг с шагом $h_{\Pi} = 100$ руб./кг, а норма расхода материала q_m – от 0,4 до

1 кг/шт. с шагом $h_q = 0,2$ кг/шт.: $\Pi_3 = q_m \cdot \Pi_M \cdot K_T$, где $K_m = 1,05$ – коэффициент транспортных расходов при приобретении материалов.

13. Определить, как изменится прогиб вала Y , установленного в центрах, если сила P_y , действующая на него, увеличивается от 400 до 1 000 Н с шагом $h_p = 200$ Н, а его длина L – от 50 до 70 мм с шагом $h_L = 5$ мм: $Y = P_y \cdot L^3 / 48EJ$, где $E = 10^6$ Н/м² – модуль упругости материала, $J = 0,05 \cdot d^4$ – момент инерции поперечного сечения вала, а $d = 40$ мм – диаметр вала.

14. Определить интенсивность изнашивания фрезы U , если ширина фрезерования B изменяется от 40 до 60 мм с шагом $h_B = 5$ мм, а начальный износ U_0 от 15 до 17 мкм/км с шагом $h_u = 0,5$ мкм/км:

$$U = (1 + 100 / B) \cdot U_0.$$

15. Определить, насколько изменится диаметр пузырьков пара d_0 , если поверхностное натяжение смазочно–охлаждающей жидкости σ возрастает от 0,02 до 0,08 г/см с шагом $h_\sigma = 0,02$ г/см, а плотность СОЖ $\rho_{\text{ж}}$ – от 0,5 до 2 г/см³ с шагом $h_\rho = 0,5$ г/см³: $d_0 = 0,2 \cdot \psi \cdot \sqrt{\sigma / \rho_c}$, где $\psi = 50^\circ$ – угол смачивания.

16. Определить удлинение заготовки ΔL при ее нагревании, если температура нагрева ΔT изменяется от 60 до 100 °С с шагом 10 °С, коэффициент линейного расширения α – от $12 \cdot 10^{-6}$ до $16 \cdot 10^{-6}$ °С⁻¹ с шагом 10^{-6} , а начальная длина заготовки L составляет 100 мм: $\Delta L = L \cdot \alpha \cdot \Delta T$.

17. Определить удлинение стальной заготовки ($\alpha = 12 \times 10^{-6}$ °С⁻¹) при ее нагревании от 20 до 80 °С с шагом 10 °С и изменении начальной длины L от 200 до 300 мм с шагом 25 мм: $\Delta L = L \cdot \alpha \cdot \Delta T$.

18. Определить удлинение заготовок, нагретых до $\Delta T = 120$ °С, если их первоначальная длина L изменялась от 100 до 120 мм с шагом 5 мм, а коэффициенты линейного расширения материалов α – от $14 \cdot 10^{-6}$ до $20 \cdot 10^{-6}$ °С⁻¹ с шагом $2 \cdot 10^{-6}$ °С⁻¹: $\Delta L = L \cdot \alpha \cdot \Delta T$.

19. Плотность газов из трубы заводской котельной убывает с высотой по закону $\rho = \rho_0 \cdot e^{-hz}$. Определить плотность газа на высотах $h = 100$,

200, 300, 400 м, считая, что $\rho_0 = 1,29 \text{ кг/м}^3$, а z изменяется от $1,23 \cdot 10^{-4} \text{ м}^{-1}$ до $1,27 \cdot 10^{-4} \text{ м}^{-1}$ с шагом 10^{-4} м^{-1} .

20. Шероховатость поверхности заготовки после электроэрозионного вырезания Rz (мкм) рассчитывают по формуле $Rz = \sqrt[3]{\frac{Q}{f}}$, где Q – производительность обработки, мм/с; f – частота импульсов, Гц. Определить значение Rz , если Q изменяется от 0,5 до 1 мм/с с шагом 0,1 мм/с, а f – от 17 до 20 кГц с шагом 1 кГц.
21. Жесткость токарного резца определяется по формуле $j = P_y / y$. Рассчитать жесткость испытываемых резцов, если P_y изменялась от 5 000 до 8 000 Н с шагом 1 000 Н, а величина смещения y имела значения 0,1; 0,2; 0,3; 0,4 мм.
22. Определить время фрезерования заготовок, если длина обработки составляет ряд 160; 180; 200; 220 мм, а подача фрезы изменяет свое значение от 0,1 до 0,15 м/мин с шагом 0,01 м/мин.
23. Электрическое сопротивление силового кабеля станка определяется по зависимости $R = \rho \cdot \ell / s$, где ρ – удельное сопротивление, Ом · мм²/м; ℓ – длина, м; s – площадь сечения кабеля в мм². Рассчитать сопротивление кусков кабеля при $\ell = 5; 10; 15; 20$ метров; $d = 3,0; 3,5; 4,0; 4,5$ мм² и $\rho = 0,017 \text{ Ом} \cdot \text{мм}^2/\text{м}$.
24. Общее сопротивление двух параллельно соединенных проводников в цепи станка рассчитывается по формуле $1 / R = 1 / R_1 + 1 / R_2$, определить общее сопротивление R , если сопротивление каждого из проводников изменялось от 20 до 100 Ом с шагом 20 Ом.
25. Общее сопротивление двух параллельно соединенных проводников в цепи станка рассчитывается по формуле $R = R_1 + R_2$, определить общее сопротивление R , если сопротивление каждого из проводников изменялось от 20 до 100 Ом с шагом 20 Ом.
26. Согласно закону Ома $I = U / R$. Определить возможную силу тока в электрической цепи станка, если напряжение U изменяется от 220 до

225 В с шагом 1 В, а сопротивление R – от 500 до 1 000 Ом с шагом 100 Ом.

27. Мощность электрической цепи станка можно определить по формуле $P = I \cdot U$. Рассчитать ее для силы тока $I = 0,2; 0,3; 0,4; 0,5$ А и для напряжения U , изменяющегося от 120 до 150 В с шагом 10 В.
28. Технологический процесс состоит из операций лезвийной и абразивной обработки. Определить возможное время технологического процесса, если продолжительность лезвийных операций колеблется от 2 до 18 мин с шагом 4 мин, а абразивных – от 12 до 30 мин с шагом 6 мин.
29. Шероховатость обработанной шлифованием поверхности можно рассчитать по формуле $R_a = c \cdot t^z$. Определить значения R_a при изменении глубины резания t от 0,01 до 0,04 мм с шагом 0,01 мм, показателя степени z от 0,4 до 0,6 с шагом 0,05 и постоянном значении коэффициента $c = 2$.
30. Определить, как будет изменяться объем цилиндра $V = \pi \cdot D^2 \cdot H$, если его диаметр D изменяется от 4 до 12 мм с шагом $d = 2$ мм, а высота H – от 18 до 30 мм с шагом $h = 4$ мм.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №6.

ОБРАБОТКА МАССИВОВ ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ ПРИ ИССЛЕДОВАНИИ ТЕХНИЧЕСКИХ СИСТЕМ

6.1. Понятие типа - массива

Массивом называется упорядоченная совокупность компонентов одинакового типа, в которой каждому компоненту соответствует набор номеров (индексов), однозначно определяющих его местоположение в общей последовательности.

Чтобы задать тип – массив, используют зарезервированное слово `ARRAY`, после которого в квадратных скобках указывают диапазон и тип индексов компонент и далее после слова `OF` – тип самих компонент:

`TYPE <имя массива> = ARRAY [<диапазон изменения и тип индексов>] OF <тип компонент>.`

Пример 6.1. Объявить в разделе типов одномерный массив `A` из трех вещественных чисел.

```
TYPE A = ARRAY [1..3] OF REAL;
```

Количество индексов n , необходимое при обращении к элементу массива, определяет n – мерность (размерность) массива.

Размерность массива может быть любой, а индекс – любого порядкового типа, кроме типа `LONGINT`. Компоненты массива могут быть любого, в том числе и структурированного типа.

После введения типа – массива можно задать переменные или типизированные константы этого типа. Причем при объявлении последних их компоненты указывают в круглых скобках и разделяются запятыми.

Пример 6.2. Для введенного одномерного массива `A` объявить переменные и константу этого типа.

```
CONST M0 : A = (1, 2, 3); VAR X, Y : A ;
```

На практике наиболее часто приходится иметь дело с двумерными массивами (матрицами).

Для объявления матриц существует несколько способов:

1. В разделе типов можно объявить матрицу как одномерный массив, элементами которого являются массивы

```
TYPE STROKA = ARRAY [1..3] OF REAL;
```

```
    MATR  = ARRAY [1..2] OF STROKA;
```

2. Объединить в одной строке запись двух одномерных массивов

```
TYPE MATR = ARRAY [1..2] OF ARRAY[1..3] OF REAL;
```

3. Указать при объявлении матрицы диапазоны изменения индексов для каждой ее размерности

```
TYPE MATR = ARRAY [1..3, 1..2] OF REAL;
```

4. Если указанный тип-массив используется в программе для работы с однородными матрицами, то удобно объявить их сразу в разделе описания переменных : VAR M : ARRAY [1..3, 1..2] OF REAL;

Ссылка на элемент матрицы A, лежащий на пересечении *i*-й строки и *j*-го столбца, имеет вид A[I, J] или A [I] [J].

Для задания значений элементам массива используют операторы назначения (присваивания :=) и ввода данных (READLN).

Для ввода матрицы A[I, J] по строкам используют вложенные циклы, состоящие из двух циклов FOR:

```
FOR I:=1 TO M DO
```

```
    FOR J:=1 TO N DO
```

```
        BEGIN
```

```
            WRITE('ВВЕДИТЕ A[', I, J, ']: ');
```

```
            READLN(A[I, J])
```

```
        END;
```

Вывод матрицы производится аналогично.

Пример 6.3. Составить программу для вычисления суммы двух матриц A и B размерностью 2×3, состоящих из целых чисел. Алгоритм программы представлен в приложении Г.8.

```
PROGRAM SUMMATR;
```

```
USES CRT;
```

```
    VAR A, B, S: ARRAY [1..2, 1..3] OF INTEGER;
```

```
        I, J: INTEGER;
```



```

BEGIN
CLRSCR;
  FOR I:=1 TO 2 DO
    FOR J:=1 TO 3 DO
      BEGIN
        WRITE ('ВВЕДИТЕ A[' ,I,J,'] :');
        READLN (A[I,J])
      END;
    FOR I:=1 TO 2 DO
      FOR J:=1 TO 3 DO
        BEGIN
          WRITE ('ВВЕДИТЕ B[' ,I,J,'] :');
          READLN (B[I,J])
        END;
      WRITELN;
      WRITELN ('РЕЗУЛЬТИРУЮЩАЯ МАТРИЦА S');
      WRITELN;
      FOR I:=1 TO 2 DO
        BEGIN
          FOR J:=1 TO 3 DO
            BEGIN
              S[I,J]:= A[I,J]+B[I,J];
              WRITE (S[I,J]:3, ' ');
            END;
          WRITELN
        END;
      END;
    READKEY
  END.

```

6.2. Задания к практическому занятию

1. Сортировочный автомат настроен на диаметр валов $D \pm \frac{\delta}{H}$, где δ и H – соответственно, верхнее и нижнее допустимые отклонения диаметра

вала D . Задайте настроечный размер, сформируйте массив из валов различного диаметра и рассортируйте их на три группы: годные детали, исправимый и неисправимый брак.

2. На операции фрезерования была обработана партия корпусных заготовок. Задайте количество и размеры заготовок. Найдите величину рассеивания размеров $\omega = A_{max} - A_{min}$, где A_{max} и A_{min} – соответственно, максимальный и минимальный размеры заготовок, и средний арифметический размер заготовок в партии.
3. На механическом участке имеется стеллаж для хранения деталей, состоящий из n полок и m ячеек. Задайте вес деталей в партии и разместите их на стеллаже так, чтобы он сохранил устойчивое положение: на нижней полке находились самые тяжелые детали, выше – детали меньшего веса и т. д.
4. На вертикально–фрезерном станке рабочий должен обработать партию из n заготовок в размер H . Задайте количество и размеры исходных заготовок в партии, а также размер готовой детали. Определите, какой припуск необходимо снять фрезеровщику с каждой заготовки, чтобы получить годные детали.
5. Технологический процесс считается надежным, если коэффициент надежности $\psi \geq 1,2$, $\psi = T / \omega$; T – допуск на выдерживаемый размер; ω – поле рассеивания этого размера у деталей в партии. Задайте количество и размеры деталей в партии, допуск на размер T . Определите поле рассеивания (см. задание 2) и сделайте вывод о надежности техпроцесса.
6. Технологом разработаны m вариантов техпроцесса обработки заготовки, в каждом из которых необходимо выдержать n размеров. Задайте допуски T и поля рассеивания ω по каждому из размеров, определите коэффициент ψ (см. задание 5) и выберите для реализации на производстве тот техпроцесс, который надежен по всем размерам.
7. В грузовом отсеке самолета симметрично расположены два багажных отделения, содержащие $m \times n$ ячеек. Задайте вес грузов, размещаемых в ячейках, и определите их суммарный вес в каждом из отделений. Если

разница между ними превышает 5%, то во избежание потери самолетом центровки укажите, из какого багажного отделения нужно удалить лишний груз.

8. Сортировочный автомат измеряет и направляет деталь в одну из n групп, на которые разбит допуск T на выдерживаемый у детали номинальный диаметр d . Задайте количество и размеры деталей в партии, а затем рассортируйте их по группам.
9. На сборочную операцию поступает n валов и втулок, которые соединяются между собой в процессе сборки. Задайте допуски на диаметр вала T_v и отверстия во втулке T_o у всех деталей. Определите разность между допусками $S = T_o - T_v$ и по характеру соединения ($S > 0$ – соединение с зазором; $S < 0$ – соединение с натягом; $S = 0$ – беззазорное соединение) рассортируйте их на три группы, подсчитав процент попадания деталей в каждую из групп.
10. После обработки на настроенном станке партии из n заготовок были измерены их размеры. Определите количество и сформируйте массив из «выбросов» (ложных замеров) среди всех измерений, считая «выбросом» такой замер, величина которого более чем в 1,5 раза превышает средний арифметический размер деталей в партии.
11. На универсальном токарном станке предстоит обработать n заготовок различной длины L и диаметра d . В зависимости от соотношения L/d заготовки могут быть закреплены в трехкулачковом патроне консольно (если $L/d \leq 0,4$) или дополнительно поджаты задним центром (если $L/d > 4$). Задайте длины и диаметры заготовок в партии и рассортируйте их на две группы в зависимости от способа крепления на станке.
12. В кладовой цеха инструменты хранятся на стеллаже из $m \times n$ ячеек. Задайте количество инструментов в каждой ячейке и месячную норму расхода каждого из них. Сформируйте из инструментов три массива: инструменты, которых не хватит до конца месяца; инструменты, расходуемые по плану; инструменты, хранящиеся в кладовой с запасом.
13. Стеллаж инструментальной кладовой цеха содержит $m \times n$ ячеек. Задайте плановое и фактически израсходованное количество инструмен-

тов в каждой ячейке. Рассортируйте инструменты на три группы: инструменты, которых не хватит для обработки всей партии заготовок; планово расходуемые инструменты; инструменты, нормируемые с запасом, указав по каждой из позиций инструмента его запас или недостачу в процентах от плана.

14. Контролер ОТК оценивает качество обработки партии из n деталей по m параметрам. Задайте количество деталей в партии, число контролируемых параметров и степень выполнения каждого из них («+» – выполнено; «-» – не выполнено). Рассортируйте детали на две группы: бракованные и годные детали.
15. По исходным условиям задания 14 рассортируйте бракованные детали на две группы: детали, у которых выдержано меньше половины контролируемых параметров, и детали, «годные» по большинству параметров.
16. При проектировании рамы автомобиля конструктор представил ее в виде решетки, состоящей из $n \times n$ узлов (n – нечетное число), в которых расположены сосредоточенные массы. Задайте количество узлов рамы и сосредоточенную массу каждого узла. Если суммы масс узлов относительно продольной оси отличаются более чем на 10%, то для устранения опасного крена укажите, с какой стороны необходимо изменить массу узлов.
17. По исходным условиям задания 15 выясните, есть ли опасность крена автомобиля относительно поперечной оси рамы, и укажите, с какой стороны необходимо изменить массу узлов для устранения опасности.
18. На механическом участке работает n рабочих m разных специальностей. Задайте количество рабочих и их специальностей, процент выполнения плана у каждого из них. Рассортируйте рабочих на группы по специальностям, выделяя среди них тех, кто не выполнил план или перевыполнил его.
19. В вибрационное загрузочное устройство (ВЗУ) засыпают партию из n деталей исполнения А и В, которые могут иметь дополнительно отверстие. ВЗУ осуществляет первичное разделение деталей на два потока

по исполнениям (А или В), которые, в свою очередь, направляются в два других ВЗУ, осуществляющих сортировку деталей на две группы – с отверстиями и без отверстий. Задайте количество деталей, их параметры и рассортируйте их на четыре группы.

20. Номенклатура выпускаемых на механическом участке деталей включает n позиций. Для изготовления каждой детали применяют материал определенной марки (сталь, чугун, латунь, алюминий, титан и т. п.). Задайте количество позиций выпускаемых деталей, их наименование и марку материала для каждой детали. Рассортируйте выпускаемые детали по группам материала.
21. Технолог анализирует n вариантов техпроцесса, каждый из которых состоит из m операций различной длительности. Задайте количество вариантов техпроцесса, число операций в каждом из них и время их выполнения. Среди всех вариантов найдите самый протяженный по времени и самый производительный техпроцесс.
22. В течение месяца токарем было изготовлено n деталей типа тел вращения, каждое из которых имело определенную длину L , диаметр d , а некоторые детали и центральное отверстие. Задайте количество деталей и их параметры. Осуществите сортировку тел вращения на следующие группы: валы (при $L > d$ и отсутствии отверстия); гильзы (если $L > d$ и есть отверстие); пальцы ($L = d$; отверстие отсутствует); втулки ($L = d$; есть отверстие); диски ($L < d$ при отсутствии отверстия); шайбы ($L < d$ при наличии отверстия).
23. Работница расфасовывает болты в m пакетов по 100 штук в каждом. Вес одного болта составляет 20 граммов. Задайте число пакетов и суммарный вес болтов в них. Определите для каждого пакета «точность» расфасовки – количество недостающих или переложённых болтов.
24. Вероятность безотказной работы машины равна произведению вероятностей узлов, из которых она состоит. Задайте количество машин m и их узлов n , а также вероятность надежной работы каждого узла. Найдите среди всех машин изделие с самой большой и самой низкой вероятностью.

25. По исходным условиям задания 24 определите самый ненадежный узел у каждого из изделий и вычислите, чему равна средняя вероятность безотказной работы машин.
26. На механическом участке в три смены работают по n человек, каждый из которых изготавливает определенное количество деталей. Задайте число рабочих и объем выпущенной каждым из них продукции. Определите количество деталей, изготовленных на участке за смену, и средний объем продукции, приходящийся на одного рабочего.
27. По исходным условиям задания 26 найдите среди рабочих в каждой смене «передовиков» и «отстающих» и вычислите у каждого из них процент опережения или отставания относительно среднего объема продукции, приходящегося на одного рабочего.
28. Транспортная служба завода имеет m электрокаров, которые перемещаются между цехами. Одной зарядки аккумулятора хватает на то, чтобы электрокар проехал 20 км. Задайте количество электрокаров, пройденное каждым из них расстояние и определите, сколько раз нужно заряжать аккумулятор у каждого транспортного средства.
29. По исходным условиям задания 28 задайте годовой пробег электрокаров и вычислите через сколько циклов «зарядки – разрядки» нужно менять аккумуляторы у каждого из них, учитывая, что ресурс аккумулятора составляет 5 000 циклов.
30. В заводском гараже имеется n машин. Задайте у каждой машины расход топлива и расстояние, пройденное автомобилем за месяц. Определите месячный расход топлива на автомобиль, суммарный объем горючего, израсходованного всеми машинами, и средние затраты на топливо по гаражу на автомобиль.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №7.

СОЗДАНИЕ И РАБОТА С БАЗАМИ ДАННЫХ ЭЛЕМЕНТОВ ТЕХНИЧЕСКИХ СИСТЕМ

7.1. Понятие типа–записи

Тип–запись – это структура данных, состоящая из фиксированного числа компонентов, называемых *полями*. При этом поля могут быть различных типов. Каждому полю в записи присваивается свое неповторимое имя и доступ к этому полю осуществляется не по индексу (как в массиве), а по имени.

При задании типа–записи после зарезервированного слова RECORD следует перечислить все поля записи с указанием через двоеточие их типов и завершить задание типа–записи словом END. Внутри записи поля отделяются друг от друга точкой с запятой. Количество полей может быть любым.

```
TYPE <ИМЯ ЗАПИСИ>=
  RECORD
    <ИМЯ ПОЛЯ 1>:<ТИП ПОЛЯ 1>;
    <ИМЯ ПОЛЯ 2>:<ТИП ПОЛЯ 2>;
    -----
    <ИМЯ ПОЛЯ N>:<ТИП ПОЛЯ N>
  END;
```

Пример 7.1. Задать тип–запись, содержащую сведения о фамилии и дате рождения рабочего.

```
TYPE RABOCH=
  RECORD
    FAM : STRING[20];
    YEAR : WORD;
    MONTH : 1..12;
    DAY : 1..31
  END;
```

После введения типа–записи можно задать переменные или типизированные константы, относящиеся к этому типу:

```
VAR X, Y : RABOCH;  
CONST IVAN : RABOCH = (FAM: 'СИДОРОВ'; YEAR: 1975;  
                        MONTH : 5; DAY: 9);
```

Если необходимо распространить записи на несколько объектов, то следует ввести в программу переменную – массив из этих записей:

```
VAR WORK : ARRAY [1..25] OF RABOCH;
```

Доступ к полям записи осуществляется путем указания имени переменной (или константы) и имени поля, разделенных точкой (сочленяющей точкой): X. DAY: = 15; {ПЕРЕМЕННАЯ ТИПА- ЗАПИСИ};

```
WORK [1]. FAM := 'ИВАНОВ'; {ПЕРЕМЕННАЯ ТИПА-  
                           МАССИВА ИЗ ЗАПИСЕЙ}.
```

Пример 7.2. Составить программу для вычисления у каждого рабочего на участке из 50 человек средней зарплаты за 3 месяца. Алгоритм программы представлен в приложении Г.9.

```
PROGRAM SREDZARPL;  
USES CRT;  
TYPE RABOCH = RECORD  
    FAM: STRING [20];  
    Z1, Z2, Z3: REAL;  
    SREDZ: REAL  
END;  
VAR WORK: ARRAY [1..50] OF RABOCH;  
    I : INTEGER;  
BEGIN  
CLRSCR;  
    FOR I :=1 TO 50 DO  
        BEGIN  
WRITE ('Ф. И. О.-'); READLN (WORK [I]. FAM);  
WRITE ('1 МЕСЯЦ.-'); READLN (WORK [I]. Z1);  
WRITE ('2 МЕСЯЦ.-'); READLN (WORK [I]. Z2);
```



```

WRITE ('3 МЕСЯЦ-'); READLN (WORK [I]. Z3);
READKEY
END;
FOR I:=1 TO 50 DO
  BEGIN
    WORK [I]. SREDZ:=(WORK[I]. Z1 + WORK [I]. Z2 + WORK [I]. Z3)/3;
    WRITELN (WORK [I]. FAM, '-' , WORK [I]. SREDZ:2:2, 'РУБ')
  END;
  READKEY
END.

```

При обращении к компонентам записи необходимо указать как имя переменной, так и имя поля, что приводит к удлинению программы и излишней громоздкости.

Для устранения этого неудобства используется *оператор присоединения WITH (оператор работы над записями)*, который позволяет осуществлять доступ к полям записи так, как если бы они были простыми переменными.

Структура оператора WITH имеет вид:

```

WITH<ИМЯ ПЕРЕМЕННОЙ> DO
  < ОПЕРАТОР >;

```

Внутри оператора WITH к полям обращаются только по имени. Например, в программе SREDZARPL вычисление средней зарплаты запишется в виде:

```

FOR I:=1 TO 25 DO
  WITH WORK[I] DO
    BEGIN
      SREDZ:= (Z1+Z2+Z3)/3; и т. д.
    END
  END

```

7.2. Задания к практическому занятию

1. Сформировать и вывести на дисплей массив записей, описывающих следующую таблицу:

Наименование детали	Номер детали в партии			
	1	2	...	n
	Длина, мм	Длина, мм		Длина, мм

Определить средний размер деталей в партии и представить в виде:

Наименование детали	Средняя длина, мм
---------------------	-------------------

2. Сформировать и вывести на дисплей массив записей, описывающих следующую таблицу:

ФИО рабочего	Специаль- ность	График работы		
		1 смена	2 смена	3 смена

Вывести список токарей, работающих в 1 смену, в виде:

ФИО рабочего	Смена
--------------	-------

3. Сформировать и вывести на дисплей массив записей, описывающих следующую таблицу:

ФИО рабочего	Цех	Число прогулов за год		
		в 1 квартале	...	в 4 квартале

Вывести фамилии рабочих, имеющих более 10 прогулов за год, по форме:

ФИО рабочего	Число прогулов
--------------	----------------

4. Сформировать и вывести на дисплей массив записей, описывающих следующую таблицу:

ФИО рабочего	Специальность	Стаж работы
--------------	---------------	-------------

Вывести информацию о всех шлифовщиках цеха в виде:

ФИО шлифовщика	Стаж работы
----------------	-------------

5. Сформировать и вывести на дисплей массив записей, описывающих следующую таблицу:

Наименование детали	Время обработки, мин		
	точение	сверление	шлифование

Вывести список деталей с циклом обработки больше 10 мин в виде:

Наименование детали	Время обработки, мин
---------------------	----------------------

6. Сформировать и вывести на дисплей массив записей, описывающих следующую таблицу:

Наименование детали	Длина, мм	Ширина, мм	Высота, мм
---------------------	-----------	------------	------------

Вывести список деталей высотой меньше 50 мм в виде:

Наименование детали	Длина, мм	Ширина, мм
---------------------	-----------	------------

7. Сформировать и вывести на дисплей массив записей, описывающих следующую таблицу:

Наименование детали	Кол-во, шт.	Материал	Вес, кг
---------------------	-------------	----------	---------

Вывести информацию о деталях весом до 1 кг в виде:

Наименование детали	Кол-во, шт.	Материал
---------------------	-------------	----------

8. Сформировать и вывести на дисплей массив записей, описывающих следующую таблицу:

Модель станка	Страна-изготовитель	Год выпуска	Завод-покупатель
---------------	---------------------	-------------	------------------

Вывести модели станков США, имеющихся на УАЗе, по форме:

Модель станка	Год выпуска
---------------	-------------

9. Сформировать и вывести на дисплей массив записей, описывающих следующую таблицу:

ФИО водителя	Год рождения	С какого года на УАПК	Марка легковой машины	Марка грузовой машины
-----------------	-----------------	--------------------------	-----------------------------	-----------------------------

Вывести фамилии всех водителей легковых машин по форме:

ФИО водителя	Год рождения
--------------	--------------

10. Сформировать и вывести на дисплей массив записей, описывающих следующую таблицу:

ФИО студента	Дата рождения	Зарплата отца	Зарплата матери	Число братьев и сестер
-----------------	------------------	------------------	--------------------	---------------------------

Вывести фамилии студентов, размер доходов на одного человека в семье которых менее 5 000 рублей, по форме:

ФИО студента	Доход на одного человека, руб.
--------------	--------------------------------

11. Сформировать и вывести на дисплей массив записей, описывающих следующую таблицу:

ФИО	Место работы	Домашний адрес	Стаж работы
-----	--------------	----------------	-------------

Вывести фамилии работников, имеющих стаж более 10 лет и работающих на УАЗе, по форме:

ФИО	Стаж работы
-----	-------------

12. Сформировать и вывести на дисплей массив записей, описывающих следующую таблицу:

Наименование кафедры	Фамилия преподавателя	Ученое звание	Стаж работы
-------------------------	--------------------------	------------------	----------------

Вывести список доцентов с кафедры «Технология машиностроения» по форме:

Фамилия преподавателя	Стаж работы
-----------------------	-------------

13. Сформировать и вывести на дисплей массив записей, описывающих следующую таблицу:

Табельный номер	ФИО рабочего	Зарплата, руб.	Премия, руб.
-----------------	--------------	----------------	--------------

Вывести информацию о рабочих, лишенных премии, по форме:

Табельный номер	ФИО рабочего
-----------------	--------------

14. Сформировать и вывести на дисплей массив записей, описывающих следующую таблицу:

№ n/n	ФИО студента	Группа	Кол-во выполненных лабораторных работ	Язык программирования
----------	-----------------	--------	--	--------------------------

Вывести список студентов, выполнивших более двух лабораторных работ на Паскале, по форме:

ФИО студента	Количество работ
--------------	------------------

15. Сформировать и вывести на дисплей массив записей, описывающих следующую таблицу:

Название книги	ФИО автора	Издательство	Число томов
----------------	------------	--------------	-------------

Вывести информацию о книгах, выпущенных издательством «Машиностроение», в виде:

Название книги	ФИО автора
----------------	------------

16. Сформировать и вывести на дисплей массив записей, описывающих следующую таблицу:

ФИО рабочего	Специаль- ность	Стаж работы в цехе	Количество прогулов за год
-----------------	--------------------	-----------------------	-------------------------------

Вывести информацию о прогулах, допущенных фрезеровщиками цеха, в следующем виде:

ФИО фрезеровщика	Количество прогулов
------------------	---------------------

17. Сформировать и вывести на дисплей массив записей, описывающих следующую таблицу:

Наименование деталей	Программа выпуска, шт.	Материал заготовки	Себестоимость изготовления
----------------------	---------------------------	-----------------------	-------------------------------

Вывести информацию о деталях, изготовленных из чугуна, в виде:

Детали из чугуна	Количество, шт.	Себестоимость, руб.
------------------	-----------------	---------------------

18. Сформировать и вывести на дисплей массив записей, описывающих следующую таблицу:

Наименование станка	Модель	Мощность, кВт	Год выпуска
---------------------	--------	---------------	-------------

Вывести информацию о списываемых токарных станках, если они выпущены ранее 1995 года, в виде:

Списываемые станки	Модель	Мощность, кВт
--------------------	--------	---------------

19. Сформировать и вывести на дисплей массив записей, описывающих следующую таблицу:

ФИО рабочего	Профессия	Разряд	Зарплата, руб.
--------------	-----------	--------	----------------

Вывести список рабочих 3-го разряда в виде:

ФИО рабочего	Профессия
--------------	-----------

20. Сформировать и вывести на дисплей массив записей, описывающих следующую таблицу:

Цех	Выполнение плана, %			
	1 квартал	2 квартал	3 квартал	4 квартал

Вывести список цехов, в которых среднеквартальное выполнение плана составляло меньше 60%, в виде:

Цех	Выполнение плана в среднем за квартал, %
-----	--

21. Сформировать и вывести на дисплей массив записей, описывающих следующую таблицу:

Режущий инструмент	Месячная норма расхода, шт.	Фактический расход за год, шт.
--------------------	-----------------------------	--------------------------------

Вывести список инструментов, годовая потребность в которых превышала плановый расход, в виде:

Режущий инструмент	Годовой расход по факту, шт.
--------------------	------------------------------

22. Сформировать и вывести на дисплей массив записей, описывающих следующую таблицу:

Марка автомобиля	Год выпуска	Стоимость, руб.
------------------	-------------	-----------------

Вывести список автомобилей, выпущенных ранее 2000 г., в виде:

Марка автомобиля	Стоимость, руб.
------------------	-----------------

23. Сформировать и вывести на дисплей массив записей, описывающих следующую таблицу:

ФИО рабочего	Профессия	Разряд	Возраст
--------------	-----------	--------	---------

Вывести список токарей IV разряда пенсионного возраста в виде:

ФИО токаря	Возраст
------------	---------

24. Сформировать и вывести на дисплей массив записей, описывающих следующую таблицу:

Цех	Кол-во рабочих	Режим работы
-----	----------------	--------------

Вывести список цехов, работающих в две и три смены, в виде:

Цех	Кол-во рабочих
-----	----------------

25. Сформировать и вывести на дисплей массив записей, описывающих следующую таблицу:

ФИО рабочего	Специальность	Наименование детали	Годовая программа выпуска, шт.	Фактический выпуск за год, шт.
-----------------	---------------	------------------------	--------------------------------------	--------------------------------------

Вывести список фрезеровщиков, ежемесячно перевыполняющих план на 15% и более, в виде:

ФИО фрезеровщика	Наименование детали	% перевыполнения плана
------------------	---------------------	------------------------

26. Сформировать и вывести на дисплей массив записей, описывающих следующую таблицу:

Станок	Модель	Наименование запасной части	Ресурс работы, час
--------	--------	--------------------------------	--------------------

Вывести список запасных частей шлифовальных станков, срок службы которых составил менее 1 000 часов, в виде:

Модель	Наименование запасной части
--------	-----------------------------

27. Сформировать и вывести на дисплей массив записей, описывающих следующую таблицу:

ФИО рабочего	Наименование детали	Материал заготовки	Брак, %
--------------	---------------------	--------------------	---------

Вывести список рабочих, допустивших выпуск бракованных деталей из цветных материалов и сплавов, в виде:

ФИО рабочего	Наименование детали	Брак, %
--------------	---------------------	---------

28. Сформировать и вывести на дисплей массив записей, описывающих следующую таблицу:

ФИО рабочего	Наименование инструмента	Количество, шт.	Срок сдачи в кладовую
--------------	--------------------------	-----------------	-----------------------

Вывести список рабочих, своевременно не сдавших в кладовую мерительный инструмент, в виде:

ФИО рабочего	Мерительный инструмент	Количество, шт.
--------------	------------------------	-----------------

29. Сформировать и вывести на дисплей массив записей, описывающих следующую таблицу:

ФИО рабочих	Пол	Специальность	Разряд	Количество детей
-------------	-----	---------------	--------	------------------

Вывести список работниц–шлифовщиц III и IV разряда, имеющих двух и более детей, в виде:

ФИО шлифовщицы	Количество детей
----------------	------------------

30. Сформировать и вывести на дисплей массив записей, описывающих следующую таблицу:

Завод	Наименование детали	Программа выпуска, шт.	Затраты на производство, руб.
-------	---------------------	------------------------	-------------------------------

Вывести список заводов, выпустивших аналогичную продукцию с наименьшей себестоимостью, в виде:

Завод	Деталь	Себестоимость выпуска, руб./шт.
-------	--------	---------------------------------

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №8.

СОЗДАНИЕ И РАБОТА СО МНОЖЕСТВЕННЫМИ БАЗАМИ ДАННЫХ ЭЛЕМЕНТОВ ТЕХНИЧЕСКИХ СИСТЕМ

8.1. Понятие типа–множества

Типом–множеством называется совокупность неповторяющихся элементов порядкового типа, представляющих собой множество различных сочетаний элементов исходного множества. При этом максимальное число элементов исходного множества не может превышать 256, а их порядковые номера лежат в диапазоне от 0 до 255. Множество, не содержащее ни одного элемента, называется *пустым* или нуль–множеством и обозначается парой квадратных скобок A:= [].

Для объявления типа–множества используют зарезервированные слова SET и OF, после которых указывают возможные элементы этого множества в виде перечисления или диапазона.

Пример 8.1. Объявить типы–множества, состоящие из букв латинского алфавита и целых чисел от 0 до 9.

```
TYPE LAT=SET OF 'A'..'Z';
```

```
NOM=SET OF 0..9;
```

После объявления множества может быть задана переменная или константа этого типа. Допускается в разделах VAR и CONST вводить множественные переменные или постоянные с заданием типа–множества непосредственно в данных разделах:

```
VAR L:LAT;
```

```
N:NOM;
```

```
CONST N1:NOM=[0,1,2,3];
```

```
CHET:SET OF 0..10=[0,2,4,6,8,10];
```

Объявление множества само по себе не создает в нем каких-либо элементов, а просто информирует компилятор о возможном наборе элементов множества, которые могут быть помещены программой в эту струк-

турную переменную. Множество–переменная может в любой момент содержать 0 или более элементов базового типа.

Чтобы поместить в множество какое–то значение, необходимо построить множественное выражение, а затем назначить его соответствующей переменной.

Множественное выражение (конструктор) может быть просто списком или отрезком из элементов, заключенных в квадратные скобки, или содержать другие выражения, соединенные множественными операторами:

A:=[0,1,2]; V:=[3..6]; C:=A+V;

Над множествами определены операции объединения (+), пересечения (*) и разности (–) .

Дополнительно к этим операциям можно применять две процедуры:

INCLUDE(B, A) – включает новый элемент A в множество B;

EXCLUDE(D, C) – исключает элемент C из множества D.

Кроме того, множества можно сравнивать между собой, используя операторы отношений (=, <>, <=, >=, in).

Пример 8.2. Составить программу, позволяющую выбирать одинаковые по названию детали из двух партий.

```
PROGRAM DETALI;
```

```
USES CRT;
```

```
VAR Z, Y, W: SET OF CHAR;
```

```
    X, V: ARRAY [1..50] OF CHAR;
```

```
    I, N, M: BYTE;
```

```
    BEGIN
```

```
CLRSCR;
```

```
Y:=[]; Z:=[]; W:=[];
```

```
WRITE('ВВЕДИТЕ КОЛ-ВО ДЕТАЛЕЙ В 1 ПАРТИИ:');
```

```
READLN(N);
```

```
FOR I:=1 TO N DO
```

```
    BEGIN
```

```
        WRITE('ВВЕДИТЕ НАЗВАНИЕ', I, '-ОЙ ДЕТАЛИ:');
```

```
        READLN(X[I]);
```

```

        INCLUDE(Y, X[I]);
    END;
CLRSCR;
    WRITE('ВВЕДИТЕ КОЛ-ВО ДЕТАЛЕЙ В 2 ПАРТИИ:');
    READLN(M);

FOR I:=1 TO M DO
BEGIN
    WRITE('ВВЕДИТЕ НАЗВАНИЕ', I, '-ОЙ ДЕТАЛИ:');
    READLN(V[I]);
    INCLUDE(Z, V[I]);
    END;
CLRSCR;
    WRITELN('1 ПАРТИЯ');

FOR I:=1 TO N DO
    WRITE(X[I]:3);
WRITELN;
    WRITELN('2 ПАРТИЯ');

FOR I:=1 TO M DO
    WRITE(V[I]:3);
WRITELN;
WRITELN('ОДИНАКОВЫЕ ДЕТАЛИ');
    FOR I:=1 TO M DO

IF X[I] IN Z THEN
BEGIN
    INCLUDE(W, X[I]);
    WRITE(X[I]:3)
    END;
READKEY
    END.

```

8.2. Задания к практическому занятию

1. На сборочном участке имеются детали, которые используются рабочими для сборки различных изделий: болты, винты, шурупы, шпильки, гайки, шайбы, заклепки, прокладки, манжеты. Задайте количество рабочих-сборщиков и наборы деталей, имеющихся у каждого из них. Постройте и распечатайте:
 - 1.1. Множество деталей, которые используются всеми рабочими;
 - 1.2. Множество деталей, которые имеет хотя бы один рабочий;
 - 1.3. Множество деталей, которые не используются всеми рабочими;
 - 1.4. Два множества: первое множество должно включать в себя детали, используемые всеми рабочими, а второе – детали, используемые хотя бы одним из них.
 - 1.5. Два множества: первое множество должно включать в себя детали, используемые хотя бы одним рабочим, а второе – детали, не используемые всеми рабочими.
 - 1.6. Два множества: первое множество должно включать в себя детали, которые не используются всеми рабочими, а второе – детали, используемые всеми рабочими.
2. В кладовой цеха имеется множество инструментов, которыми может быть обеспечен рабочий: резцы, метчики, плашки, сверла, фрезы, зенкеры, протяжки, развертки, цековки. Задайте количество рабочих и наборы инструментов, имеющихся у каждого из них. Необходимо:
 - 2.1. Построить и распечатать множество, включающее в себя инструменты, которыми обеспечены все рабочие;
 - 2.2. Построить и распечатать множество, включающее в себя инструменты, которые имеет хотя бы один рабочий;
 - 2.3. Построить и распечатать множество, включающее в себя инструменты, которых нет ни у одного рабочего;
 - 2.4. Построить и распечатать два множества: первое множество должно включать в себя инструменты, имеющиеся у всех рабочих, а второе – имеющиеся хотя бы у одного из них;

- 2.5. Построить и распечатать два множества: первое множество должно включать в себя инструменты, имеющиеся хотя бы у одного рабочего, а второе – инструменты, которых нет ни у одного из них;
- 2.6. Построить и распечатать два множества: первое множество должно включать в себя инструменты, которых нет ни у одного рабочего, второе – инструменты, имеющиеся у всех рабочих;
- 2.7. Построить и распечатать три множества: первое должно включать в себя инструменты, имеющиеся у всех рабочих; второе – инструменты, которые есть хотя бы у одного из них; третье – инструменты, которых нет ни у одного рабочего.
3. В гараже предприятия имеются автомобили, которые используются различными цехами и службами завода: ГАЗ, МАЗ, УАЗ, ВАЗ, ЛАЗ, КамАЗ, ЗИЛ, КраЗ, ТАТРА, ФОРД. Задайте количество заводских служб и цехов, а также наборы машин, эксплуатируемых этими подразделениями. Постройте и распечатайте:
- 3.1. Множество автомобилей, которые используются всеми подразделениями;
- 3.2. Множество автомобилей, которые имеет хотя бы одна служба;
- 3.3. Множество автомобилей, которых нет ни в одном цехе;
- 3.4. Два множества: первое множество должно включать в себя машины, работающие во всех службах завода, а второе – эксплуатирующиеся хотя бы в одной из них;
- 3.5. Два множества: первое множество должно включать в себя автомобили, работающие хотя бы в одном цехе, а второе – машины, которых нет ни в одном цехе;
- 3.6. Два множества: первое множество должно включать в себя машины, которые не используются ни одной заводской службой, а второе – автомобили, работающие во всех службах.
4. На складе готовой продукции хранятся корпусные детали (бабки, основания, плиты, планки, станины, пиноли) и тела вращения (валы, оси, шпиндели, диски, кольца, шайбы), изготовленные различными цехами.

Задайте количество цехов и наименования деталей в них изготовленных. Необходимо:

4.1. Построить и распечатать множество корпусных деталей, изготавливаемых всеми цехами;

4.2. Построить и распечатать множество тел вращения, изготавливаемых всеми цехами;

4.3. Построить и распечатать множество корпусных деталей, которые изготавливают хотя бы в одном цехе;

4.4. Построить и распечатать множество тел вращения, которые изготавливают хотя бы в одном цехе;

4.5. Построить и распечатать два множества корпусных деталей и тел вращения, изготавливаемых хотя бы в одном цехе;

4.6. Построить и распечатать два множества корпусных деталей и тел вращения, изготавливаемых всеми цехами;

4.7. Построить и распечатать два множества: первое множество включает в себя корпусные детали, изготавливаемые всеми цехами, а второе – тела вращения, которые изготавливают хотя бы в одном цехе;

4.8. Построить и распечатать два множества: первое множество включает в себя корпусные детали, которые изготавливают хотя бы в одном цехе, а второе – тела вращения, изготавливаемые всеми цехами.

5. В конструкторском бюро имеются следующие чертежные принадлежности: кульманы, циркули, рейсфедеры, транспортиры, линейки, лекала, ватман, карандаши, ластик, фломастеры. Задайте количество конструкторов и наборы чертежных принадлежностей, имеющихся у них. Необходимо:

5.1. Построить и распечатать множество, включающее в себя принадлежности, имеющиеся у всех конструкторов;

5.2. Построить и распечатать множество, включающее в себя принадлежности, которые имеет хотя бы один конструктор;

5.3. Построить и распечатать множество, включающее в себя принадлежности, которых нет ни у одного конструктора.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №9. СОХРАНЕНИЕ РЕЗУЛЬТАТОВ РАСЧЕТОВ ПАРАМЕТРОВ ТЕХНИЧЕСКИХ СИСТЕМ

9.1. Основные понятия о типах–файлах

Рассмотренные ранее (см. практические занятия № 1-8) типы данных не обеспечивают сохранность вводимой информации и полученных результатов для последующего использования их другими программами или этой же программой в другом сеансе работы. Для сохранения информации в языке FREE PASCAL используют файловые типы данных.

Тип–файл – это последовательность компонент, расположенных на внешнем устройстве (обычно на винчестере, дискете или flash–носителе). Компоненты могут быть любого типа, за исключением типа–файла. Число компонент в файле не объявляется.

В FREE PASCAL имеется 3 вида файлов: типизированные, текстовые и файлы без типа.

Для задания **типизированного файла** используют зарезервированные слова FILE и OF, после которых указывают в явном виде тип компонент, составляющих этот файл.

Пример 9.1. Объявить типы–файлы целых чисел и букв латинского алфавита:

```
TYPE NOM=FILE OF INTEGER;  
      LAT=FILE OF 'A'..'Z';
```

Введя тип–файл, можно определить переменные файлового типа:

```
VAR F1,F2: NOM; L: LAT;
```

Все типы–файлы можно объявить и непосредственно в разделе переменных:

```
VAR F1,F2 : FILE OF INTEGER;
```


Стандартный *тип TEXT* определяет тип–файл, который может содержать различные элементы текста (буквы, символы, слова, строки, формулы и т. п.), причем в конце каждой строки стоит признак ее окончания:

```
VAR T: TEXT;
```

При объявлении *файла без типа* не указывают тип составляющих его компонент:

```
VAR F3,F4 : FILE;
```

В этом случае работа с файлом осуществляется на физическом уровне в соответствии с внутренним форматом его данных. При этом компоненты файла могут иметь различный тип.

Наиболее часто на практике применяют текстовые файлы, так как они могут корректно сохранять данные различных типов.

9.2. Процедуры и функции для типов–файлов

После объявления файловой переменной в программе необходимо связать ее с конкретным внешним файлом с помощью процедуры ASSIGN. После этого файл должен быть открыт для чтения и (или) записи информации. В конце программы, после работы с файлом, он должен быть закрыт процедурой CLOSE.

Все стандартные процедуры и функции для работы над типами–файлами делятся на следующие группы: общие для всех типов–файлов; используемые для текстовых файлов; применяемые для типизированных файлов; процедуры и функции для работы над файлами без типа. Рассмотрим подробнее первые три группы.

Общие процедуры и функции для всех типов–файлов:

– ASSIGN (<F>,<ROAD>) – процедура связи файловой переменной с внешним файлом. Здесь F – имя файловой переменной; ROAD – путь к внешнему файлу, представляющий собой переменную или константу типа STRING.

- CLOSE (<F>) – процедура закрытия внешнего файла, с которым связана файловая переменная F.
- ERASE (<F>) – процедура уничтожения (удаления) внешнего файла.
- RENAME (<F>,<NEW NAME>) – процедура переименования внешнего файла. При ее выполнении внешний файл, с которым связана файловая переменная F, получает новое имя, заданное параметром NEW NAME.
- RESET (<F>) – процедура открытия существующего файла. Она открывает существующий файл, с которым связана файловая переменная F, и указатель текущей компоненты файла настраивается на его начало.
- REWRITE (<F>) – процедура открытия нового файла. Она открывает новый пустой файл и присваивает ему имя, заданное процедурой ASSIGN. Если файл с таким именем уже существует, то он уничтожается.
- EOF (<F>) – функция конца файла. Принимает значение TRUE, если указатель текущей компоненты файла находится за последней компонентой файла (за последним символом, если файл текстовый), и FALSE – в противном случае.

Текстовые файлы. При работе с ними параметры, значения которых вводятся и выводятся с помощью процедур READ или WRITE, могут быть не только типа CHAR или STRING, но и других простых типов (целых, вещественных – при вводе; целых, логических, вещественных – при выводе).

Файл типа TEXT может быть открыт либо для чтения процедурой RESET, либо для записи процедурой REWRITE или APPEND.

Наиболее часто для работы с текстовыми файлами используют следующие процедуры и функции:

- APPEND (<F>) – процедура открытия существующего файла, с которым связана файловая переменная, для добавления к его концу информации. При этом указатель текущей компоненты файла настраивается на его конец.

– READ (<F>,<список ввода>) – процедура чтения информации из файла. Из файла, с которым связана файловая переменная F, читаются значения для одной или нескольких переменных списка ввода.

– READLN (<F>,<список ввода>) – процедура чтения строки из файла. Отличается от READ тем, что непрочитанная часть строки, включая признак конца строки, пропускается.

– WRITE (<F>,<список ввода>) – процедура записи информации в файл. В файл, с которым связана файловая переменная F, записываются значения выражений списка вывода.

– WRITELN (<F>,<список ввода>) – процедура записи строки в файл. То же, что и WRITE, но выводимая информация завершается признаком конца строки.

– EOLN (<F>) – функция конца строки файла. Принимает значение TRUE, если текущей компонентой файла является признак конца строки или если функция EOF (<F>) принимает значение TRUE. В остальных случаях функция принимает значение FALSE.

– SEEK EOF (<F>) – функция конца файла. Отличается от EOF (<F>) тем, что стоящие в конце файла символы пробела и табуляции пропускаются.

– SEEK EOLN (<F>) – функция конца строки файла. Отличается от EOLN (<F>) тем, что стоящие в конце строки символы пробела и табуляции пропускаются.

Пример 9.2. Составить программу для определения усилия $P_{и}$, изгибающего оправку с торцовой фрезой под действием составляющих сил резания: $P_{и} = \sqrt{P_y^2 + P_z^2}$, и сохранить результаты расчета на диске D в файле SILA.DAT. Алгоритм программы представлен в приложении Г.10.

```
PROGRAM IZGIB;  
USES CRT;  
VAR PY, PZ, PI: REAL;  
F:TEXT;
```

```

BEGIN;
CLRSCR;
ASSIGN(F, 'D:\SILA.DAT');
REWRITE(F);
WRITE('ВВЕДИТЕ PY:');
READLN(PY);
WRITE('ВВЕДИТЕ PZ:');
READLN(PZ);
PI:=SQRT(SQR(PY)+SQR(PZ));
WRITELN('РЕЗУЛЬТАТЫ РАСЧЕТА');
WRITELN(F, 'РЕЗУЛЬТАТЫ РАСЧЕТА');
WRITELN;
WRITELN(F);
WRITELN('PY=', PY:4:2, 'PZ=', PZ:4:2, 'PI=', PI:4:2 );
WRITELN(F, 'PY=', PY:4:2, 'PZ=', PZ:4:2, 'PI=', PI:4:2 );
CLOSE(F);
READKEY
END.

```

Типизированные файлы. При работе с ними наиболее часто применяют следующие процедуры и функции:

– READ (<F>,<список ввода>) – процедура чтения информации из файла. То же, что и процедура READ для текстовых файлов, но переменные, в которые считывается информация, должны быть того же типа, что и компоненты файла.

– SEEK (<F>,<NOM>) – процедура настройки на требуемую компоненту файла. Осуществляет настройку на компоненту файла, с которым связана файловая переменная F. Компонента файла определяется номером NOM, причем нумерация компонент начинается с нуля.

– TRUNCATE (<F>) – процедура удаления части файла, начиная с текущей позиции. Удаляет часть файла, начиная с текущей позиции и до его конца.

– WRITE (<F>,<список ввода>) – процедура записи информации в файл. То же, что и процедура WRITE для текстовых файлов, но список вывода представляет собой переменные того же типа, что и компоненты файла.

– FILE POS (<F>) – функция номера текущей компоненты файла. Возвращает номер текущей компоненты файла, с которым связана файловая переменная F. Нумерация компонент начинается с нуля.

– FILE SIZE (<F>) – функция текущего размера файла. Возвращает текущий размер файла, с которым связана файловая переменная F, в компонентах этого файла.

Пример 9.3. Вычислить сумму двух матриц из целых чисел размером (2×3) и записать результат в файл MATR.DAT на диске C.

```
PROGRAM SUMMATR;
USES CRT;
VAR A,B,C: ARRAY [1..2,1..3] OF INTEGER;
    I,J: INTEGER;
    SUM: FILE OF INTEGER;
BEGIN
CLRSCR;
FOR I:=1 TO 2 DO
    FOR J:=1 TO 3 DO
        BEGIN
            WRITE('ВВЕДИТЕ A[' ,I,J,']-');
            READLN(A[I,J])
        END;
    FOR I:=1 TO 2 DO
        FOR J:=1 TO 3 DO
            BEGIN
                WRITE('ВВЕДИТЕ B[' ,I,J,']-');
                READLN(B[I,J])
```

```

        END;
    WRITELN;
    ASSIGN(SUM, 'C:\MATR.DAT');
    REWRITE(SUM);
    WRITELN ('РЕЗУЛЬТИРУЮЩАЯ МАТРИЦА C');
    FOR I:=1 TO 2 DO
        BEGIN
            FOR J:=1 TO 3 DO
                BEGIN
                    C[I,J]:=A[I,J]+B[I,J];
                    WRITE(SUM,C[I,J]:3)
                    WRITE(C[I,J]:3,' ');
                END;
            WRITELN(SUM);
            WRITELN
        END;
    CLOSE (SUM);
    WRITELN;
    READKEY
END.

```

9.3. Задания к практическому занятию

Реализовать запись в файл и чтение из файла исходной информации и результатов расчета в соответствии с заданием к практическому занятию №5.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №10.

ПРИМЕНЕНИЕ ПРОЦЕДУР И ФУНКЦИЙ ПРИ РАСЧЕТАХ ПАРАМЕТРОВ ТЕХНИЧЕСКИХ СИСТЕМ

10.1. Понятие процедур и функций

На практических занятиях №4 и 5 многократное повторение некоторой совокупности действий (операторов) на каком-либо этапе выполнения программы реализовывалось при помощи циклических структур. Однако при программировании часто приходится выполнять одну и ту же последовательность действий на различных этапах обработки информации.

В алгоритмах такого рода в различных местах встречаются фрагменты, одинаковые по выполняемым действиям и отличающиеся только значениями исходных данных.

Для более эффективного программирования подобных повторений в языке FREE PASCAL введено понятие *подпрограммы*.

Подпрограмма. Это оформленная в виде самостоятельной программной единицы (модуля) повторяющаяся группа операторов, записанная в тексте программы однократно и вызываемая к выполнению лишь обращением к ней (ссылкой).

Структура любой подпрограммы аналогична структуре всей программы, а сама она должна быть описана до того, как будет использована в программе или другой подпрограмме. Различают две разновидности подпрограмм – процедуры и функции.

Процедура. Предназначена для выполнения какой-то законченной последовательности действий. Любая процедура начинается с заголовка. В отличие от основной программы его наличие здесь обязательно. Он состоит из зарезервированного слова PROCEDURE, за которым следует ее имя, и далее в круглых скобках – список формальных параметров с указанием их типов:

PROCEDURE<имя процедуры> (<список формальных параметров>);

За заголовком могут идти такие же разделы, что и в основной программе. В отличие от нее процедура завершается не точкой, а точкой с запятой.

Пример 10.1. Составить программу для вычисления массы стальной заготовки в форме параллелепипеда при помощи процедуры. Алгоритм программы представлен в приложении Г.11.

```
PROGRAM MASSA_ZAG1;
USES CRT;
VAR A, B, C, G, M: REAL;
    I, N: INTEGER;
PROCEDURE MASSA(X, Y, Z, W: REAL; VAR MAS: REAL);
BEGIN
    MAS:=X*Y*Z*W/1000;
END;
BEGIN
CLRSCR;
WRITE('ВВЕДИТЕ КОЛ-ВО РАСЧЕТОВ');
READLN(N);
FOR I:=1 TO N DO
BEGIN
    WRITELN('РАСЧЕТ -', I);
    WRITELN;
    WRITE('ВВЕДИТЕ ДЛИНУ ЗАГОТОВКИ В ММ:');
    READLN(A);
    WRITE('ВВЕДИТЕ ШИРИНУ ЗАГОТОВКИ В ММ:');
    READLN(B);
    WRITE('ВВЕДИТЕ ВЫСОТУ ЗАГОТОВКИ В ММ:');
    READLN(C);
    WRITE('ВВЕДИТЕ ПЛОТНОСТЬ ЗАГОТОВКИ В КГ/М ^3:');
    READLN(G);
    MASSA(A, B, C, G, M);
    WRITELN('МАССА РАВНА', M:3:3, 'КГ');
```



```
END;  
READKEY  
END.
```

Наряду с процедурами, получающими параметры тем или иным способом, в FREE PASCAL допустимы процедуры, которые вообще не получают параметров (процедуры без параметров).

Пример 10.2. Записать процедуру для вывода на печать заголовка таблицы:

```
PROCEDURE ZAGTABL;  
BEGIN  
  WRITELN ('СПИСОК РАБОЧИХ ЦЕХА');  
  WRITELN ;  
  WRITELN ('№ n/n': 3, 'Ф.И.О.': 20, 'ПРОФЕССИЯ': 15, 'РАЗРЯД': 3);  
END;
```

Для вызова процедуры из основной программы или другой подпрограммы записывают оператор, состоящий из имени процедуры и списка фактических параметров, которые должны совпадать по количеству и типам с формальными параметрами процедуры:

```
MASSA (A, B, C, M); ZAGTABL; .
```

Как правило, процедуры используют для решения задач двух типов:

1. Когда требуется выполнить некоторую последовательность действий, не возвращая результирующего значения.
2. Когда необходимо изменить значения одного или нескольких фактических параметров.

Функция. Предназначена для вычисления какого-либо единственного параметра. Она является частным случаем процедуры и отличается от нее следующим:

1. Заголовок функции состоит из слова FUNCTION, после которого следует ее имя, далее в круглых скобках – список формальных параметров, а затем через двоеточие записывается тип функции, т. е. тип возвращаемого параметра. Возвращаемый параметр может быть любого порядкового, вещественного или строкового типа:

FUNCTION <имя функции> (<список параметров>):<тип функции >;

2. Результатом выполнения функции является одно значение, а у процедуры одно или несколько.

3. Результат выполнения функции передается в основную программу как значение имени этой функции, а результат выполнения процедуры как значение ее параметров.

4. В теле функции хотя бы раз имени функции должно быть присвоено значение.

Пример 10.3. Усовершенствовать программу из примера 10.1, используя для вычисления массы заготовки функцию.

```
PROGRAM MASSA_ZAG1;
USES CRT;
VAR A, B, C, G, M: REAL;
    I, N: INTEGER;
FUNCTION MASSA(X, Y, Z, W: REAL): REAL;
BEGIN
    MASSA:=X*Y*Z*W/1000;
END;
BEGIN
CLRSCR;
WRITE('ВВЕДИТЕ КОЛ-ВО РАСЧЕТОВ');
READLN(N);
FOR I:=1 TO N DO
BEGIN
    WRITELN('РАСЧЕТ -', I);
    WRITELN;
    WRITE('ВВЕДИТЕ ДЛИНУ ЗАГОТОВКИ В ММ:');
    READLN(A);
    WRITE('ВВЕДИТЕ ШИРИНУ ЗАГОТОВКИ В ММ:');
    READLN(B);
    WRITE('ВВЕДИТЕ ВЫСОТУ ЗАГОТОВКИ В ММ:');
    READLN(C);
```

```

WRITE('ВВЕДИТЕ ПЛОТНОСТЬ ЗАГОТОВКИ В КГ/М ^3:');
READLN(G);
    M:=MASSA(A, B, C, G);
    WRITELN('МАССА ПАВНА', M:3:3, 'КГ');
END;
READKEY
END.

```

Для вызова функции из основной программы или другой подпрограммы следует в выражении, где необходимо использовать значение функции, указать ее имя со списком фактических параметров, которые должны совпадать по количеству и типам с формальными параметрами функции: $M := \text{MASSA}(A, B, C);$.

10.2. Параметры процедур и функций

Все параметры, которые использует подпрограмма, можно разбить на две группы: локальные параметры, объявленные внутри подпрограммы и доступные только ей самой, и глобальные – объявленные в основной программе и доступные как программе, так и всем ее подпрограммам. Обмен информацией между основной программой и подпрограммой может осуществляться только с помощью глобальных параметров.

Подпрограмма может использовать глобальные параметры, или непосредственно обращаясь к ним по имени, или используя механизм формальных параметров. Подпрограмма может применять любые глобальные параметры за исключением тех, которые имеют те же имена, что и ее локальные параметры.

Формальные параметры подпрограммы указывают, с какими параметрами следует к ней обращаться (количество параметров, их последовательность, типы). Они задаются в заголовке подпрограммы в виде списка, разбитого на группы, разделенные точками с запятой. При этом в группу формальных параметров включаются однотипные параметры одной категории. Все формальные параметры условно разделяют на 5 категорий: па-

раметры–значения, параметры–переменные, параметры–константы, параметры–процедуры и параметры–функции.

Параметры–значения. Используются основной программой для передачи исходных данных в подпрограмму (процедуру и функцию), поэтому их значения в основной программе подпрограмма изменить не может. Параметр–значение обозначается в заголовке подпрограммы своим именем с указанием через двоеточие типа. Тип параметра–значения может быть любым, за исключением файлового.

Если параметров–значений одного типа несколько, то их можно объединить в одну группу, перечислив их имена через запятую, а затем уже указать общий тип. При этом отдельные группы параметров отделяются друг от друга точкой с запятой, например:

```
PROCEDURE VWU (M, N: REAL; I, J: INTEGER);
```

```
FUNCTION ABC (I1, J2: INTEGER; R: REAL): REAL;
```

Параметры–переменные. Используются для определения результатов выполнения подпрограммы. При передаче параметров–переменных в подпрограмму передаются их адреса в порядке, объявленном в заголовке подпрограммы. При этом подпрограмма имеет к ним доступ и может их изменять. Параметр–переменная указывается в заголовке подпрограммы аналогично параметру–значению, но только перед именем параметра записывается зарезервированное слово `VAR`, действие которого распространяется до ближайшей точки с запятой, т. е. в пределах одной группы. Тип параметров–переменных может быть любым, включая и файловый. При вызове подпрограммы на месте параметра–переменной в качестве фактического параметра должна использоваться переменная идентичного типа, например:

```
PROCEDURE MAX (A: REAL; VAR B, C: INTEGER; D: WORD); .
```

Часто в качестве параметра в подпрограмму следует передать ту или иную переменную, но изменять ее подпрограмма не должна. В этом случае нежелательно передавать этот параметр как параметр–переменную. Можно его передать как параметр–значение, однако, если эта переменная имеет большой размер, то копия такого параметра займет большую часть

стека и может его переполнить. В этом случае параметр лучше передать как параметр–константу. Такой параметр, если он структурированного типа, передается своим адресом с защитой от его изменения.

Параметр–константа. Указывается в заголовке подпрограммы аналогично параметру–значению, но перед именем параметра записывается зарезервированное слово `CONST`, действие которого распространяется только в пределах одной группы типов. Тип параметра–значения может быть любым за исключением файлового. При вызове подпрограммы на месте параметра–переменной в качестве фактического параметра может использоваться любое выражение совместного для присваивания типа, не содержащее файловую компоненту. Параметр–константу нельзя передавать в другую подпрограмму в качестве фактического параметра, например:

```
FUNCTION NEW (CONST S: STRING): STRING;
```

Передаваемым параметром в подпрограмме может быть также параметр–процедура или параметр–функция, т. е. параметр процедурного типа. Фактически этот параметр является параметром–значением, так как записывается без зарезервированного слова `VAR`. В качестве фактического параметра в этом случае используется соответствующая процедура или функция, имеющая необходимое количество параметров требуемых типов. При этом для параметров–процедур и параметров–функций существуют те же правила, что и для других переменных процедурного типа, например: `TYPE FUNC=FUNCTION (X, Y: INTEGER): INTEGER;`

```
-----  
PROCEDURE ABC (A, B: INTEGER; C: FUNC);
```

10.3. Задания к практическому занятию

Составить программу, реализующую вычисления функции или процедуры в соответствии с вариантом заданий к практическому занятию №7.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №11.

УСОВЕРШЕНСТВОВАННЫЕ ВАРИАНТЫ ОТОБРАЖЕНИЯ РЕЗУЛЬТАТОВ ПРОЕКТИРОВАНИЯ ТЕХНИЧЕСКИХ СИСТЕМ

11.1. Возможности модуля CRT при организации интерфейса в текстовом режиме

Модуль CRT включает в себя подпрограммы (функции и процедуры), с помощью которых ввод и вывод информации при программировании на языке FREE PASCAL реализуется более наглядно и интересно.

Для организации интерфейса в текстовом режиме наиболее часто используются следующие функции и процедуры модуля CRT:

1. **Функция KEYPRESSED** – возвращает результат TRUE (истина), если на клавиатуре была нажата клавиша, и FALSE (ложь) в противном случае. Не задерживает исполнение программы.

Цикл REPEAT – UNTIL KEYPRESSED с использованием этой функции будет безостановочно работать, пока не будет нажата какая – либо клавиша. Полезная роль цикла REPEAT – UNTIL KEYPRESSED состоит в отсрочке завершения программы, а именно, замене окна пользователя с результатами окном редактора.

2. **Функция READKEY** – читает символ с клавиатуры без эхо – повтор на экране. Она приостанавливает исполнение программы до нажатия на любую клавишу, кроме SHIFT, CTRL, ALT, CAPSLOCK, NUMLOCK или SCROLLLOCK.

3. Процедура TEXTCOLOR – определяет цвет выводимых символов.

Заголовок процедуры :

PROCEDURE TEXTCOLOR(COLOR:BYTE);,

где COLOR – код цвета типа Byte, который можно установить с помощью констант цвета.

Например:

CONST

BLACK=0; {черный}

BLUE=1; {темно-синий}

GREEN=2; {темно-зеленый}

CYAN=3; {бирюзовый}

RED=4; {красный}

MAGENTA=5; {фиолетовый}

BROWN=6; {коричневый}

LIGHTBLUE=9; {синий}

YELLOW=14; {желтый}

WHITE=15; {белый}.

4. Процедура TEXTBACKGROUND – определяет цвет фона.

Заголовок процедуры:

PROCEDURE TEXTBACKGROUND (COLOR:BYTE);.

5. Процедура CLRSCR – очищает экран или окно.

6. Процедура WINDOW – определяет текстовое окно (область экрана), которое в дальнейшем будет рассматриваться процедурами вывода как весь экран.

Заголовок процедуры:

PROCEDURE WINDOW (X1,Y1,X2,Y2:BYTE);.

где X_1, Y_1 – координаты верхнего левого угла окна; X_2, Y_2 – координаты нижнего правого угла окна. Эти координаты задаются в координатах экрана. Левый верхний угол экрана имеет координаты (1;1). Горизонтальная координата X увеличивается слева направо, а вертикальная Y – сверху вниз. Координаты (границы) экрана в текстовом режиме могут меняться в пределах от 1 до 80 по оси X и от 1 до 25 по оси Y и задаются в символах.

7. Процедура GOTOXY – переводит курсор в нужное место экрана или текущего окна.

Заголовок процедуры:

PROCEDURE GOTOXY (X,Y:BYTE);,

где X, Y – новые координаты курсора, которые задаются относительно границ экрана или окна. Запись в тексте программы GOTOXY (1,1) переведет курсор в верхний левый угол экрана или окна, если это окно ранее определено. Обращение к данной процедуре будет проигнорировано в случае, если координаты будут выходить за границы экрана или окна.

8. Процедура SOUND – определяет звуковой сигнал динамика компьютера с нужной частотой.

Заголовок процедуры:

PROCEDURE SOUND (F:WORD);,

где F – частота звука в Гц, переменная типа WORD.

9. Процедура DELAY – обеспечивает задержку работы программы на заданный интервал времени.

Заголовок процедуры:

PROCEDURE DELAY (T:WORD);,

где T – интервал времени в мс, в течение которого задерживается выполнение следующего оператора программы. Переменная T устанавливается типом данных WORD.

10. Процедура NOSOUND – выключает динамик после активизации процедуры SOUND.

Пример 11.1. Усовершенствовать программу из примера 4.2, реализовав ввод исходных данных и вывод результата в отдельных окнах с различным цветом фона, расположенных одно за другим. Предусмотреть различный цвет вводимых и выводимых данных, звуковой сигнал по завершению работы программы.

```
PROGRAM SRED_RAZM_M;  
USES CRT;  
  VAR I, N: INTEGER;  
      Y: BYTE;  
  SUM,SR,R: REAL;  
BEGIN  
  CLRSCR; {очистка экрана}  
  TEXTBACKGROUND (0); {установка цвета фона полно-  
го экрана}  
  WINDOW (2,2,40,14); {создание окна}  
  TEXTBACKGROUND (1); {создание цвета фона окна}  
  TEXTCOLOR (4); {задание цвета текста}  
  GOTOXY (7,2); {задание положения курсора}  
  WRITE ('ИСХОДНЫЕ ДАННЫЕ');  
  TEXTCOLOR (15);  
  GOTOXY (2,4);  
  WRITE ('Введите количество деталей');  
  GOTOXY (2,5);  
  WRITE ('в партии:');  
  READ (N);  
  SUM:= 0;Y:= 7;  
  FOR I:=1 TO N DO  
  BEGIN
```

```

GOTOXY (2,Y);
WRITE ('Введите размер ',I,'-й детали в мм:');
READ (R);
SUM:= SUM+R;
Y:= Y+1;
END;
SR:= SUM/N;
WINDOW (42,2,80,14);
TEXTBACKGROUND (15);
TEXTCOLOR (4);
WRITELN ('РЕЗУЛЬТАТ РАСЧЕТА');
WRITELN;
TEXTCOLOR (1);
WRITELN ('Средний арифметический размер');
WRITE (' деталей в партии равен ',SR:4:3,' мм.');
```

SOUND (600); {задание частоты звукового сигнала}
 DELAY (1000); {задержка исполнения процедуры NOSOUND}
 NOSOUND; {отмена звука}
 READKEY {приостановка исполнения программы}
 END.

11.2. Задания к практическому занятию

Составить программу в соответствии с заданием к практическому занятию №7, обязательно предусмотрев в ней табличный ввод исходных данных и вывод результата в отдельных окнах с различным цветом фона, расположенных одно под другим. Реализовать различный цвет вводимых и выводимых данных, звуковой сигнал с изменяющимся тоном по окончании вводимых данных и завершении работы программы.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №12.

СОЗДАНИЕ ГРАФИЧЕСКИХ МОДУЛЕЙ ЭЛЕМЕНТОВ ТЕХНИЧЕСКИХ СИСТЕМ

12.1. Возможности модуля GRAPH при организации графического интерфейса

Модуль GRAPH включает в себя большое количество подпрограмм (функций и процедур) и предоставляет пользователю широкие возможности управления графическим экраном для построения различных черно – белых или цветных статических или движущихся изображений, диаграмм, графиков при программировании в среде FREE PASCAL.

Для организации интерфейса в графическом режиме наиболее часто используют следующие функции и процедуры модуля GRAPH:

1. **Процедура INITGRAPH** – иницирует графический режим работы адаптера.

Заголовок процедуры:

```
PROCEDURE INITGRAPH (VAR DRIVER,MODE:INTEGER;  
                      PATH:STRING);,
```

где DRIVER – переменная типа INTEGER, которая определяет тип графического драйвера (служебной программы), задаваемый в виде констант, например:

```
CONST  
    EGA=3;  
    VGA=9; ,
```

где EGA, VGA обозначают соответственно цветной усиленный графический адаптер и графический видеомассив.

В случае, если тип графического драйвера, соответствующий типу графического адаптера, не известен, может быть установлен режим автоопределения типа:

```
CONST
```

```
    DETECT=0; .
```

MODE – переменная типа INTEGER, которая определяет режим работы графического адаптера, задаваемый в виде констант, например:

```
CONST
```

```
    EGAHI=1; {адаптер EGA, разрешение экрана 640×350 пикселей, 16  
    цветов}
```

```
    VGAHI=2; {адаптер VGA, разрешение экрана 640×480 пикселей}; .
```

PATH – выражение типа STRING, которое должно содержать имя файла драйвера (служебной программы) с маршрутом его поиска.

В большинстве случаев, если тип адаптера персонального компьютера заранее неизвестен, можно обратиться к процедуре INITGRAPH с требованием автоматического определения типа драйвера и режима работы графического адаптера:

```
USES GRAPH;
```

```
VAR DRIVER, MODE:INTEGER;
```

```
BEGIN
```

```
    DRIVER := DETECT;
```

```
    INITGRAPH (DRIVER, MODE, '<имя файла драйвера  
    с маршрутом его поиска>'); .
```

При программировании инициация графического режима работы адаптера обязательна.

2. Функция **GRAPHRESULT** – возвращает код ошибки для последней графической операции. Позволяет определить правильно или неправильно настроен графический режим. В случае отсутствия ошибок настройки графического режима устанавливается код GROK = 0, в противном случае – код от -1 до -14.

3. **Функция GRAPHERRORMSG** – возвращает строку сообщения об ошибке для заданного кода. Выводит расшифровку ошибки, допущенной при настройке графического режима.

Заголовок функции:

FUNCTION GRAPHERRORMSG (CODE: INTEGER): STRING; ,
где CODE – код ошибки.

Инициацию графического режима с автоматическим определением типа драйвера и режима работы графического адаптера при программировании на языке FREE PASCAL можно представить следующим образом:

```
USES GRAPH;
VAR DRIVER, MODE, ERROR: INTEGER;
BEGIN
  DRIVER := DETECT;
  INITGRAPH (DRIVER, MODE, 'D:\TP7\BGI\EGAVGA.BGI');
  ERROR := GRAPHRESULT;
  IF ERROR<>GROK THEN
    BEGIN
      WRITELN(GRAPHERRORMSG (ERROR));
      .....
    END
  ELSE
    .....
  .
```

4. **Процедура CLOSEGRAPH** – завершает работу адаптера в графическом режиме и восстанавливает текстовый режим работы экрана.

Заголовок процедуры:

PROCEDURE CLOSEGRAPH; .

При программировании использование этой процедуры обязательно. Как правило, CLOSEGRAPH устанавливается перед зарезервированным словом «END.»(конец программы).

5. Функции GETMAXX и GETMAXY – автоматически определяют максимальную горизонтальную или вертикальную координату графического экрана.

Любая координата на графическом экране при программировании на языке FREE PASCAL задается в пикселах (точках) относительно левого верхнего угла, который имеет координаты (0;0). Горизонтальная координата экрана увеличивается слева направо, а вертикальная – сверху вниз. При программировании использование этих функций позволит правильно отобразить местоположение объекта на экране.

6. Процедура PUTPIXEL – выводит заданным цветом точку по указанным координатам.

Заголовок процедуры:

PROCEDURE PUTPIXEL (X,Y: INTEGER; COLOR:WORD); ,

где X, Y – координаты точки; COLOR – код цвета точки (см. параграф 11.1).

7. Процедура LINE – вычерчивает линию с указанными координатами начала и конца.

Заголовок процедуры:

PROCEDURE LINE (X1, Y1, X2, Y2: INTEGER); ,

где X1, Y1; X2, Y2 – соответственно координаты начала и конца линии.

8. Процедура SETLINESTYLE – устанавливает стиль вычерчиваемых линий.

Заголовок процедуры:

PROCEDURE SETLINESTYLE (TYPE, PATTERN, THICK:
WORD); ,

где TYPE – тип линии, который можно задать с помощью одной из следующих констант.

Например:

CONST SOLIDLN = 0; {сплошная}

DOTTEDLN = 1; {точечная}

CENTERLN = 2; {штрих-пунктирная}

DASHEDLN = 3; {пунктирная}; .

PATTERN – образец линии. В большинстве случаев задается кодом 0.

THICK – толщина линии. Задается с помощью одной из следующих констант:

CONST

NORMWIDTH = 1; {один пиксел}

THICKWIDTH = 3; {три пиксела} .

9. Процедура RECTANGLE – вычерчивает прямоугольник с указанными координатами углов.

Заголовок процедуры:

PROCEDURE RECTANGLE (X1, Y1, X2, Y2:INTEGER); ,

где X1, Y1; X2, Y2 – соответственно координаты левого верхнего и правого нижнего углов прямоугольника.

10. Процедура CIRCLE – вычерчивает окружность.

Заголовок процедуры:

PROCEDURE CIRCLE (X, Y: INTEGER; R: WORD); ,

где X, Y – координаты центра окружности; R – радиус окружности.

11. Процедура ARC – предназначена для вычерчивания дуги окружности.

Заголовок процедуры:

PROCEDURE ARC (X,Y:INTEGER;BEGA,ENDA,R:WORD); ,

где X, Y – координаты центра дуги окружности; BEGA, ENDA – начальный и конечный углы дуги окружности (отсчитываются против часовой стрелки; указываются в градусах; нулевой угол соответствует горизонтальному направлению радиуса – вектора слева направо); R – радиус дуги окружности.

12. Процедура ELLIPSE – позволяет вычертить эллипсную дугу.

Заголовок процедуры:

PROCEDURE ELLIPSE (X,Y:INTEGER;BEGA,ENDA,RX,RY:
WORD); ,

где X, Y – координаты центра дуги эллипса; BEGA, ENDA – соответственно начальный и конечный углы дуги эллипса, град.; RX, RY – горизонтальный и вертикальный радиусы эллипса.

13. Процедура SETCOLOR – устанавливает текущий цвет для выводимых линий и символов.

Заголовок процедуры:

PROCEDURE SETCOLOR (COLOR: WORD); ,

где COLOR – код текущего цвета (см. параграф 11.1).

14. Процедура SETBKCOLOR – устанавливает цвет фона.

Заголовок процедуры:

PROCEDURE SETBKCOLOR (COLOR: WORD); ,

где COLOR – код цвета фона (см. параграф 11.1).

15. Процедура SETFILLSTYLE – устанавливает стиль (тип и цвет) заполнения.

Заголовок процедуры:

PROCEDURE SETFILLSTYLE (FILL, COLOR: WORD); ,

где FILL – тип заполнения. Задается с помощью одной из следующих констант:

CONST

EMPTYFILL = 0; {фон, узоры отсутствуют}

SOLIDFILL = 1; {сплошное заполнение}

LTSLASHFILL = 3; {штриховка /////}

LTBKSLASHFILL = 6; {штриховка \\\\) .

COLOR – код цвета заполнения (см. параграф 11.1).

16. Процедура FLOODFILL – заполняет произвольную замкнутую фигуру, используя текущий стиль заполнения (узор и цвет).

Заголовок процедуры:

PROCEDURE FLOODFILL (X, Y: INTEGER; BORDER: WORD); ,

где X, Y – координаты любой точки внутри замкнутой фигуры; BORDER – цвет граничной линии (см. параграф 11.1).

17. Процедура FILLELLIPSE – обводит линией и заполняет эллипс.

Заголовок процедуры:

PROCEDURE FILLELLYPS (X, Y, RX, RY: INTEGER); ,

где X, Y – координаты центра; RX, RY – горизонтальный и вертикальный радиусы эллипса.

18. Процедура BAR – заполняет прямоугольную область экрана.

Заголовок процедуры:

PROCEDURE BAR (X1, Y1, X2, Y2: INTEGER); ,

где X1, Y1; X2, Y2 – соответственно координаты левого верхнего и правого нижнего углов закрашиваемой области.

19. Процедура OUTTEXT – выводит текстовую строку, начиная с текущего положения указателя.

Заголовок процедуры:

PROCEDURE OUTTEXT (TXT: STRING); ,

где TXT – выводимая строка.

20. Процедура OUTTEXTXY – выводит строку, начиная с заданного места.

Заголовок процедуры:

Procedure OUTTEXTXY (X, Y: INTEGER; TXT: STRING); ,

где X,Y – координаты точки начала вывода строки; TXT – выводимая строка.

21. Процедура SETTEXTSTYLE – устанавливает текущий шрифт, стиль и размер текста.

Заголовок процедуры:

Procedure SETTEXTSTYLE (FONT, DIRECT, SIZE: WORD); ,

где FONT – код шрифта. Задается с помощью одной из следующих констант, например:

CONST

DEFAULTFONT = 0; {точечный шрифт}

SMALLFONT = 2; {уменьшенный шрифт}

SANSSERIFFONT = 3; {прямой шрифт}

GOTHICFONT = 4; {готический шрифт} .

DIRECT – код направления шрифта. Задается одной из двух констант:

CONST

HORIZDIR = 0; {слева направо}

VERTDIR = 1; {снизу вверх};

SIZE – код размера шрифта. Задается в диапазоне от 1 до 10. Для точечного шрифта – в диапазоне от 1 до 32. Минимальный размер шрифта, при котором отчетливо различаются все его детали, равен 4 (для точечного – 1).

22. Процедура SETTEXTJUSTIFY – задает выравнивание выводимого текста по отношению к текущему положению указателя или к заданным координатам.

Заголовок процедуры:

PROCEDURE SETTEXTJUSTIFY (HORIZ, VERT: WORD); ,

где HORIZ, VERT – соответственно горизонтальное и вертикальное выравнивание. Под выравниванием понимают возможность размещения текста левее или правее указанного места, выше, ниже или по центру.

Для задания выравнивания можно использовать следующие константы:

CONST

LEFTTEXT = 0; {указатель слева от текста}

CENTERTEXT = 1; {симметрично слева и справа, сверху и снизу}

RIGHTTEXT = 2; {указатель справа от текста}.

Пример 12.1. Составить программу графического отображения корпусной детали представленной на рисунке 12.1.

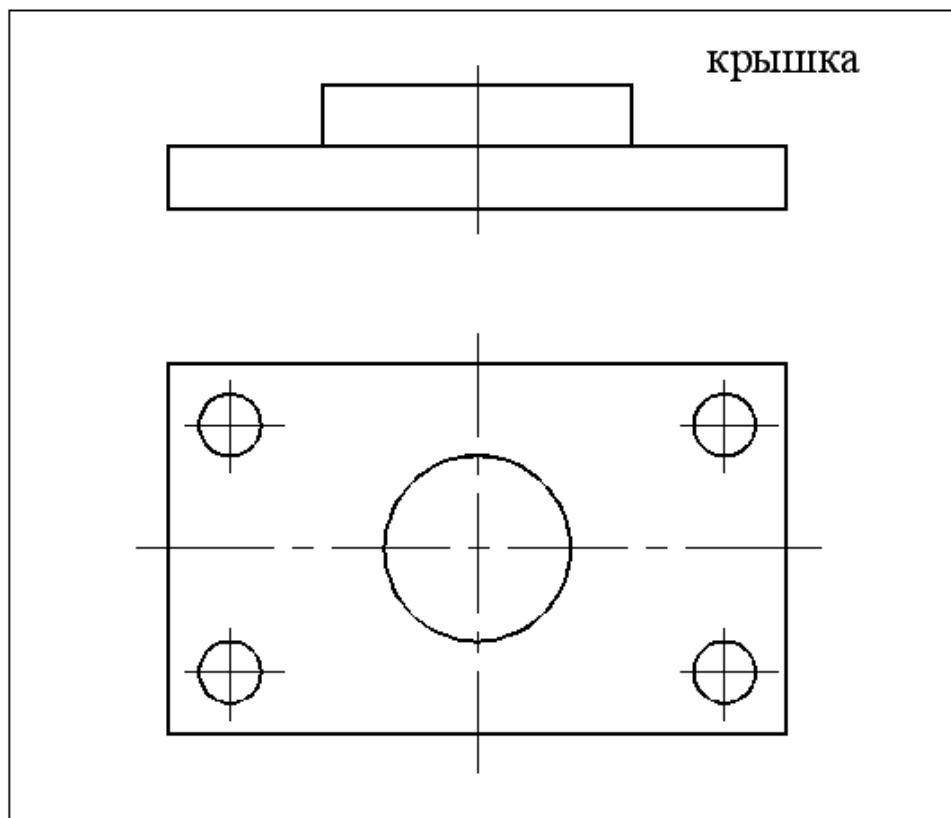


Рис. 12.1. Эскиз корпусной детали (крышка)

Для удобства программирования графического отображения корпусной детали (см. рис. 12.1) можно реализовать наложение этой детали на координатную сетку (рис. 12.2) и с учетом соотношений длин сторон и диаметров окружностей, если их размеры не заданы, установить координаты узловых точек, по которым будет осуществлено графическое построение этой детали с помощью программы, представленной ниже.

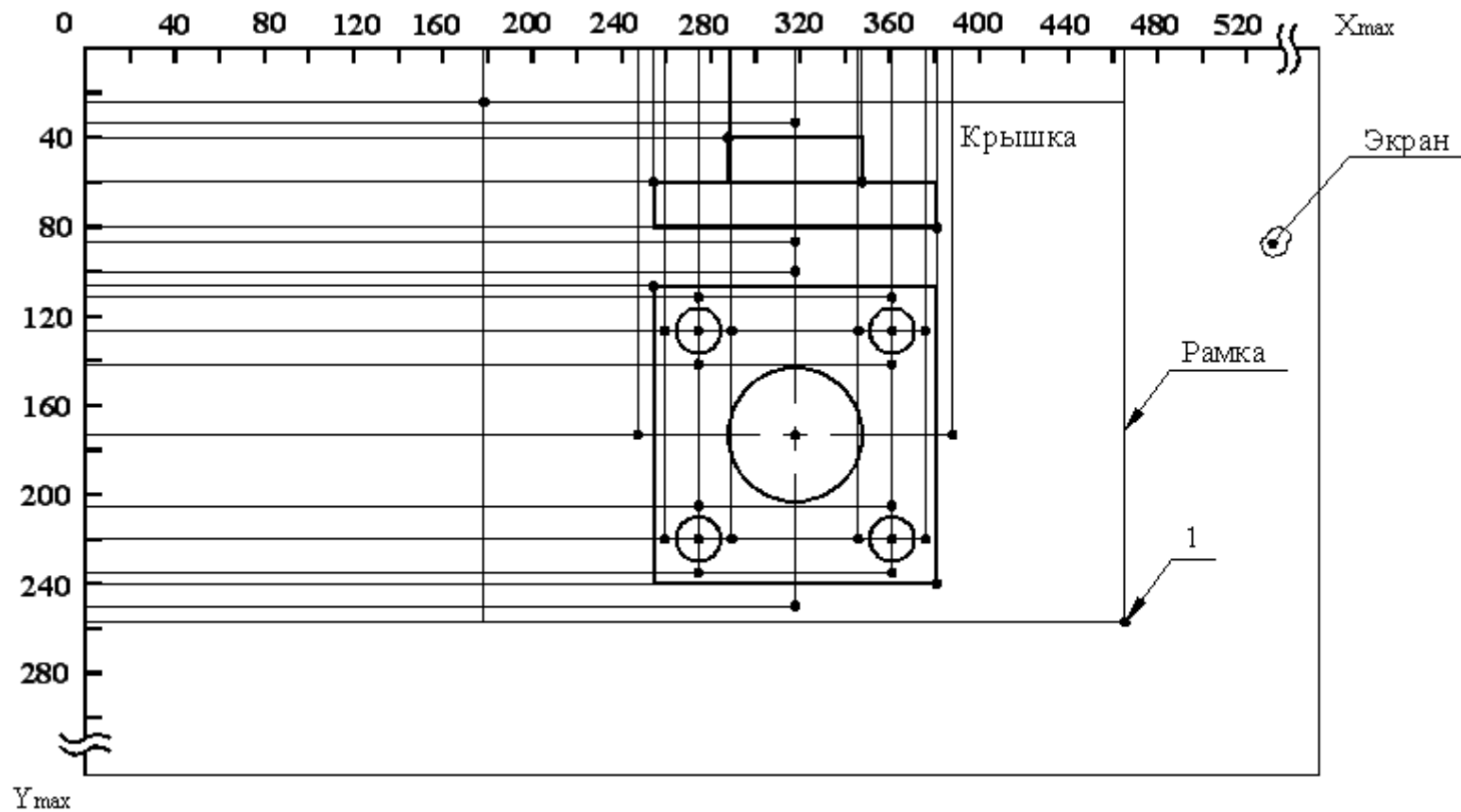


Рис. 12.2. Корпусная деталь в координатной сетке экрана: X_{max} , Y_{max} – координаты, устанавливающие максимальный размер (разрешение) графического экрана в пикселах (например, 640 x 480):
1 – узловая точка


```

PROGRAM DETAL;
USES GRAPH;
    VAR DRIVER, MODE, ERROR: INTEGER;
BEGIN {инициация графического режима с проверкой наличия ошибок
        инициации}
    DRIVER := DETECT;
    INITGRAPH (DRIVER, MODE, 'EGAVGA.BGI');
    ERROR := GRAPHRESULT;
    IF ERROR<>GROK THEN WRITELN (GRAPHERRORMSG (ERROR))
        ELSE
            BEGIN
                BAR (0, 0, 640, 480); {задание области экрана}
                SETBKCOLOR (15); {задание фона экрана}
                SETCOLOR (1); {задание цвета линий}
                    {отображение рамки}
                SETLINESTYLE (0, 0, 1); {задание типа линий рамки}
                RECTANGLE (200, 20, 450, 280); {графическое отображение рамки}
                    {графическое отображение главного вида крышки}
                SETLINESTYLE (0, 0, 3); {задание типа основных линий}
                RECTANGLE (290, 40, 350, 60); {графическое отображение верхне-
                    го прямоугольника}
                RECTANGLE (260, 60, 380, 80); {графическое отображение нижне-
                    го прямоугольника}
                SETLINESTYLE (2, 0, 1); {задание типа осевых линий}
                LINE (320, 35, 320, 85); {графическое отображение осевой линии}
                    {графическое отображение вида сверху крышки}
                SETLINESTYLE (0, 0, 3); {задание типа основных линий}
                RECTANGLE (260, 100, 380, 260); {графическое отображение пря-
                    моугольника}
                CIRCLE (320, 180, 30); {графическое отображение окружности,
                    расположенной по центру}
            END
    END

```

```

CIRCLE (280, 120, 8);{графическое отображение окружности в
                        верхнем левом углу}
CIRCLE (360, 120, 8);{графическое отображение окружности в
                        верхнем правом углу}
CIRCLE (280, 240, 8);{графическое отображение окружности в
                        нижнем левом углу}
CIRCLE (360, 240, 8);{графическое отображение окружности в
                        нижнем правом углу}
SETLINESTYLE(2,0,1);{задание типа осевых линий}
LINE (320, 95, 320, 265); {графическое отображение осевых линий}
LINE (255, 180, 385, 180); {окружности, расположенной по центру}
LINE (280, 102, 280, 138); {графическое отображение осевых линий}
LINE (262, 120, 298, 120); {окружности, в верхнем левом углу}
LINE (360, 102, 360, 138); {графическое отображение осевых линий}
LINE (342, 120, 378, 120); {окружности, в верхнем правом углу}
LINE (280, 222, 280, 258); {графическое отображение осевых линий}
LINE (262, 240, 298, 240); {окружности, в нижнем левом углу}
LINE (360, 222, 360, 258); {графическое отображение осевых линий}
LINE (342, 240, 378, 240); {окружности, в нижнем правом углу}
{графическое отображение надписи названия детали – “КРЫШКА”}
  SETTEXTSTYLE (3, 0, 5); {задание типа надписи}
  OUTTEXTXY (380, 35, 'КРЫШКА');{графическое отображение
                                названия детали в верхнем правом углу рамки}
  READLN;
  CLOSEGRAPH;{закрытие графического режима}
  END;
END.

```

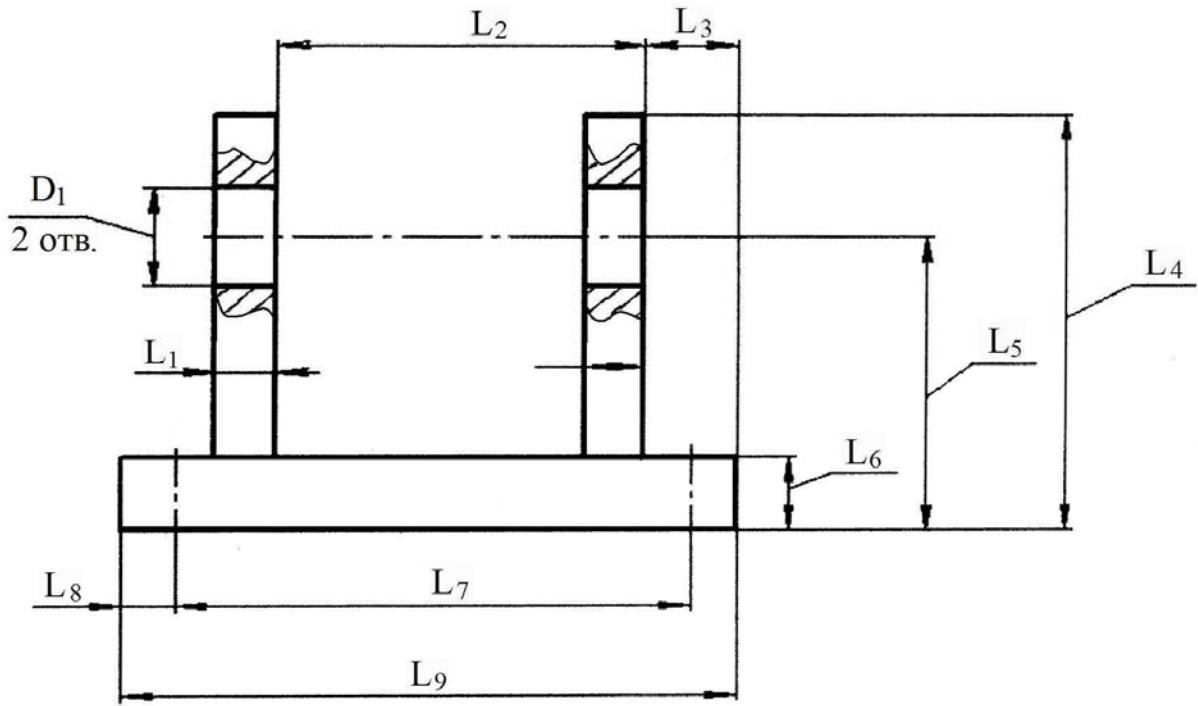
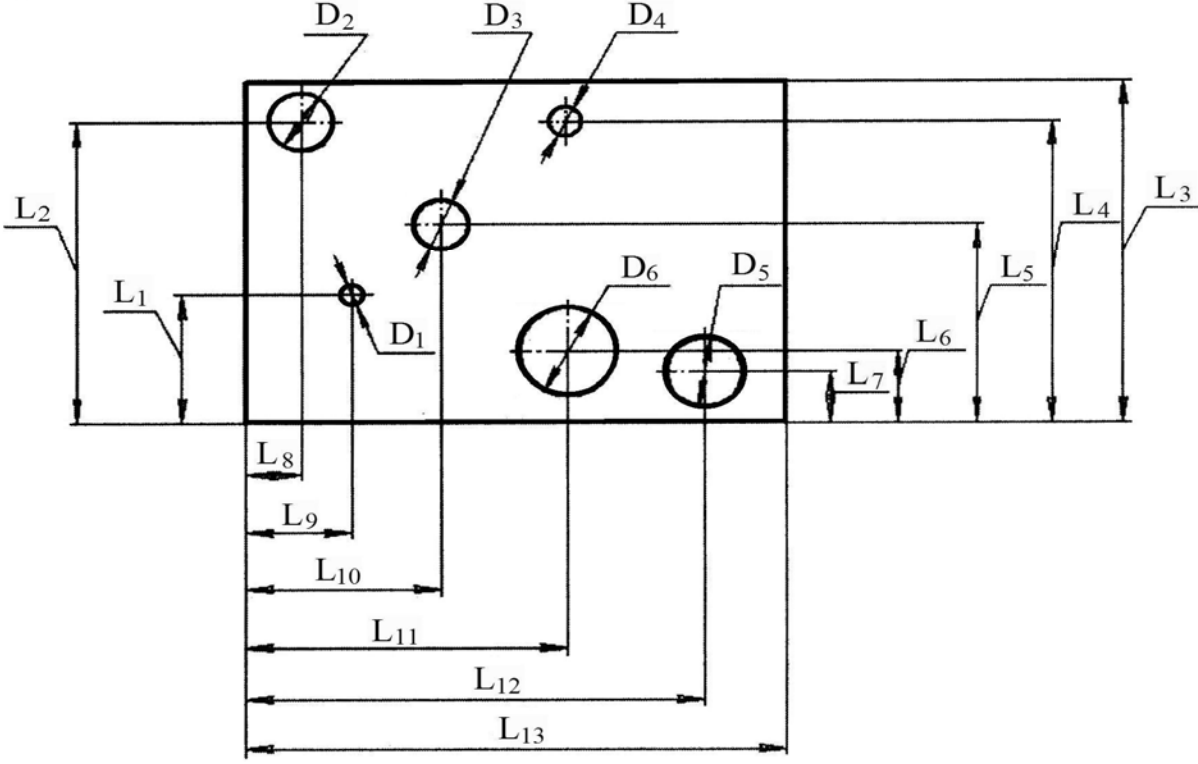
12.2. Задания к практическому занятию

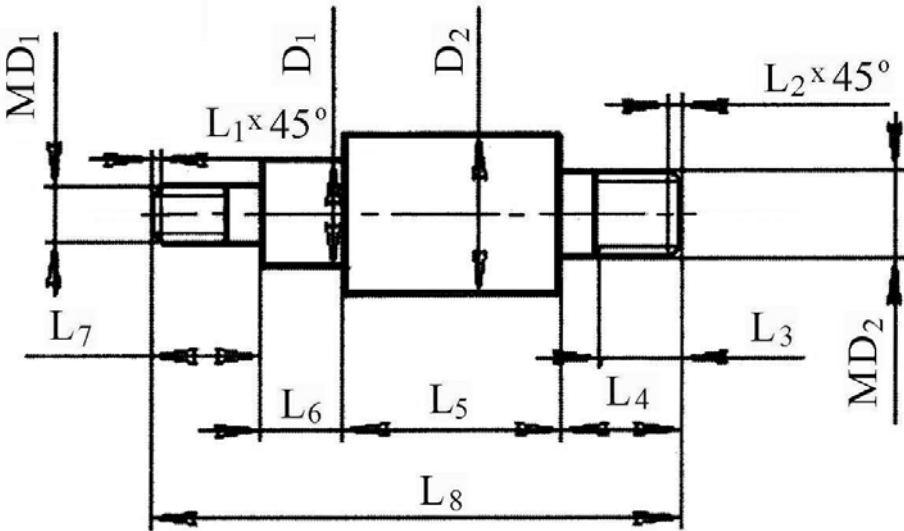
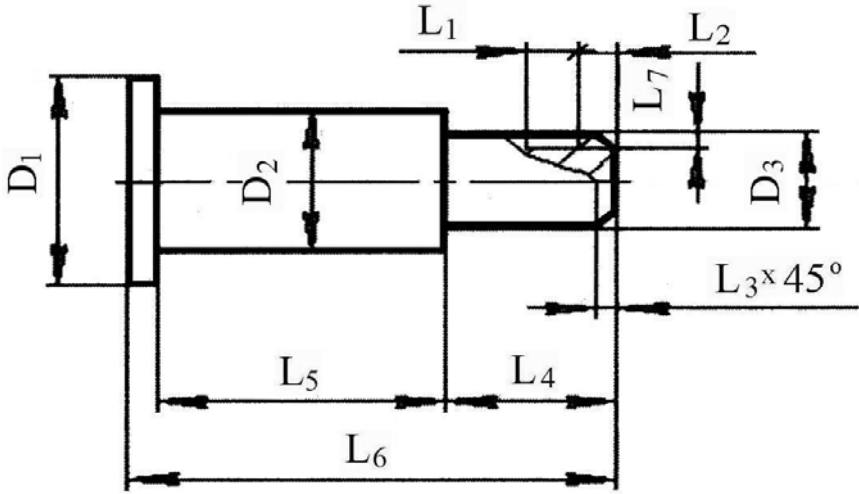
Составить программу отображения элементов технических систем в графическом режиме согласно выданному преподавателем варианту задания из таблицы 12.1.

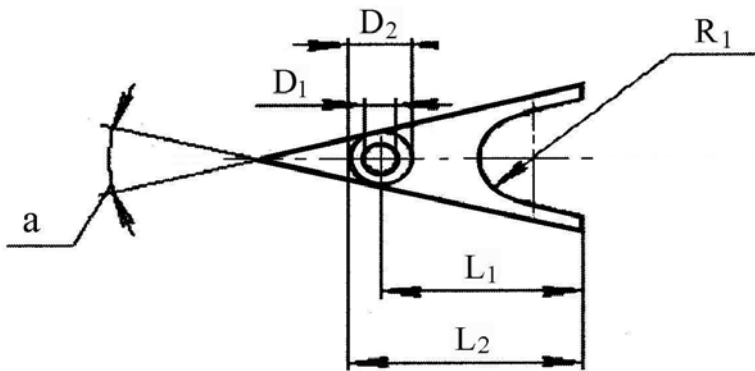
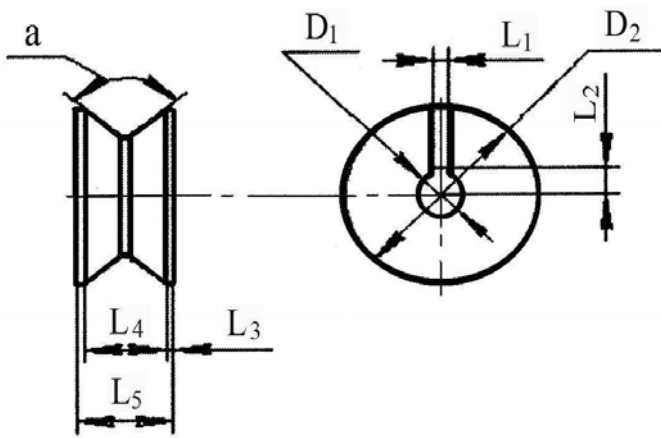
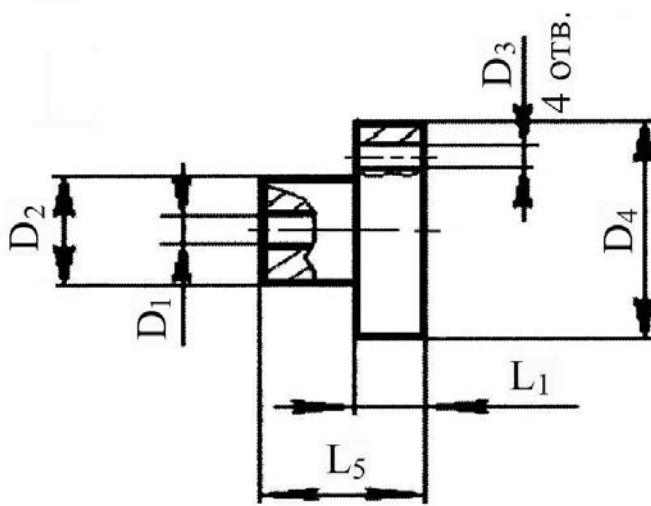
Задания к практическому занятию

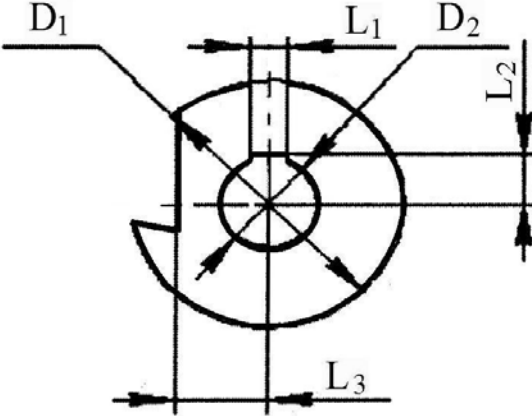
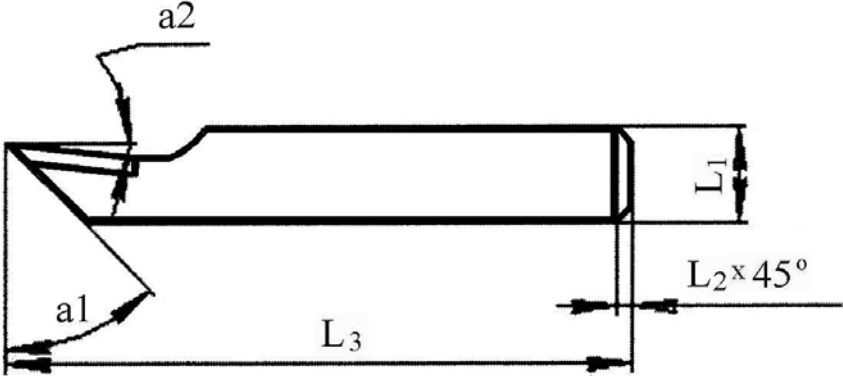
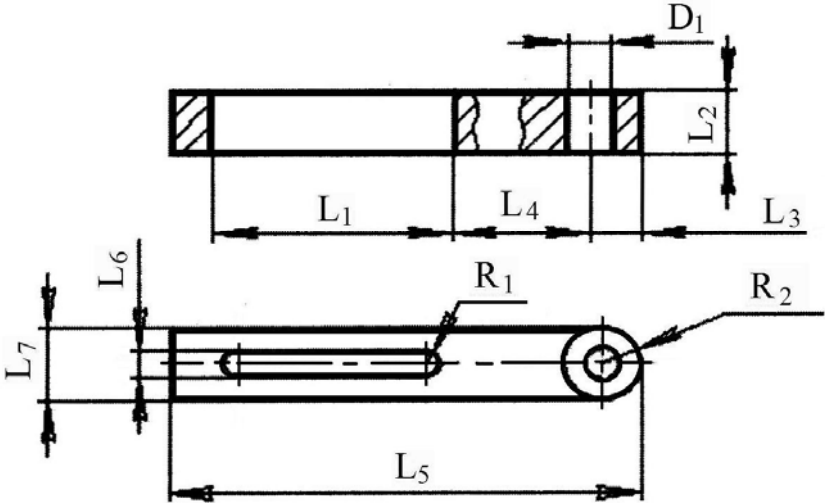
Вариант задания	Эскиз элемента технической системы
1	2
1	Корпус 1
2	Корпус 2

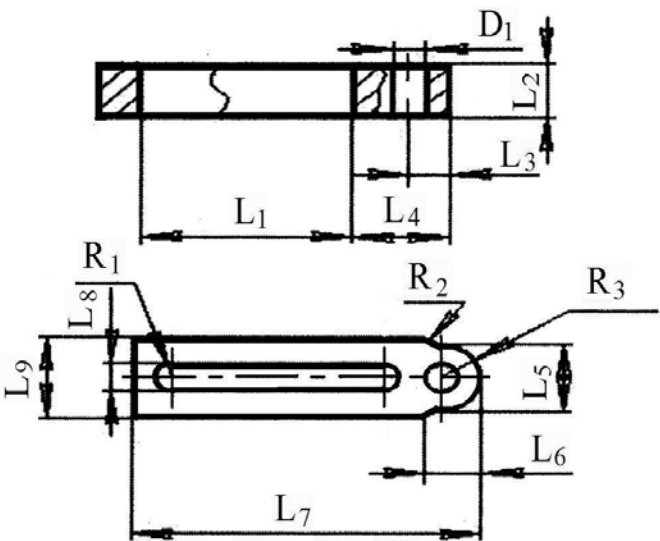
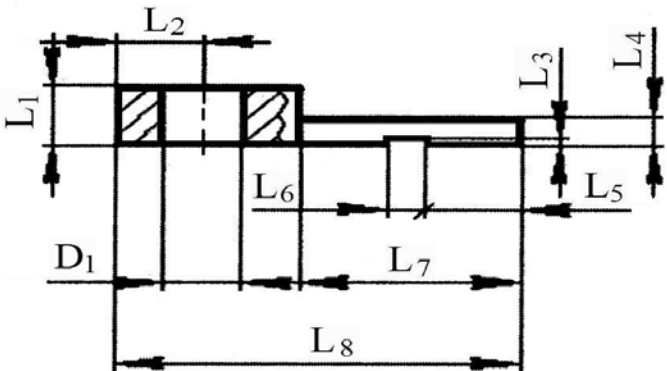
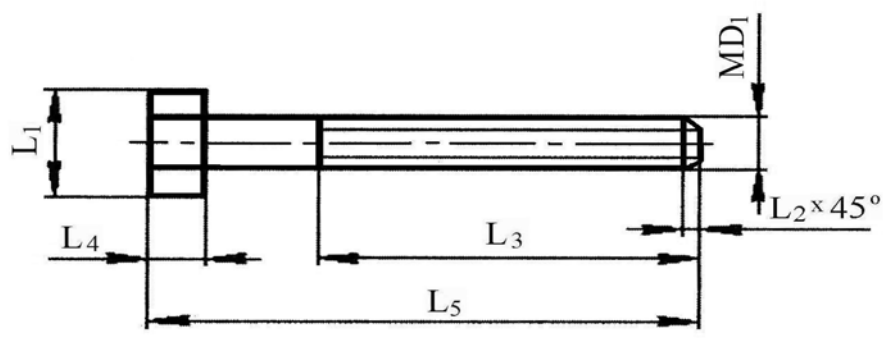
1	2
3	Корпус 3 Technical drawing of Корпус 3 (Body 3). The drawing shows a stepped cylindrical component with three circular features labeled D1, D2, and D3. Dimensions are indicated by arrows and labels: L1, L2, L3, L4, L5, L6, L7, L8, L9, L10, and L11. L1, L2, and L3 are vertical dimensions on the left. L4, L5, and L6 are horizontal dimensions at the top. L7 is a vertical dimension on the right. L8 is a vertical dimension at the bottom right. L9, L10, and L11 are horizontal dimensions at the bottom. The circular features D1, D2, and D3 are located at different heights and horizontal positions.
4	Корпус 4 Technical drawing of Корпус 4 (Body 4). The drawing shows a stepped cylindrical component with a circular feature labeled D1. Dimensions are indicated by arrows and labels: L1, L2, L3, L4, L5, L6, L7, L8, and L9. L1 is a vertical dimension on the left. L2 and L3 are horizontal dimensions at the top. L4 is a vertical dimension on the right. L5, L6, and L7 are vertical dimensions at the bottom right. L8 and L9 are horizontal dimensions at the bottom. The circular feature D1 is located on the left side. There is a hatched area on the left side of the component.

1	2
5	Корпус 5 
6	Кондукторная плита 

1	2
11	Вал 1  The diagram shows a shaft assembly with a central shaft of diameter D_2. On the left, there is a component with diameter D_1 and a shoulder of length $L_1 \times 45^\circ$. To the right of the central shaft is another component with diameter D_1 and a shoulder of length $L_2 \times 45^\circ$. The shaft has a total length L_8. The distance from the left end to the first shoulder is L_7. The distance between the two shoulders is L_5. The distance from the second shoulder to the right end is L_4. The distance from the left end to the start of the second shoulder is L_6. The distance from the right end to the start of the second shoulder is L_3. The diameters D_1 and D_2 are indicated at the ends of the shaft. The diameters MD_1 and MD_2 are indicated at the ends of the components.
12	Вал 2  The diagram shows a shaft assembly with a central shaft of diameter D_2. On the left, there is a component with diameter D_1. On the right, there is a component with diameter D_3 and a shoulder of length $L_3 \times 45^\circ$. The shaft has a total length L_6. The distance from the left end to the start of the right component is L_5. The distance from the right end to the start of the right component is L_4. The distance from the left end to the start of the right component is L_1. The distance from the right end to the start of the right component is L_2. The distance from the left end to the start of the right component is L_7. The diameters D_1, D_2, and D_3 are indicated at the ends of the shaft. The diameters L_1 and L_2 are indicated at the ends of the components.

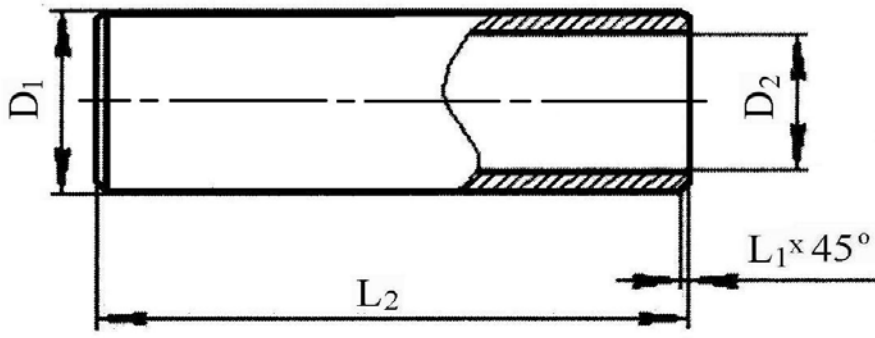
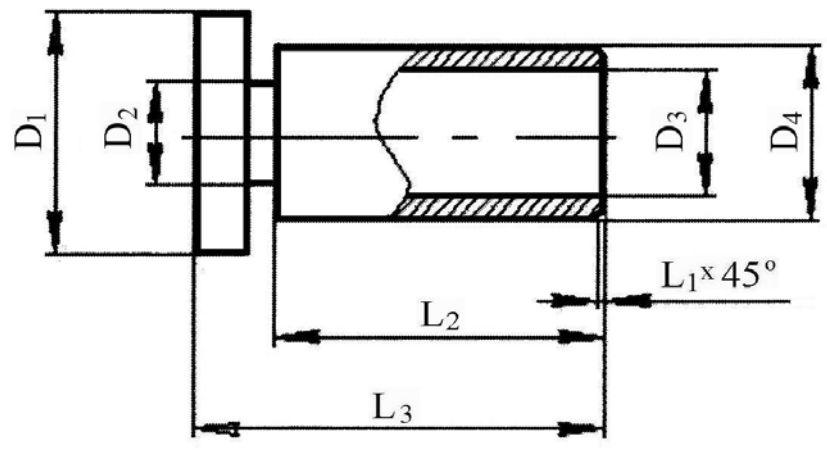
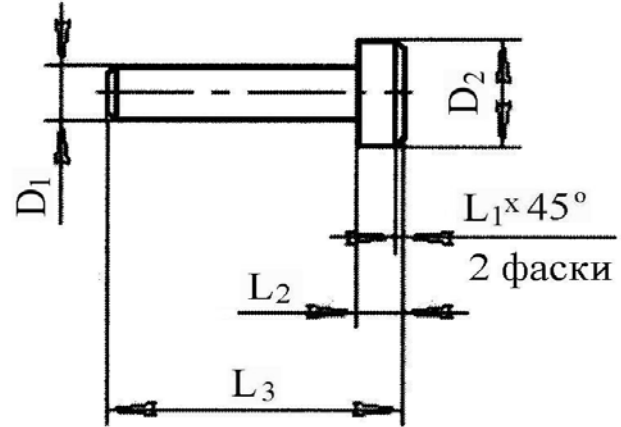
1	2
13	Вилка 
14	Шкив 
15	Фланец 

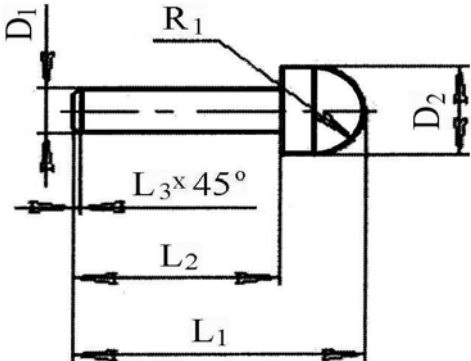
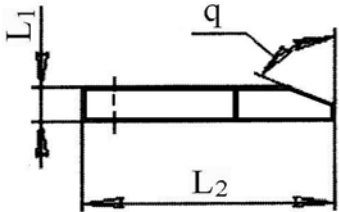
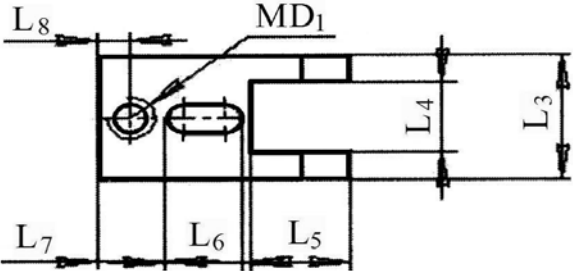
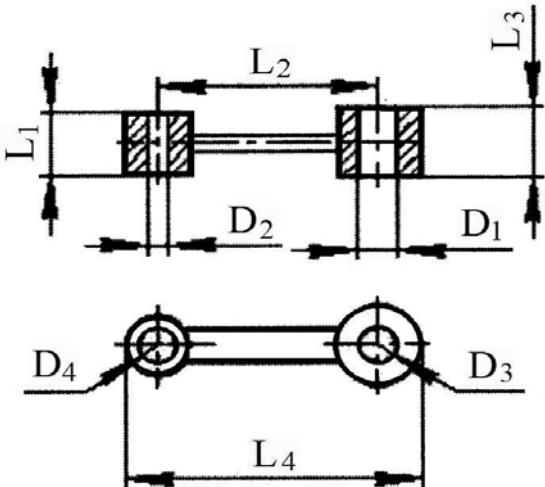
1	2
16	Фасонный резец 
17	Токарный резец 
18	Планка 1 

1	2
19	Планка 2 
20	Планка 3 
21	Болт 

Продолжение табл. 12.1

1	2
22	Винт Technical drawing of a screw (Винт) showing dimensions L_1, R_1, L_3, L_4, $L_2 \times 45^\circ$, and MD_1.
23	Шпонка 1 Technical drawing of a key (Шпонка 1) showing dimensions L_1, L_2, L_3, L_4, L_5, and MD_1.
24	Шпонка 2 Technical drawing of a key (Шпонка 2) showing dimensions L_1, L_2, L_3, L_4, and MD_1.

1	2
25	Втулка 1  Technical drawing of a sleeve (Втулка 1). The drawing shows a cylindrical part with an outer diameter D_1 and an inner diameter D_2. The length of the sleeve is L_2. A chamfer with a slope of $L_1 \times 45^\circ$ is indicated at the right end.
26	Втулка 2  Technical drawing of a sleeve (Втулка 2). The drawing shows a cylindrical part with an outer diameter D_1 and an inner diameter D_2. The length of the sleeve is L_2. The total length of the assembly is L_3. A chamfer with a slope of $L_1 \times 45^\circ$ is indicated at the right end. The drawing also shows dimensions D_3 and D_4 for the inner part.
27	Палец  Technical drawing of a pin (Палец). The drawing shows a cylindrical part with an outer diameter D_1 and an inner diameter D_2. The length of the pin is L_2. The total length of the assembly is L_3. A chamfer with a slope of $L_1 \times 45^\circ$ is indicated at the right end. The drawing also shows the text "2 фаски" (2 chamfers).

1	2
28	Опора 
29	Прихват  
30	Рычаг 

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

М

Множество пустое 73

О

Оператор

— выбора CASE 33

— присоединения WITH 64

П

Переменные цикла 123

Подготовка цикла 123

Подпрограмма 86

Процедура

— ARC 102

— BAR 104

— CIRCLE 102

— CLRSCR 94

— CLOSEGRAPH 100

— DELAY 95

— ELLIPSE 102

— FILLELLPSE 104

— LINE 101

— NOSOUND 96

— OUTTEXT 104

— OUTTEXTXY 104

— PUTPIXEL 101

— RECTANGLE 102

— SETBKCOLOR 103

— SETCOLOR 103

— SETFILLSTYLE 103

— SETLINESTYLE 101

— SETTEXTJUSTIFY 105

— SETTEXTSTYLE 105

— SOUND 95

Т

Тело цикла 47, 125

Тип

— запись 62

— массив 54

— множество 73

— файл 79

У

Условный оператор IF 13

Ф

Функция

— GETMAXX 101

— GETMAXY 101

— GRAPHRESULT 99

— GRAPHERRORMSG 100

— KEYPRESSED 93

— READKEY 93

Ц

Цикл

— вложенный 47

— внешний 47

— внутренний 47

— с параметром 38

— с постусловием 41

— с предусловием 40

ОСНОВНЫЕ ИСПОЛЬЗОВАННЫЕ ПОНЯТИЯ

Вложенный (сложный) цикл – цикл, телом которого является циклическая структура.

Внешний цикл – цикл, содержащий в себе другой цикл.

Внутренний цикл – цикл, содержащийся в теле другого цикла.

Оператор выбора CASE – оператор, позволяющий выбрать вариант выполнения в программе каких-либо операторов из любого количества вариантов.

Оператор присоединения WITH – оператор, который позволяет осуществлять доступ к полям записи так, как если бы они были простыми переменными.

Переменные цикла – переменные, изменяющиеся в цикле.

Подготовка цикла – задание начальных значений переменным цикла перед первым его выполнением.

Подпрограмма – это оформленная в виде самостоятельной программной единицы (модуля) повторяющаяся группа операторов, записанная в тексте программы однократно и вызываемая к выполнению лишь обращением к ней (ссылкой).

Процедура ARC – подпрограмма модуля GRAPH, которая вычерчивает дугу окружности по указанным координатам центра, начального и конечного угла, а также радиуса.

Процедура BAR – подпрограмма модуля GRAPH, которая заполняет прямоугольную область экрана с заданными координатами его левого верхнего и правого нижнего углов.

Процедура CIRCLE – подпрограмма модуля GRAPH, которая вычерчивает окружность по указанным координатам центра и радиуса.

Процедура CLRSCR – подпрограмма модуля CRT, очищающая экран или окно.

Процедура CLOSEGRAPH – подпрограмма модуля GRAPH, которая завершает работу адаптера в графическом режиме и восстанавливает текстовый режим работы экрана.

Процедура DELAY – подпрограмма модуля CRT, обеспечивающая задержку работы программы на заданный интервал времени.

Процедура ELLIPSE – подпрограмма модуля GRAPH, которая вычерчивает эллипсоидную дугу по указанным координатам центра, начального и конечного угла, горизонтального и вертикального радиуса.

Процедура FILLELLPSE – подпрограмма модуля GRAPH, которая обводит линией и заполняет эллипс.

Процедура FLOODFILL – подпрограмма модуля GRAPH, которая заполняет произвольную замкнутую фигуру узором или цветом.

Процедура LINE – подпрограмма модуля GRAPH, которая вычерчивает линию с указанными координатами начала и конца.

Процедура NOSOUND – подпрограмма модуля CRT, которая выключает динамик после активизации процедуры SOUND.

Процедура OUTTEXT – подпрограмма модуля GRAPH, которая выводит строку, начиная с текущего положения указателя.

Процедура OUTTEXTXY – подпрограмма модуля GRAPH, которая выводит строку, начиная с заданного листа координатами точки начала вывода строки.

Процедура PUTPIXEL – подпрограмма модуля GRAPH, которая выводит заданным цветом точку по указанным координатам.

Процедура RECTANGLE – подпрограмма модуля GRAPH, которая вычерчивает прямоугольник с указанными координатами углов (левого верхнего и правого нижнего).

Процедура SETBKCOLOR – подпрограмма модуля GRAPH, которая устанавливает цвет фона экрана.

Процедура SETCOLOR – подпрограмма модуля GRAPH, которая устанавливает текущий цвет для выводимых линий и символов.

Процедура SETFILLSTYLE – подпрограмма модуля GRAPH, которая устанавливает стиль (тип и цвет) заполнения.

Процедура SETLINESTYLE – подпрограмма модуля GRAPH, которая устанавливает стиль вычерчиваемых линий (сплошная, точечная, штрихпунктирная, пунктирная толщиной в один или три пикселя).

Процедура SETTEXTJUSTIFY – подпрограмма модуля GRAPH, которая задает выравнивание выводимого текста по отношению к текущему положению указателя или к заданным координатам.

Процедура SETTEXTSTYLE – подпрограмма модуля GRAPH, которая устанавливает стиль шрифта выводимой строки (тип, направление, размер).

Процедура SOUND – подпрограмма модуля CRT, которая определяет звуковой сигнал динамика компьютера с нужной частотой.

Пустое множество – множество, не содержащее ни одного элемента.

Тело цикла – операторы (действия), повторяющиеся в цикле для различных значений переменных цикла.

Тип–запись – структура данных, состоящая из фиксированного числа компонентов, называемых полями.

Тип–массив – упорядоченная совокупность компонентов одинакового типа, в которой каждому компоненту соответствует набор номеров (индексов), однозначно определяющих его местоположение в общей последовательности.

Тип–множество – совокупность неповторяющихся объектов порядкового типа, представляющих собой множество различных сочетаний элементов исходного множества.

Тип–файл – это последовательность компонент, расположенных на внешнем устройстве

Условный оператор IF – оператор, реализующий алгоритмическую конструкцию «развилка» и изменяющий порядок выполнения операторов в зависимости от истинности или ложности некоторого логического условия.

Функция GETMAXX – подпрограмма модуля GRAPH, которая автоматически определяет максимальную горизонтальную координату графического экрана.

Функция GETMAXY – подпрограмма модуля GRAPH, которая автоматически определяет максимальную вертикальную координату графического экрана.

Функция GRAPHRESULT – подпрограмма модуля GRAPH, которая возвращает код ошибки для последней графической операции и определяет правильность настройки графической операции и определяет правильность настройки графического режима.

Функция GRAPHERRORMSG – подпрограмма модуля GRAPH, которая выводит расшифровку ошибки, допущенной при настройке графического режима.

Функция KEYPRESSED – подпрограмма модуля CRT, которая возвращает TRUE (истина), если на клавиатуре была нажата клавиша, и FALSE (ложь) в противном случае.

Функция READKEY – подпрограмма модуля CRT, которая читает символ с клавиатуры без эхо – повтора на экране и приостанавливает исполнение программы до нажатия на любую клавишу.

Цикл – многократно повторяющийся участок вычислений, заложенный алгоритмом программы.

Цикл с параметром – цикл, в котором многократно выполняется один оператор с заранее известным числом повторений.

Цикл с постусловием – цикл, в котором многократно выполняется любое число операторов с заранее неизвестным числом повторений, при этом условие выхода из цикла вычисляется и анализируется после очередного его выполнения.

Цикл с предусловием – цикл, в котором многократно выполняется один оператор с заранее неизвестным числом повторений, при этом условие выхода из цикла вычисляется и анализируется перед каждым его выполнением.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Алексеев, Е. Р. Free Pascal и Lazarus : учебник по программированию / Е. Р. Алексеев, О. В. Чеснокова, Т.В. Кучер. – СПб. : Alt Linux ; ДМК – пресс, 2010. – 442 с.
2. Андреева, Т. А. Программирование на языке Pascal : учебное пособие / Т. А. Андреева. – М. : Интернет – Университет Информационных Технологий ; БИНОМ. Лаборатория знаний, 2006. – 234 с.
3. Робаков, Г. Г. Программирование на языке Pascal : учебное пособие / Г. Г. Робаков, С. Ю. Ржецкая. – СПб. : БХВ – Петербург, 2004. – 480 с.

РАБОТА В СРЕДЕ FREE PASCAL

Прежде чем приступить к выполнению практических занятий, необходимо ознакомиться с операционной системой FREE DOS и принципами программирования в среде FREE PASCAL [1–3].

А.1. 1. Общие положения

Интегрированная среда программирования FREE PASCAL включает в себя экранный редактор, компилятор, компоновщик и отладчик программ, а также систему контекстной информационной помощи.

Для входа в интегрированную среду необходимо найти в соответствующем каталоге командный файл, запускающий FREE PASCAL, выделить его и нажать на клавишу «Enter».

После загрузки этого файла на экране дисплея появится основной экран среды FREE PASCAL, состоящий из трех частей: строки основного меню (верхней строки экрана), поля экрана (средней части экрана) и строки состояния (нижней строки экрана).

А.1. 2. Основное меню

Основное меню содержит все возможные режимы работы FREE PASCAL. Для его активизации (выделения более ярким цветом) необходимо нажать функциональную клавишу F10. После этого при помощи клавиш управления курсором выбирают интересующее пользователя поле и после нажатия «Enter» заходят в подменю этого поля и т. д., пока не будет выбрана какая-либо команда или параметр.

Основное меню состоит из следующих полей:

File – Работа с файлами;

Edit – Редактирование части программы;

Search – Поиск фрагментов текста, подпрограмм или места ошибок;
Run – Выполнение программы;
Compile – Компиляция программы;
Debug – Отладка программы;
Tools – Инструментальные средства среды;
Options – Установка параметров среды;
Window – Работа с окнами;
Help – Помощь (справка).

Для выбора поля основного меню можно также воспользоваться мышью, подведя ее курсор к нужному полю и нажав левую клавишу, или клавишей «Alt» совместно с клавишей первой буквы имени поля (например, для выполнения программы можно нажать «Alt + R»).

Для выхода из основного меню в окно редактора (поле экрана) необходимо нажать клавишу «Esc».

Рассмотрим подробнее наиболее часто используемые поля и команды основного меню.

Меню работы с файлами (File) состоит из следующих полей:

New – Создать новый файл;
Open – Открыть существующий файл;
Save – Сохранить файл;
Save as – Сохранить файл под именем...;
Save all – Сохранить все измененные файлы;
Change dir – Изменить текущий каталог;
Print – Распечатать файл;
Printer setup – Установить параметры принтера;
Dos shell – Временный выход в операционную систему;
Exit – Выход из среды FREE PASCAL в операционную систему.

Выйти из интегрированной среды FREE PASCAL (т. е. закончить работу в среде) можно также, нажав комбинацию клавиш «Alt + X».

Меню редактирования (Edit) позволяет выполнить ряд операций с фрагментами текста программы:

Undo – Удалить изменения, внесенные в редактируемую строку;

Redo – Восстановить изменения, удаленные командой Undo;

Cut – Удалить выделенный блок текста и поместить его в промежуточный буфер;

Copy – Скопировать выделенный блок текста с размещением его в промежуточном буфере;

Paste – Поместить информацию из промежуточного буфера в текущий файл, начиная с позиции курсора;

Clear – Удалить выделенный блок текста без записи его в буфер;

Show clipboard – Открыть окно промежуточного буфера и сделать его активным.

Меню выполнения программы (RUN) позволяет запускать программу, а также выполнять отдельные ее части во время отладки:

Run – Запустить программу;

Step over – Выполнить очередные операторы программы, соответствующие одной строке текста, без захода в подпрограммы;

Trace into – То же, что и Step over, но с заходом в подпрограммы;

Goto cursor – Выполнить программу до строки, на которой находится курсор;

Program reset – Прекратить выполнение программы;

Params – Задать программе параметры.

Меню компиляции (COMPILE) содержит следующие основные поля:

Compile – Скомпилировать программу, преобразовав ее исходный текст в вид, доступный для ПЭВМ, т. е. в машинных кодах;

Make – Собрать программу из отдельных модулей;

Destination – Разместить файл программы в оперативной памяти (Memory), и она будет выполняться из среды FREE PASCAL или на диске (Disk), т. е. создать командный файл с расширением .exe, выполняемый из любого места каталога ПЭВМ;

Information – Получить информацию о файле.

А.1. 3. Экранный редактор

Встроенный экранный редактор предназначен для создания исходных текстов программ и их коррекции.

Сразу после запуска командного файла среда автоматически переходит в режим редактирования текста. Выйти из него в основное меню можно, нажав клавишу F10. Для возврата в режим редактирования необходимо нажать или «F10», или «Esc».

Текст программы набирается обычным образом, как при печати на пишущей машинке. Для перехода к новой строке (или открытия новой строки) следует нажать клавишу «Enter». Набираемый текст размещается в средней части (поле) экрана между строкой основного меню и строкой состояния. Компилятор FREE PASCAL воспринимает строки программы длиной не более 126 символов.

Для выполнения различных операций над текстом программы экранный редактор содержит ряд команд, условно разделяемых на несколько групп.

А.1. 3. 1. Команды перемещения курсора

←	– Переместить влево на символ;
→	– Переместить вправо на символ;
Ctrl + ←	– Переместить влево на одно слово;
Ctrl + →	– Переместить вправо на одно слово;
↑	– Переместить вверх на одну строку;
↓	– Переместить вниз на одну строку;
Page Up	– Переместить на страницу вверх;
Page Down	– Переместить на страницу вниз;
Home	– Переместить к началу строки;
End	– Переместить к концу строки;
Ctrl + Home	– Переместить на первую строку окна;

- Ctrl + End – Переместить на последнюю строку окна;
- Ctrl + Page Up – Переместить на начало текста;
- Ctrl + Page Down – Переместить на конец текста.

А.1. 3. 2. Команды вставки и удаления информации

Ins – Переключить режим замены и вставки: в режиме вставки добавляемая информация помещается в месте расположения курсора, при этом уже имеющийся текст раздвигается; в режиме замены добавляемая информация помещается в место расположения курсора, при этом она накладывается на уже имеющийся текст;

- Backspace – Удалить символ слева от курсора;
- Del – Удалить символ, на который указывает курсор;
- Ctrl + T – Удалить слово или его часть справа от курсора;
- Ctrl + QY – Удалить символы от курсора до конца строки;
- Ctrl+Y – Удалить строку, указанную курсором.

А1. 3. 3. Команды работы с блоками информации

Под блоком понимается выделенный фрагмент текста программы (символ, слово, строка, абзац и т. д.), который можно удалить, скопировать, переместить и т. п.

Выделить блок можно:

1. Пометив его начало комбинацией клавиш «Ctrl + KB» и конец – «Ctrl + KK»;
 2. Используя мышь и помещая ее курсор на начало блока. Далее нажимается ее левая клавиша, затем, не отпуская левой клавиши, курсор перемещается в конец блока, после чего левая клавиша отпускается;
 3. Используя комбинации клавиш «Shift» + «Стрелки».
- Ctrl + KT – Пометить блок размером в одно слово;
 - Ctrl + Ins – Скопировать помеченный блок в промежуточный буфер;
 - Shift + Del – Перенести помеченный блок в промежуточный буфер;

Shift + Ins – Перенести блок из промежуточного буфера в файл;
Ctrl + KC – Скопировать помеченный блок в место расположения курсора;
Ctrl + KV – Переместить помеченный блок в место расположения курсора;
Ctrl + Del или Ctrl + KY – Удалить помеченный блок;
Ctrl + KW – Записать блок на диск;
Ctrl + KR – Прочитать блок с диска;
Ctrl + KP – Распечатать блок на принтере.
Ctrl + KH – Отмена (восстановление) выделения.

А.1. 4. Строка состояния

Строка состояния, занимающая нижнюю строку экрана дисплея, содержит перечень клавиш и их комбинаций для быстрого управления, которые наиболее часто используются в данной ситуации, а также то действие, которое выполняется или может выполняться в данный момент в среде FREE PASCAL. Содержание строки меняется при изменении режима работы среды (переходе к другому полю, окну или другим действиям).

Отдельные имена клавиш быстрого управления в этой строке выделены ярким цветом, означающим, что данная клавиша (или комбинация клавиш) является активной и может выполнять свои функции в данный момент времени.

Если же имя клавиши изображено тусклым цветом, то данная клавиша или их комбинация не может в данный момент времени выполнять свои функции.

СОЗДАНИЕ ПРОГРАММ НА ЯЗЫКЕ FREE PASCAL

Программа, создаваемая на языке FREE PASCAL, может содержать следующие разделы:

PROGRAM	– Заголовок программы;
USES	– Раздел объявления модулей;
LABEL	– Раздел объявления меток;
CONST	– Раздел объявления констант;
TYPE	– Раздел объявления типов данных;
VAR	– Раздел объявления переменных;
PROCEDURE	– Раздел объявления процедур;
FUNCTION	– Раздел объявления функций;
BEGIN....END	– Тело программы (обязательная часть).

При написании программ необходимо выполнять следующие правила и рекомендации:

1. Заголовок программы необязателен и игнорируется компилятором.
2. Порядок размещения разделов произвольный, можно создавать несколько одинаковых разделов.
3. Разделы отделяются друг от друга точкой с запятой.
4. Имена программы, постоянных, переменных и других элементов должны, по возможности, отражать их суть.
5. Во избежание недоразумений не следует локальным и глобальным параметрам давать одинаковые имена.
6. В любом месте программы можно использовать лишь те элементы (метки, типы, константы, и т. д.), которые были определены ранее по тексту программы или являются предопределенными элементами языка.

7. Тело программы начинается словом BEGIN и заканчивается словом END с точкой, которая является признаком конца программы.

8. Операторы в теле программы необходимо отделять друг от друга точкой с запятой.

9. Если оператор закрывается операторной скобкой END, то между ними ставить точку с запятой необязательно.

10. Не следует в одной строке объединять несколько операторов, за исключением простейших, т. к. это затруднит локализацию ошибки во время отладки программы.

11. Последовательность операторов, выполняющих какое-либо законченное действие, целесообразно отделять от предыдущих и последующих операторов пустыми строками.

12. При написании программы желательно использовать систему отступов, когда операторы, вложенные в другие операторы или операторные скобки, пишутся на строке с отступом вправо по отношению к другим операторам (обычно отступ делается в две позиции).

13. Рекомендуется широко использовать написание комментариев, заключаемых в фигурные скобки.

14. Все лишние, с точки зрения языка FREE PASCAL, пробелы, знаки табуляции, пустые строки, комментарии компилятором игнорируются и никак не сказываются на исполняемом файле программы.

15. Для того чтобы после завершения работы программы выведенные на экран результаты расчетов не «закрывались» текстом исходной программы, необходимо в конце текста добавить оператор READ (или READLN):

```
.....;  
READLN  
END.
```

Аналогичного результата можно добиться, если включить модуль CRT и воспользоваться функцией READKEY, добавив ее в конец программы:


```
Program TM;  
USES CRT;  
.....  
READKEY  
END.
```

В этом случае результаты останутся на экране дисплея до тех пор, пока не будет нажата клавиша «Enter».

Если не использовать этот прием, то посмотреть результаты расчетов можно, только нажав комбинацию клавиш «Alt + F5».

Для лучшего восприятия программ необходимо придерживаться правил их стилизации.

Стиль программирования – набор приемов или методов программирования, которые используют программисты, чтобы получить эффективные, удобные для применения и легко читаемые программы.

Ниже приведены простейшие правила стилизации программ в виде перечня из ключевых слов этих правил и развернутых толкований правил:

1. «Удобочитаемость» («читабельность»);
2. «Комментируемость»;
3. «Корректность комментариев»;
4. «Наглядность комментариев»;
5. «Пропуск строк»;
6. «Пробелы»;
7. «Корректность имен переменных» («идентификаторов»);
8. «Аббревиатуры» («стандартизация сокращений»);
9. «Размещение операторов»;
10. «Упорядочивание списка идентификаторов»;
11. «Скобки»;
12. «Отступы».

Правило 1. «Удобочитаемость».

Программы должны составляться таким образом, чтобы их могли прочитать в первую очередь люди, а не машины. Людям это необходимо для корректировки, применения и модификации программы.

Программа – это:

- 1) документ для последующего использования;
- 2) учебный материал;
- 3) средство для дальнейшей разработки более совершенных программ.

Правило 2. «Комментируемость».

Делайте комментарии больше, чем это кажется необходимым.

Некомментируемая программа – это, вероятно, наихудшая ошибка, которую может сделать программист, а также свидетельство дилетантского подхода.

Программу с пояснительными комментариями значительно легче отлаживать, а также читать (понимать) другим программистам.

Включать комментарии следует по мере написания программы, а не после.

Цель комментариев – облегчить понимание программы.

Типы комментариев:

- вводные;
- оглавления;
- пояснительные.

Вводные комментарии должны начинать каждую программу, подпрограмму или процедуру для пояснения того, что они делают.

Минимальная информация в вводных комментариях:

1. Назначение программы;
2. Указания по вызову программы и ее использованию;
3. Список и назначение основных переменных или массивов;
4. Указания по вводу-выводу. Список всех файлов;
5. Список используемых подпрограмм;

6. Название примененных математических методов, а также ссылки на материалы-источники, где содержится их описание;
7. Сведения о времени выполнения программы;
8. Требуемый объем памяти;
9. Специальные указания оператору;
10. Сведения об авторе;
11. Дата написания программы.

Оглавление – это комментарий, который используют в больших программах и содержит название, размещение и функцию каждого программного модуля.

Пояснительные комментарии необходимы в тех частях программы, которые трудно понять без комментариев.

Пояснительные комментарии обеспечивают словесное описание логики программы и изменения данных.

Комментировать следует каждый логически выделенный блок программы.

Комментарии должны содержать некоторую дополнительную информацию, а не перефразировать программу.

Комментарии должны указывать цель действия или объяснять логику программы, а не пояснять синтаксис языка программирования.

Правило 3. «Корректность комментариев».

Неправильные комментарии хуже, чем их отсутствие.

Правило 4. «Наглядность комментариев».

Располагать комментарии следует таким образом, чтобы это не делало программу менее наглядной.

Правило 5. «Пропуск строк» – метод улучшения наглядности программ.

В программе должны быть выделены фрагменты, логически самостоятельные блоки.

Правило 6. «Пробелы» следует ставить везде, где это приводит к улучшению читаемости программы.

Пример невыполнимого условия: `FOR I:=1 TO 5 DO`

Пример правильного написания оператора (с пробелом):

`FOR I:=1 TO 5 DO`

Правило 7. «Корректность имен переменных» – это залог удобочитаемости программ.

Используйте имена с подходящей мнемоникой, т. е. в качестве имен переменных должны употребляться термины, используемые в данной области.

`ADD_GIN_TO_VERMOUTH_GIVING_MARTINI`

Правило 8. «Аббревиатуры» («стандартизация сокращений»).

1. Каждое значащее слово подлежит сокращениям.
2. В аббревиатуру всегда должны включаться начальные буквы слов.
3. Согласные важнее гласных.
4. Начало слова важнее его конца.
5. Аббревиатура должна включать в себя от 6 до 15 букв.

Исходный вариант:

`COST PLUS`

`RECORD`

Вариант с аббревиатурами:

`CST PLS`

`RCRD`

Правило 9. «Размещение операторов»: одного оператора в строке достаточно.

Правило 10. «Упорядочивание списка идентификаторов».

`BETA, COST, DOBT, KEP, Z: INTEGER;`

Правило 11. «Скобки» ставятся всегда в сомнительных случаях – это сделает программу более понятной и предотвратит ошибки.

Правило 12. «Отступы» используйте для выявления структуры данных.

Пример:

```
IF (A<B) THEN
    BEGIN
        A:= -A;
        B:= A *B;
    END
ELSE
    BEGIN
        A:= A*B;
        B:= -B;
    END;
```

После разработки текста программы можно приступить к ее набору и отладке на ПЭВМ.

Для этого нужно выполнить следующую совокупность действий.

Вначале необходимо найти в каталоге командный файл, выделить его и нажать клавишу «Enter».

После загрузки в ПЭВМ командного файла среды FREE PASCAL пользователь попадает в окно экранного редактора.

Для того чтобы начать ввод новой программы, необходимо, нажав клавишу F10, выйти из редактора в основное меню и открыть поле File. Аналогичный результат получается при нажатии клавиш «Alt + F». Войдя в меню поля File, нужно выбрать команду New (создание нового файла) и нажать «Enter».

После этого среда автоматически создает новый файл с именем nopame00.pas и переходит в окно экранного редактора.

Далее можно набирать текст программы, используя для этого команды редактора, рассмотренные в приложении А.1.3.

После набора программы необходимо визуально убедиться в правильности ее написания, а затем попытаться скомпилировать ее. Для этого нужно войти в поле Compile основного меню при помощи клавиши F10, мыши или нажатия комбинации клавиш «Alt + C». Затем следует выбрать команду Compile в этом поле и нажать «Enter». Аналогичный результат достигается при нажатии комбинации клавиш «Alt + F9».

Если компиляция прошла удачно и машина не обнаружила в программе синтаксических ошибок, то она сообщает об этом и предлагает нажать любую клавишу для возврата в экран редактора.

Если же компилятор находит ошибку, то он прекращает дальнейшую компиляцию программы, выводит в нижней части экрана диагностическое сообщение о причине ошибки и номере строки, в которой находится эта ошибка. После исправления ошибки необходимо снова провести компиляцию.

Когда все ошибки устранены, можно запустить программу на выполнение, войдя в поле Run основного меню при помощи клавиш F10, «Alt + R», мыши или нажав комбинацию «Ctrl + F9».

При совпадении результатов машинного счета с ручным можно считать разработку программы законченной и записать ее в файл путем нажатия клавиши F2, после чего выйти из интегрированной среды FREE PASCAL, нажав «Alt + X».

ОБЩИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

В.3.1. Цель выполнения занятий

Основной целью является приобретение студентами навыков работы за компьютером и решения задач программирования алгоритмов различных структур.

В.3.2. Содержание отчета по практическим занятиям

Отчет по каждому практическому занятию должен включать в себя титульный лист; цель занятия; постановку задачи; алгоритм решения; программу; ручной расчет и результат, полученный с помощью ПЭВМ; выводы. Причем программа и расчет выполняются в соответствии с вариантом, указанным преподавателем, и представляются в напечатанном на принтере виде. Отчет оформляется на листах белой бумаги формата А4 (допускается использование двойных листов из тетради) для того, чтобы на них можно было получить распечатку.

**В.3.3. Пример оформления титульного листа отчета
к практическому занятию**

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Государственное образовательное учреждение
высшего профессионального образования

Ульяновский государственный технический университет

Кафедра «Технология машиностроения»

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №
<НАЗВАНИЕ ЗАНЯТИЯ>

ВАРИАНТ №

Выполнил: <ФИО студента>

Группа: <номер группы>

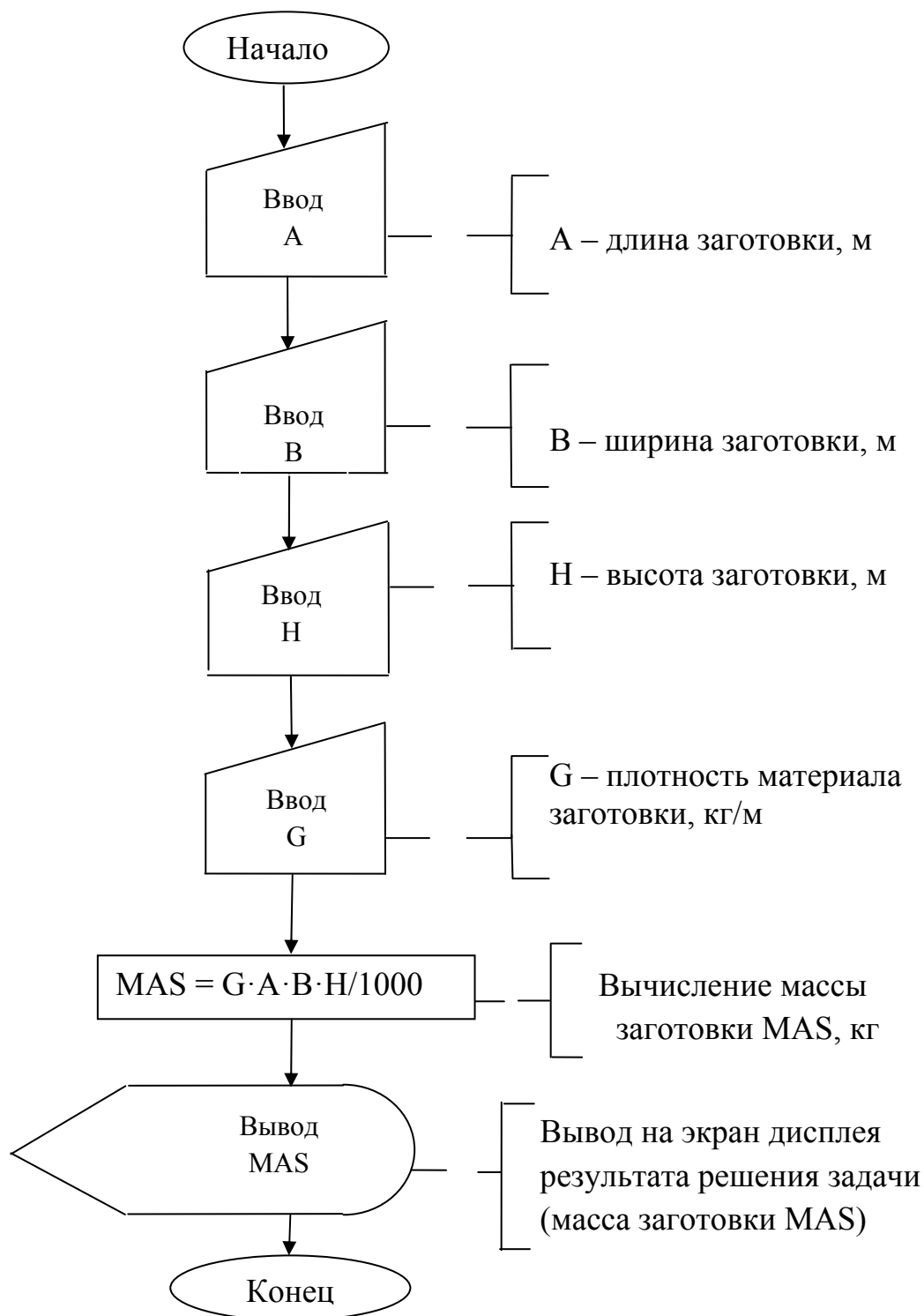
Проверил: <ФИО преподавателя>

Ульяновск

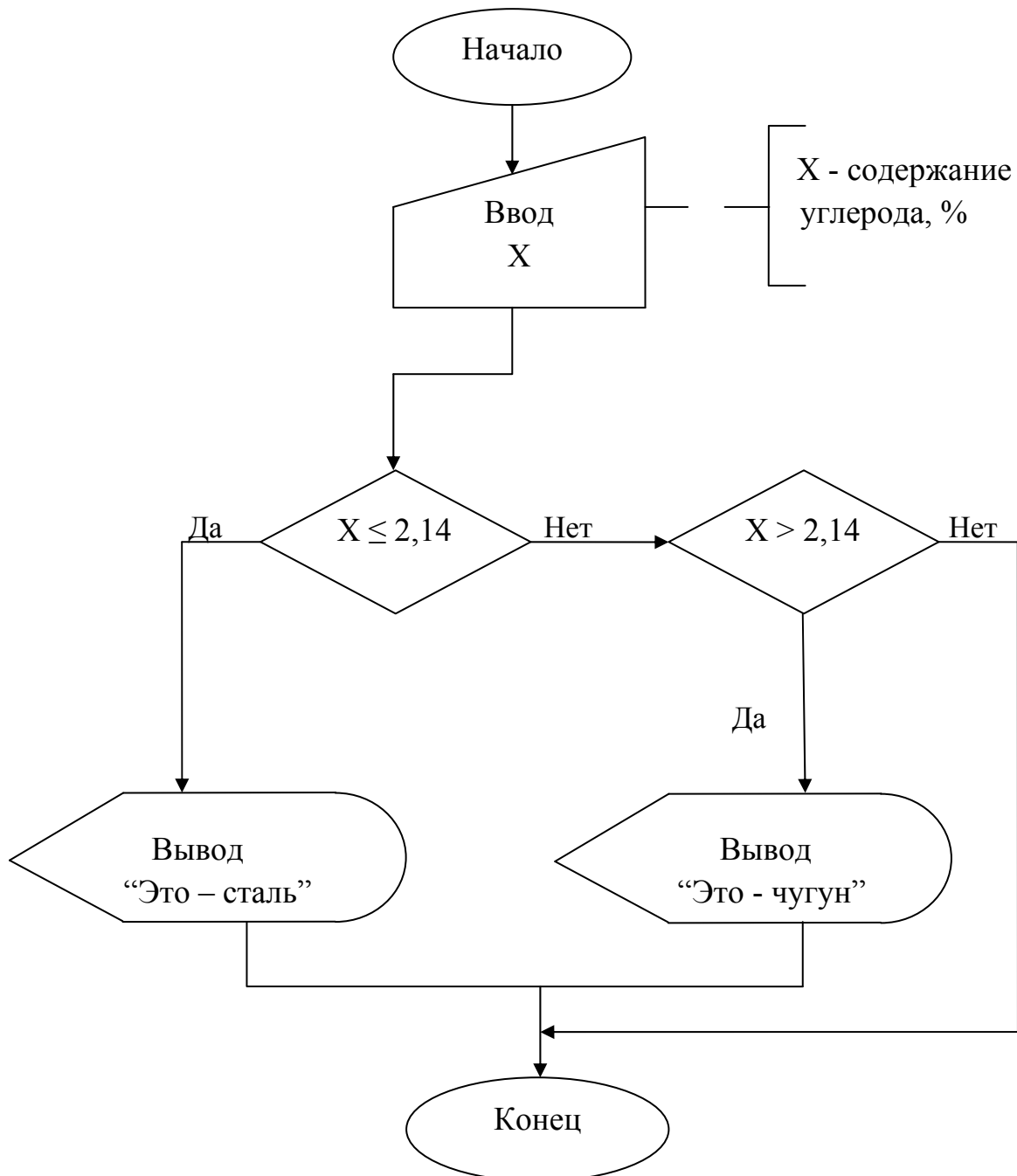
ПРИЛОЖЕНИЕ Г

ПРИМЕРЫ РАЗРАБОТКИ АЛГОРИТМОВ ПРОГРАММ

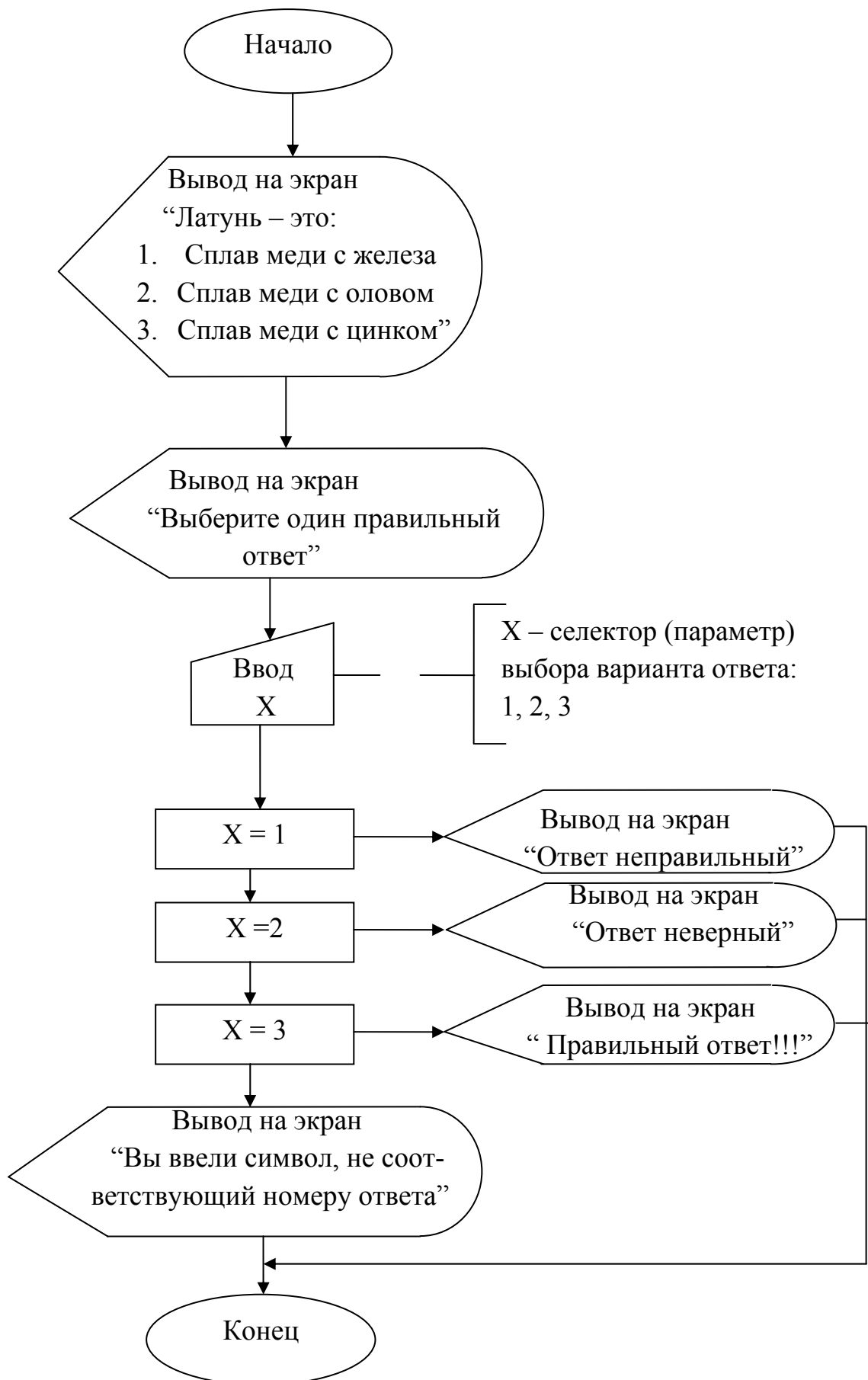
Г.1. Алгоритм задачи по расчету конструктивных параметров деталей машин (постановку задачи см. в примере 1.1)



**Г.2. Алгоритм задачи по расчету функциональных показателей
технических систем и процессов по условиям
(постановку задачи см. в примере 2.2)**



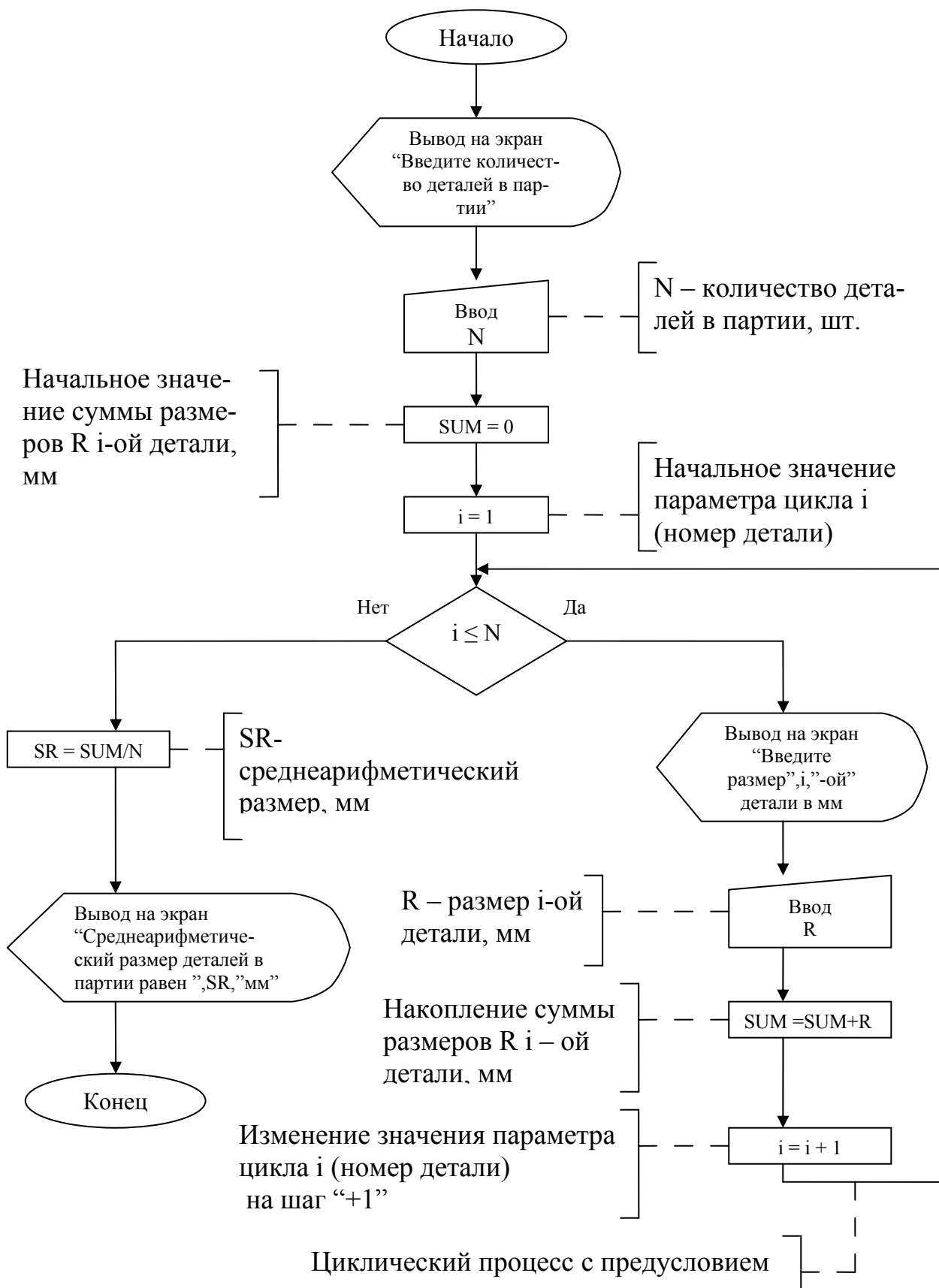
Г.3. Алгоритм задачи с инвариантным выбором
(постановку задачи см. в примере 3.1)



**Г.4. Алгоритм задачи по расчету параметров
технических систем при изменении одного фактора
(постановку задачи см. в примере 4.1)**



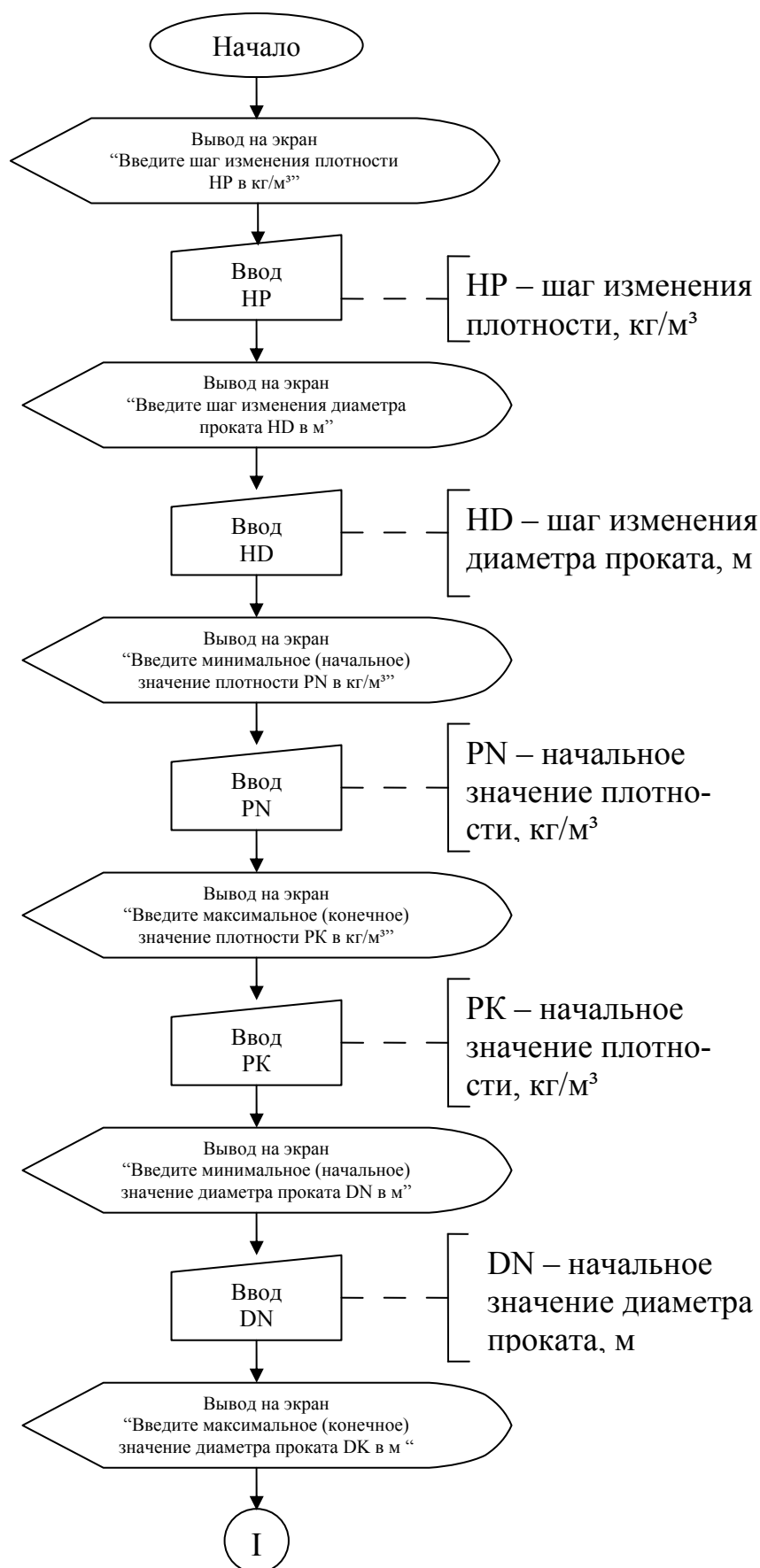
Г.5. Алгоритм задачи по расчету параметров технических систем при изменении одного фактора (постановку задачи см. в примере 4.2)

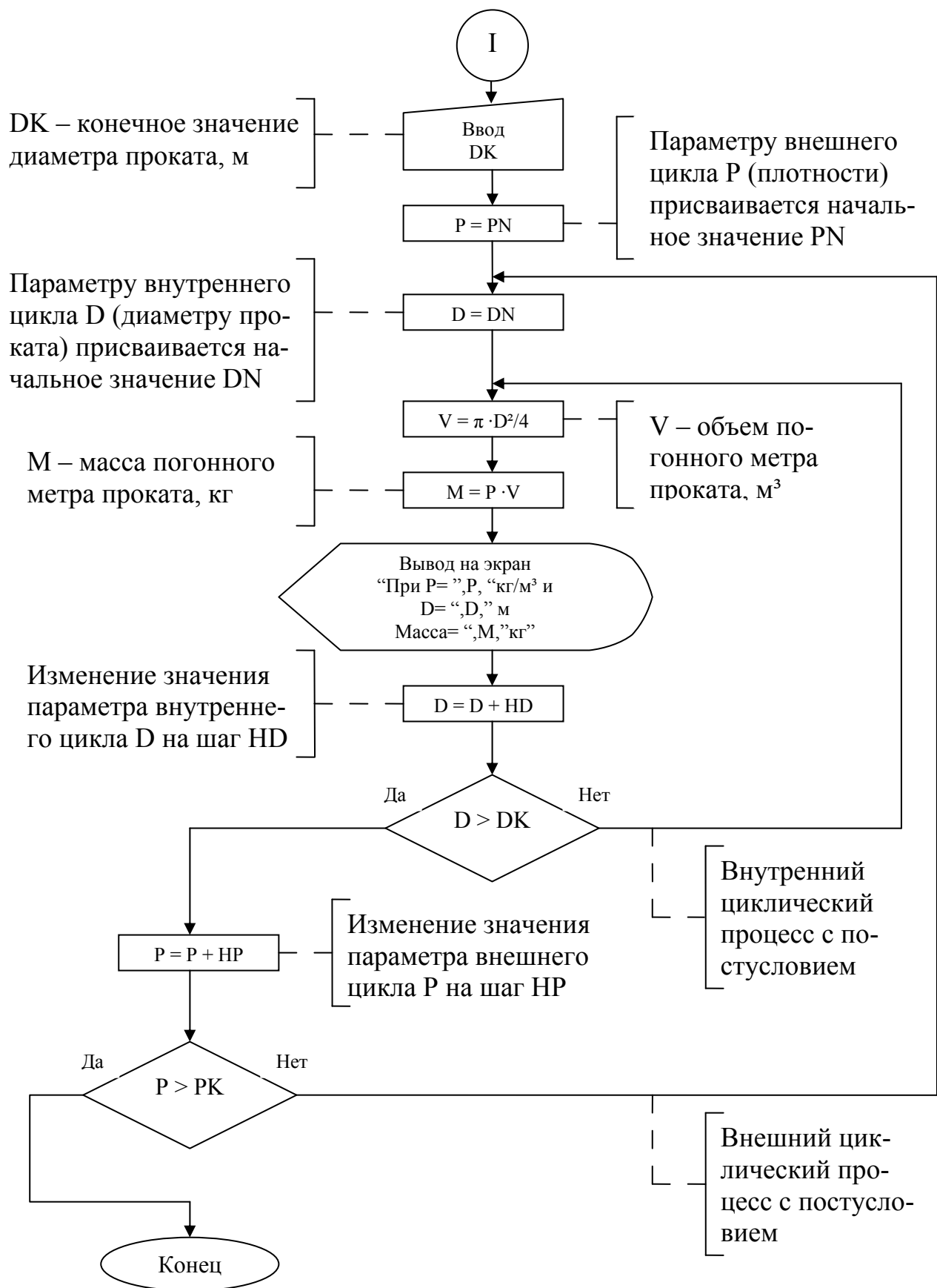


Г.6. Алгоритм задачи по расчету параметров технических систем при изменении одного фактора (постановку задачи см. в примере 4.3)



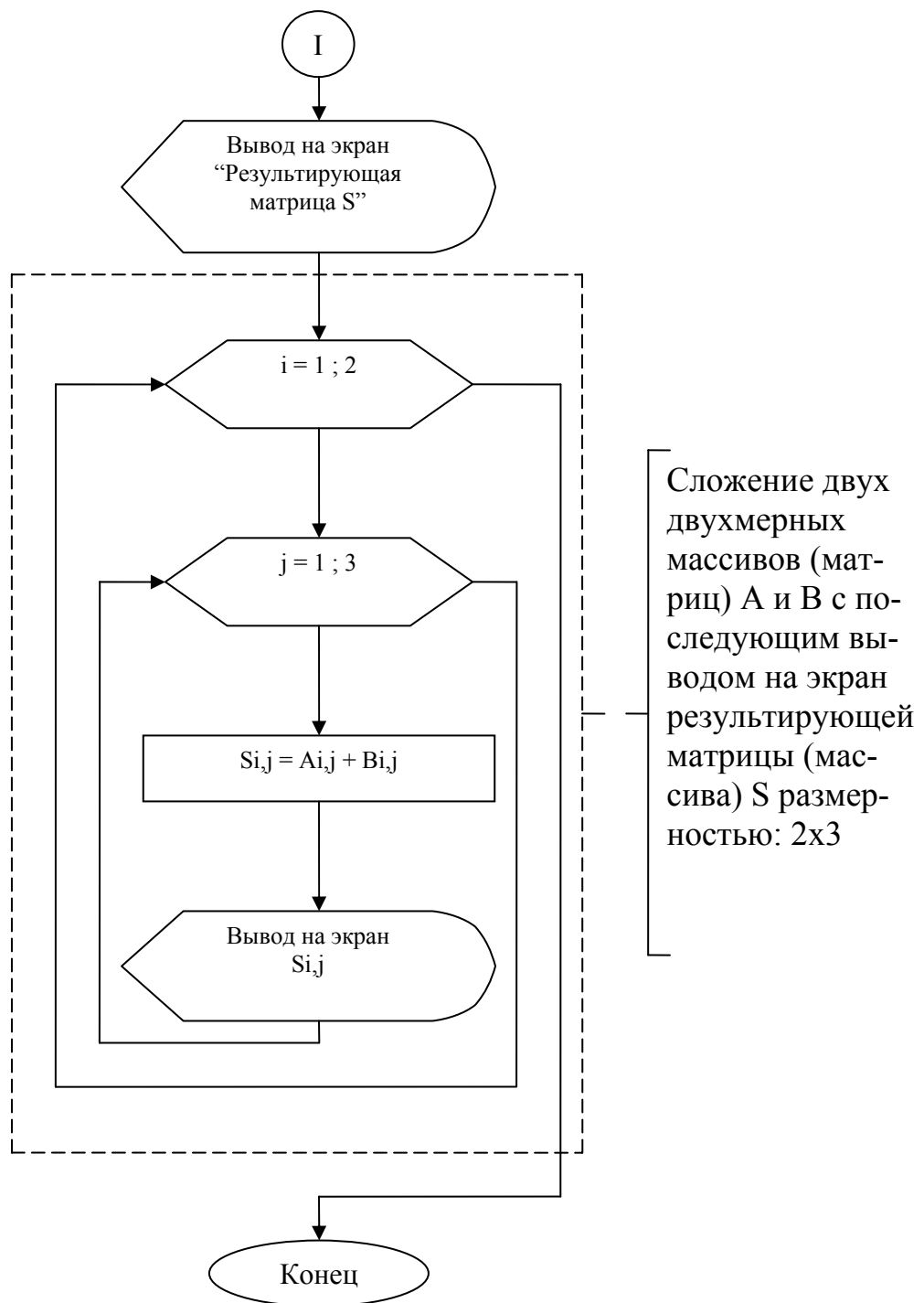
Г.7. Алгоритм задачи по расчету параметров при изменении нескольких факторов (постановку задачи см. в примере 5.1)



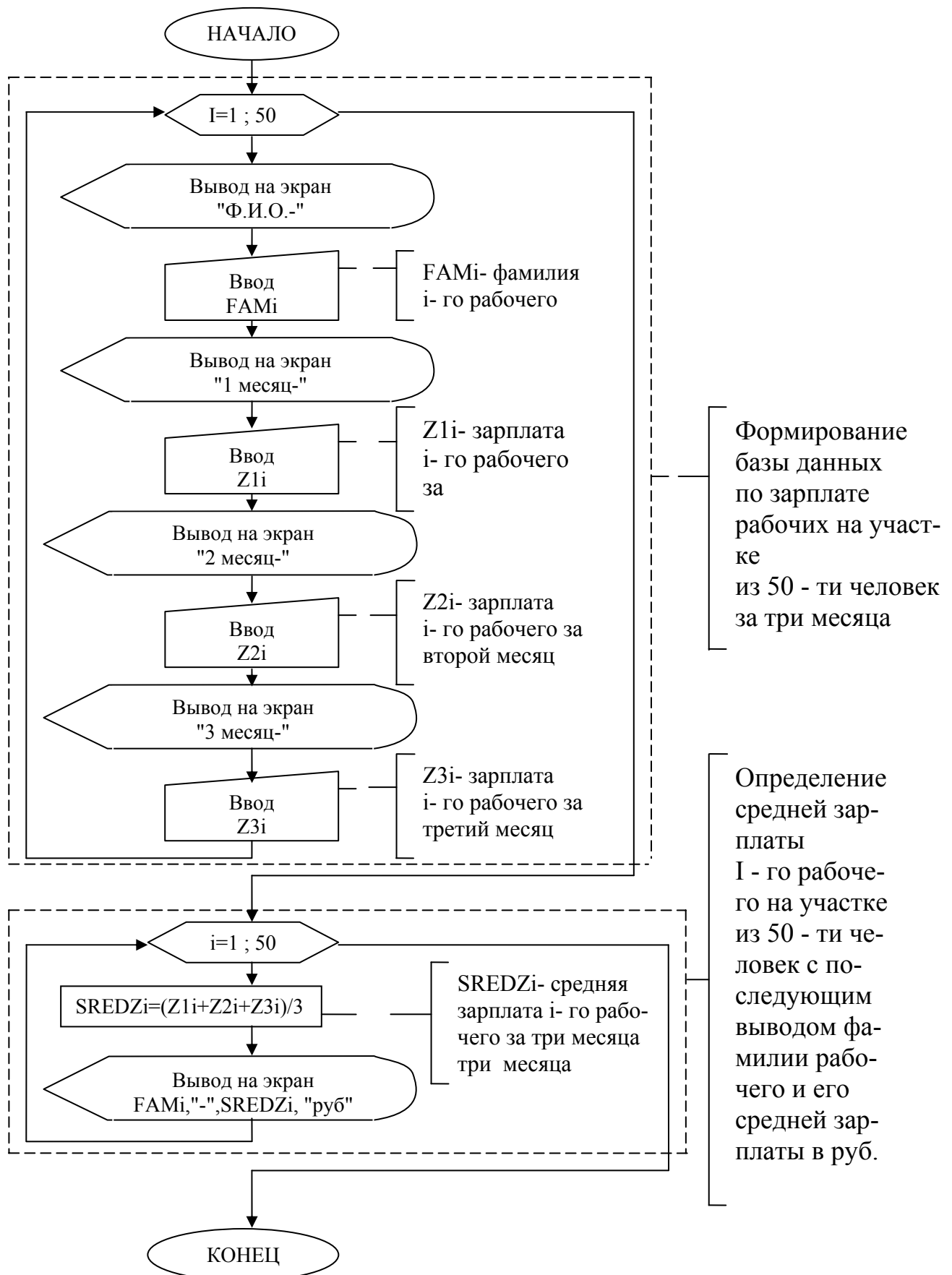


**Г.8. Алгоритм задачи обработки массивов экспериментальных данных при исследовании технических систем
(постановку задачи см. в примере 6.3)**

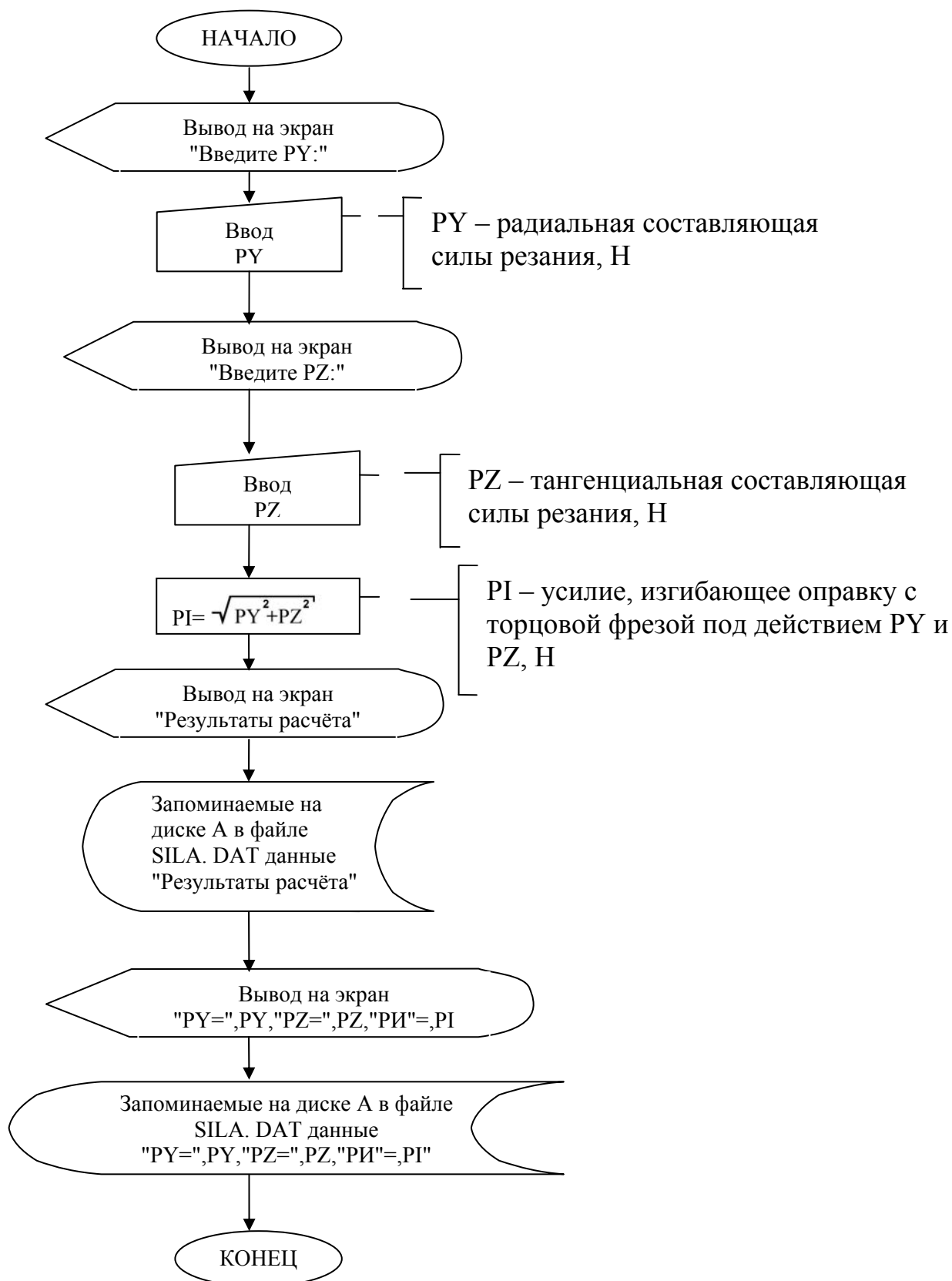




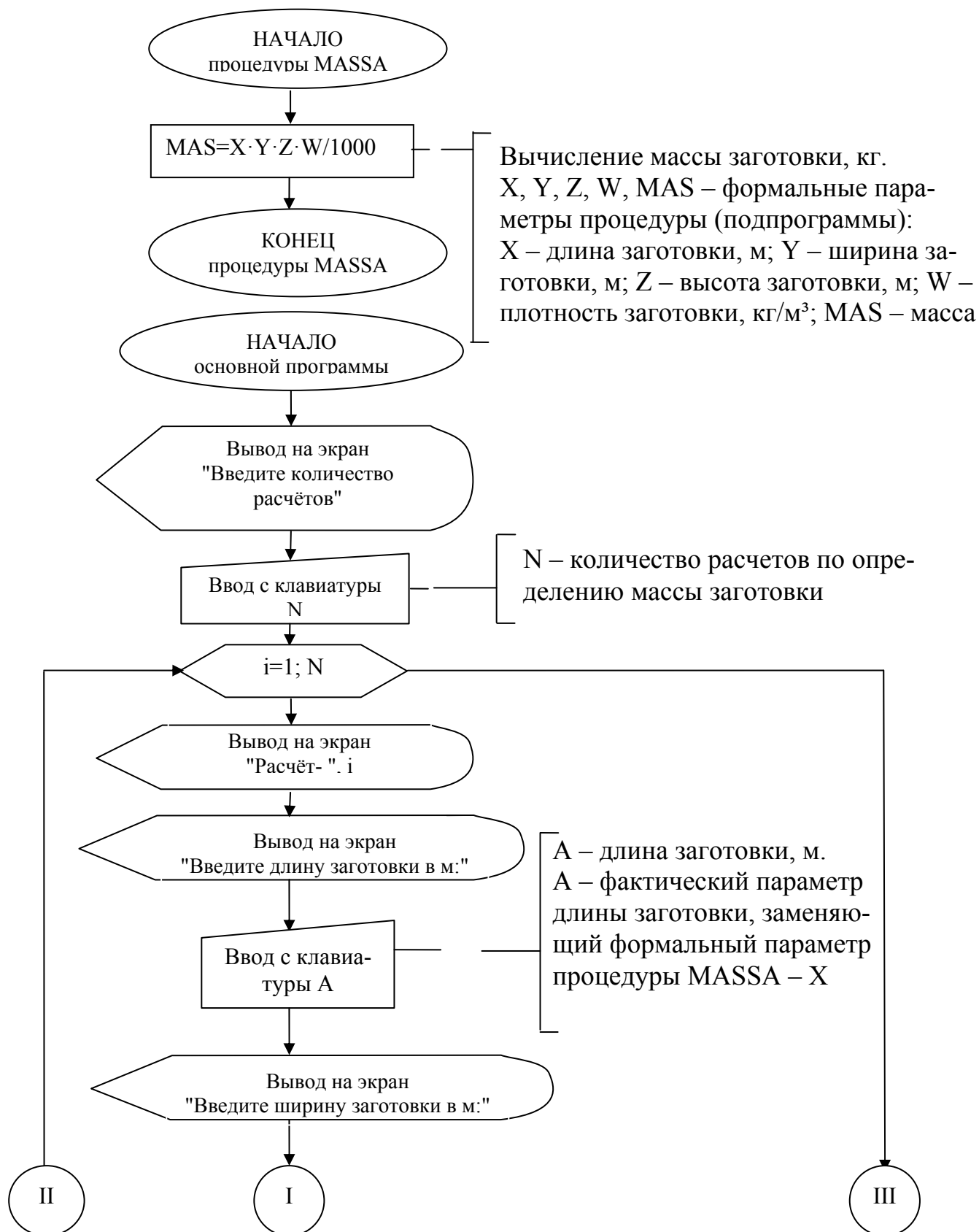
**Г.9. Алгоритм задачи по созданию и работе с базами данных
элементов технических систем (постановку задачи см. в примере 7.2)**

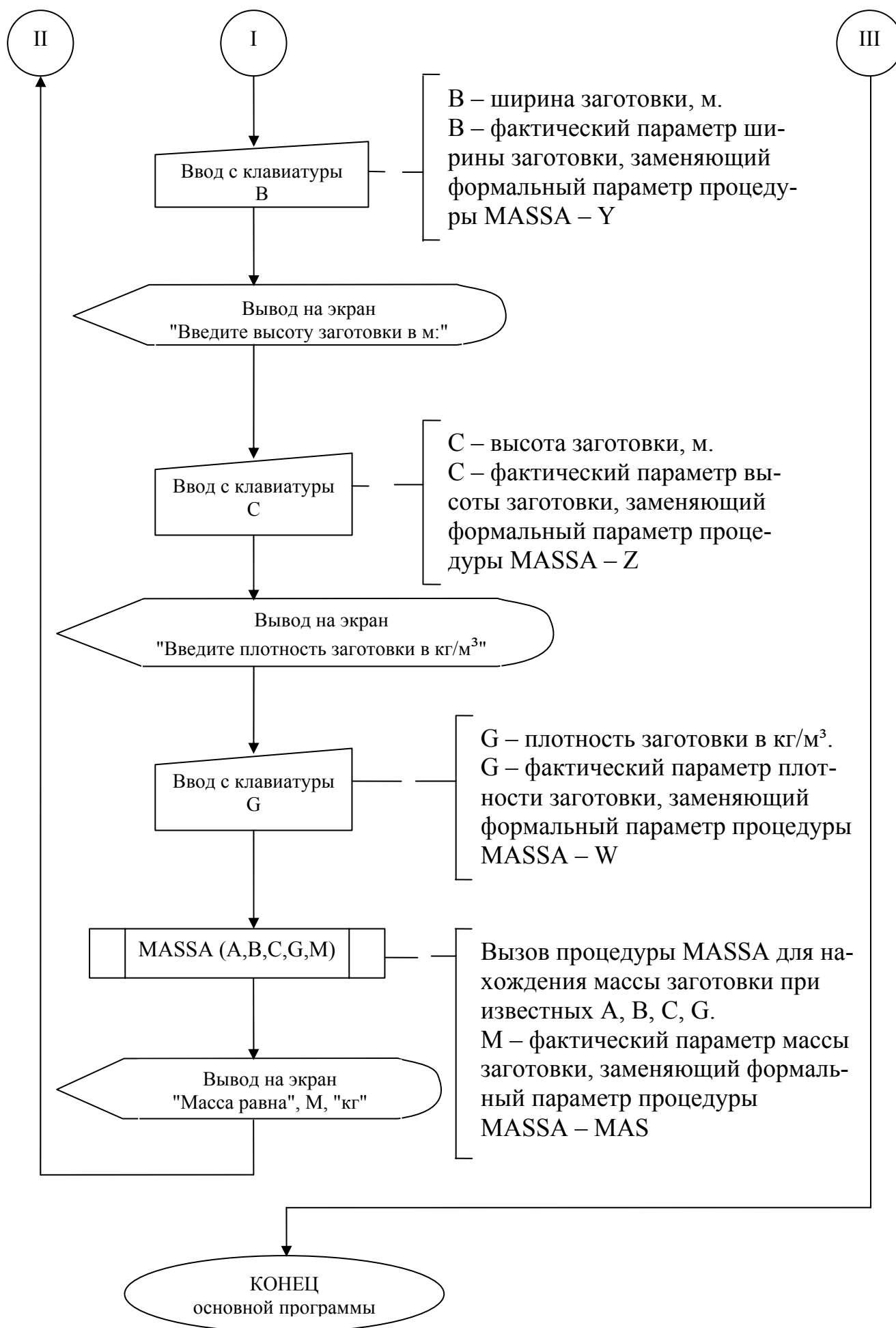


Г.10. Алгоритм задачи по сохранению результатов расчета параметров технических систем (постановку задачи см. в примере 9.2)



**Г.11. Алгоритм задачи по расчету параметров технических систем
с применением процедур и функций
(постановку задачи см. в примере 10.1)**





Учебное издание

КРУПЕННИКОВ Олег Геннадьевич,
РЯЗАНОВ Сергей Иванович,
ПСИГИН Юрий Витальевич,
КРАВЧЕНКО Дмитрий Валерьевич

**СБОРНИК ЗАДАНИЙ ДЛЯ ПРАКТИЧЕСКИХ ЗАНЯТИЙ
ПО ИНФОРМАТИКЕ**

Учебное пособие

Редактор М. В. Штаева

ЛР №020640 от 22.10.97.

Подписано в печать 29.10. 2010. Формат 60×84/16.

Усл. печ. л. 9,30. Тираж 300 экз. Заказ 1149.

Ульяновский государственный технический университет,
432027, г. Ульяновск, ул. Северный Венец, 32.

Типография УлГТУ, 432027, г. Ульяновск, ул. Северный Венец, 32.