

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНА МЕТАЛУРГІЙНА АКАДЕМІЯ УКРАЇНИ

**Швачич Г.Г, Пасинков В. М., Гуляєва О.А, Єфанова Л.М.,
Скороход Б.М., Романова Н.С., Кузьменко В. В.**

АЛГОРИТМІЧНА МОВА QBASIC

ЧАСТИНА 2

**Затверджено на засіданні Вченої ради академії
як навчальний посібник**

Дніпропетровськ НМетАУ 2004

УДК 004.075.8

Швачич Г.Г, Пасинков В.М., Гуляєва О.А, та ін. Алгоритмічна мова Qbasic.
Частина 2: Навч. посібник.– Дніпропетровськ: НМетАУ, 2004. – 36с..

Висвітлює особливості редагування та відладку підпрограм, створення та використання багатомодульних програм бібліотек, файлів різноманітного доступу. Викладені графічні можливості мови Qbasic, приведені приклади написання програм.

Призначений для студентів усіх спеціальностей
Друкується за авторською редакцією.

Відповідальний за випуск Швачич Г. Г. к. т. н. проф.

Рецензенти: Т.А.Чупілко, канд.техн.наук, доц. (Державний фінансово-економічний інститут)

Д.Г.Зеленцов, канд.техн.наук, доц.(УДХТУ)

© Національна металургійна
академія України, 2004

ВСТУП

Даний навчальний посібник містить приклади побудови програм рішення найбільш поширених задач в області математичного аналізу і лінійної алгебри, а також виводу графічної інформації.

Викладений матеріал по програмуванню дозволяє сформувати у навчаючихся логічно-математичне мислення, системний, операційний підхід до рішення задач різноманітного характеру і ступеня складності. Слід зазначити перспективність уміння програмувати на мові Quick Basic, тому що його конструкції використовуються у мові Visual Basic.

Зміст приведених розділів розкриває особливості створення багатомодульних програм і бібліотек користувача, файлів даних різноманітного доступу, графічних зображень.

Приведені завдання розраховані на індивідуальне виконання студентами групи при участі викладача, як консультанта.

Теоретичні викладки на початку кожного розділу нагадують про використанні в ньому засоби мови.

Після кожного розділу студент складає звіт, що містить формулювання завдання, програму і результат її виконання.

1. НАЛАГОДЖЕННЯ ПРОГРАМ З ВИКОРИСТАННЯМ МЕНЮ DEBUG

Пункт меню DEBUG використовується для того, щоб полегшити програмі коректне виконання шляхом відкриття Watch Window, яке показує як змінюються значення змінних на протязі виконання програми, а також встановленням breakpoints, які переривають виконання програми. Можна використати Shift/F9 для того, щоб включити або виключити breakpoints.

ADD WATCH... - відображення в оглядовому вікні значень змінних і виражень, введених у відповідь на запит у вікні.

INSTANT WATCH... (Shift+F9) - негайне відображення значення змінної, відзначеної курсором, або виразу, виділеного за допомогою Shift + клавіши пересування курсора (тільки у режимі переривання).

WATCH POINT... - останов виконання програми по заданій умові.

DELETE WATCH... - видалення завдань з оглядового вікна.

TOGGLE BREAKPOINT (F9) - установка відзначеної курсором точки переривання у виконуваний програмі. Скасування – натиснути **F9** ще раз.

CLEAR ALL BREAK POINT - видалення всіх точок переривання.

SET NEXT STATEMENT - перехід до оператора, раніше відзначеного курсором.

DELETE ALL WATCH... - видалення оглядового вікна.

Вказівки до вивчення розділу

1. Ввести і виконати програму:

```
INPUT A,B,H,D
10 S=(A+B)/2*H-3.14*D^2/4
PRINT S
END
```

при A=3.5; B=2.3; H=2; D=1.8

2. Задати відображення в оглядовому вікні змінної D і вираження $D^2/4$.

3. Виконати програму для вхідних даних, спостерігаючи в оглядовому вікні значення.

4. Установити точку переривання у рядку PRINT S. Виконати програму. Подивитися значення (A+B) у режимі негайного відображення.

5. Внести зміни в програму з метою організації циклічних обчислень, тобто після оператора PRINT S додати рядки

```
H=H+2
```

```
IF H<12 THEN 10
```

Виконати програму.

6. Встановити умову переривання $H>8$, виконати програму, порівняти отримані значення з попередніми.

7. Видалити умову $H>8$, видалити оглядове вікно.

8. Очистити пам'ять, ввести і виконати програму для

```
A=(1; 3; 5; 7;0 )
```

```
FOR i=1 TO 5
```

```
INPUT A(i)
```

```
NEXT
```

```
S=0
FOR i=1 TO 5
S=S+A(i)
NEXT
PRINT S
PRINT "END"
END
```

9. Встановити точку переривання на PRINT "END", виконати програму, подивитись значення S.

10. Внести зміни в програму:

оператор $S=S+A(i)^2$ замість оператора $S=S+A(i)$

11. З метою використання вже введеного масиву запустити програму з оператора $S=0$. Для цього:

- встановити курсор на цей оператор;
- задати перехід до нього за допомогою відповідного пункту меню;
- натиснути **F5**.

Перевірити вірність отриманих результатів.

2.СТВОРЕННЯ І ВИКОРИСТАННЯ БАГАТОМОДУЛЬНОЇ ПРОГРАМИ

При створенні одномодульної програми, та створенні підпрограм, часто виникають ситуації, коли, наприклад, в відведений час треба надрукувати, відлагодити та зберегти ПП

Створений модуль, окрім тексту ПП, має містити виклик ПП. Автоматично QB доповнює модуль оператором **DECLARE**.

Після відладки треба зберегти модуль під будь-яким ім'ям. В подальшому можливо використовувати цей файл для створення багатомодульної програми.

Якщо програма містить одну ПП, необхідно:

- відокремити цю ПП, щоб зберегти її в окремому файлі;
- зберегти всю програму на диску;

- повернутися до своєї програми;
- відредагувати робочу програму так, щоб залишився тільки текст ПП, який потім буде збережений як модуль.

При використанні в програмі. модуля треба бути впевненим в тому, що не буде зруйнована пам'ять завантаженням невикористованих модулів.

Якщо такий модуль існує, необхідно вибрати **Unload File** команду з **File** меню.

Unload команда висвітлить лист імен модулів робочої програми. Користуючись стрілками, треба висвітлити непотрібний модуль і натиснути Enter. Потім за допомогою Save All треба модернізувати .МАК файл. Команда Unload File знищує модуль в пам'яті, але він залишається на диску.

В оточенні QBASIC існують спеціальні команди редагування для створення та модифікації програм з багатьма модулями .

OPEN PROGRAM - очищення пам'яті і завантаження програмного файла з диска. Пункт FILE NAME визначає ввід імені файла;

CREATE FILE - підготування вводу слідуючого файла в багатомодульну програму;

SAVE ALL - запис на диск програмного файла-модуля багатомодульної програми;

При цьому створюється спеціальний файл із расширенням .МАК, що містить імена всіх складових модулів;

SET MAIN MODULE - призначення головного модуля в багатомодульній програмі (виконуваного при натисканні **F5** або **RUN**);

UNLOAD FILE - видалення непотрібного модуля з багатомодульної програми.;

SPLIT - поділ екрана на 2 вікна. Розміру вікон змінюють за допомогою **ALT/+** або **ALT/-**.

Натисканням клавіші **F2** здійснюється перегляд списку імен модулів.

Вказівки до вивчення розділу

1. Ввести і записати по **SAVE ALL** на диск E: програму:

```
A=2
B=3
C=A*B
PRINT C                (модуль 1 , ім'я PRI)
D=A-B
PRINT D
END
```

2. Використовуючи **CREATE FILE** і **SAVE ALL** записати на диск C: програми:

```
R=6
S1=3.14*R^2           (модуль 2 , ім'я PR2)
PRINT S1
END
M=3.5
L=8.6                 (модуль 3, ім'я PR3)
H=12.8
S2=(M+L)/2*H
PRINT S2
END
```

3. Перевірити список імен.
4. По черзі виконати кожний модуль, попередньо призначаючи його головним.
5. Розділити екран на вікна, відобразити в них два різних модулі.
6. Вийти з QB. Знайти на диску E: файл із розширенням .mak, подивитися його зміст.
7. Ввійти у QB і завантажити створену багатомодульну програму.
8. Видалити модуль 2 із багатомодульній програми.

3.ВИКОРИСТАННЯ ВНУТРІШНІХ НЕСТАНДАРТНИХ ФУНКЦІЙ

В алгоритмічній мові Qbasic використовується оператор DEF, який задає необхідну функцію з ім'ям та параметром. Іншими словами цей оператор дозволяє задавати за допомогою імені та формальних параметрів функцію для багатострокового використання для різних значень фактичних параметрів.

Оператор-функцію називають ще нестандартною функцією або функцією користувача.

Звернення до оператору здійснюється за ім'ям, як до стандартних функцій. Кожне ім'я функції може зустрічатися лише в одному операторі DEF.

Оператор DEF розташовується спочатку програмної одиниці перед виконуваними операторами і має вид:

DEF FNA(B) = 2*SIN(B + 1).

Оператор опису однострокової внутрішньої функції:

DEF FNім'я (список формальних параметрів)= вираз.

Оператор опису багатострокової внутрішньої функції:

DEF FNім'я (список формальних параметрів)

оператори

ім'я=вираз

оператори

END DEF

Звертання до функції:

Fним'я (список фактичних параметрів)

Вказівки до вивчення розділу

1. Скласти програму для обчислення:

$$A = \frac{\sqrt{2,5^2 + \cos^2 0,8 + \sin^2 1,8}}{\sqrt{R^2 + n^2 + M^2}},$$

у якій використати опис функції

DEF FNS(x,y,z)=SQR(x^2+y^2+z^2) і оператор

INPUT R,N,M

Вхідні дані: R=1.5; N=4.8; M=0.8

2. Ввести і виконати програму:

CLS

DEF FNS (x, y, z) = SQR(x ^ 2 + y ^ 2 + z ^ 2)

INPUT R, N, M

A = FNS(2.5, 1.8, 0.8) / FNS(R, N, M)

PRINT A

END

3. Скласти програму для обчислення функції:

$$Z = \frac{\log_2 x + \log_b y}{2 * \log_{b+2} (x+y)},$$

у якій використати опис функції для обчислення логарифма по будь-якій основі.

Вхідні дані: x=4; y=2; b=3

4. Ввести и виконати програму.

4.ВИКОРИСТАННЯ ПІДПРОГРАМ

Для визначення підпрограми – функції в її опису може бути використана будь-яка кількість операторів, число вихідних величин також довільно.

Відладка підпрограми може виконуватися окремо від основної програми, це самостійна програмна одиниця.

При визначенні підпрограми типа **Function** використовуються формальні параметри; інформація, необхідна при виконанні ПП, поставляється фактичними параметрами.

Термін "підпрограма" (ПП) об'єднує поняття процедури, визначеної за допомогою SUB-структури, та зовнішньої функції, визначеної **Function** - структурою.

Підпрограми вводяться в пам'ять після основної програми (Alt, EDIT, NEW SUB/FUN, ввод імені, потім текст підпрограми), або перед основною програмою.

Якщо створена програма в оточенні QB і вона містить визначення і виклик SUB або **Function**, то при зберіганні на диску ви помітите, що з'явився новий оператор **DECLARE**, структура якого

DECLARE {FUNCTION / SUB} ім'я[parameters]

ВИКОРИСТАННЯ ЗОВНІШНІХ НЕСТАНДАРТНИХ ФУНКЦІЙ

Опис зовнішньої функції:

FUNCTION ім'я [(список формальних параметрів)]

оператори

ім'я=вираз

END FUNCTION

Звертання до зовнішньої функції:

ім'я (список фактичних параметрів)

Можливо визивати зовнішню функцію рекурсивно.

Вказівки до вивчення розділу

1. Задани два масиви натуральних чисел $A(t_1)$, $B(t_2)$, де t_1 і t_2 – кількість елементів. Потрібно обчислити середнє арифметичне сум елементів цих масивів. Скласти програму зі звертанням до зовнішньої функції для розрахунку суми елементів одномірного масиву.

RND(1) - вибір випадкових чисел із діапазону (0,1);

(B-A)*RND(1)+A - вибір випадкових чисел із діапазону (A,B);

INT((L-K +1)*RND(1)+ K)- вибір цілих випадкових чисел із діапазону (K,L);

RANDOMIZE TIMER - вхід у таблицю випадкових чисел.

2. Ввести і виконати програму:

REM середнє арифметичне сум елементів 3-х масивів

CLS

INPUT t1, t2

DIM a(t1), b(t2)

RANDOMIZE TIMER

FOR i = 1 TO t1

```

a(i) = INT(10*RND(1) ): PRINT a(i); : NEXT: PRINT
RANDOMIZE TIMER
FOR i = 1 TO t2
b(i) = INT(10*RND(1)): PRINT b(i); : NEXT
sr = (sum(a(), t1) + sum(b(), t2)) / 2
PRINT sr
END

```

```

FUNCTION sum (x(), n)
s = 0
FOR i = 1 TO n
s = s + x(i)
NEXT
sum = s
END FUNCTION

```

3. Використовуючи приведені описи зовнішніх функцій, скласти програму для рішення задачі:

Маємо 7 злитків. 4 виготовлені у першому цеху і 3 злитка - у другому цеху. Наугад беруться 3 злитка. Чому дорівнює можливість того, що серед відібраних є 2 злитка, виготовлених у першому цеху?

Рішення задачі визначається формулою

$$P(A) = C_4^2 * C_3^1 / C_7^3,$$

де A - подія появи двох злитків із першого цеху й одного з другого цеху.

Опис зовнішньої функції для обчислення кількості сполучень із n по k відповідно формулі $C_n^k = n! / (k!(n-k)!)$, має вигляд:

```

FUNCTION C(n,k)
c=factor(n)/(faktor(k)*faktor(n-k))
END FUNCTION

```

Опис зовнішньої функції для обчислення факторіала:

```

FUNCTION faktor(t)
fak=1
FOR i=1 TO t
fak=fak*i
NEXT i

```

```
faktor=fak  
END FUNCTION
```

4. Ввести програмні одиниці в пам'ять і одержати результат у виді:
Можливість дорівнює 0.5142.

Програма:

```
CLS  
P(A) = C(4, 2) * C(3, 1) / C(7, 3)  
PRINT "Можливість дорівнює" ; P(A)  
END
```

```
FUNCTION C (n, k)  
C = faktor(n) / (faktor(k) * faktor(n - k))  
END FUNCTION
```

```
FUNCTION faktor (t)  
fak = 1  
FOR i = 1 TO t  
fak = fak * i  
NEXT  
faktor = fak  
END FUNCTION
```

5. ВИКОРИСТАННЯ ПІДПРОГРАМ SUB – СТРУКТУРИ

Підпрограма називається також підпрограмою-процедурою типа SUB і являється самостійною програмною одиницею, має поширені можливості при програмуванні. Звернення до підпрограми здійснюється спеціальним оператором виклику, який дає вказівку виконати підпрограму.

Написання програми, що містить ПП, здійснюється в три кроки. Спочатку треба визначити ПП, далі оголосити її і нарешті ввійти в неї.

Визначення ПП містить в собі оператори, які виконуються підпрограмою, та інформацію, що передається в викликаючу програму.

Структура SUB має вигляд :

```
SUB ім'я(список формальних параметрів)  
оператори  
END SUB
```

Звертання до підпрограми з основної програми або іншої програмної одиниці здійснюється оператором

```
CALL ім'я(список фактичних параметрів)
```

Коли виконання ПП дійде до рядка END SUB, керування передається у ту програму, що викликала і продовжується її виконання з рядка після оператора CALL.

Якщо ви, знаходячись в середині ПП, бажаєте вийти з неї "достроково", то використовуйте оператор

```
EXIT SUB
```

Масив у список параметрів включається у формі: **ім'я()**

Опис масиву приводиться в основній програмі. Якщо фактичний параметр у списку оператора CALL укладений у круглі скобки, то його значення не замінюється обчисленням у підпрограмі після звертання до неї.

Вказівки до вивчення розділу

1. Скласти програму для обчислення середнього значення елементів одномірного масиву $A = \{12, 3, -4, 9.5, -17.4\}$, передбачивши в ней використання підпрограми SUM для обчислення суми S елементів одномірного масиву .

2. Ввести основну програму і підпрограму SUM у пам'ять, одержати результат у формі:

“середнє значення =”

Програма:

```
CLS  
DIM a(5)  
FOR i = 1 TO 5  
READ a(i)  
NEXT  
DATA 12, 3, -4, 9.5, -17.4  
CALL sum(a(), 5, s1)
```

```

sr = s1 / 5
PRINT "Середнє значення =", sr
END
SUB sum (x(), n, s1)
s1 = 0
FOR i = 1 TO n
s1 = s1 + x(i)
NEXT
END SUB

```

4. Додати в основну програму опис масиву d і його обчислення по виразу $d(i) = a(i)^2$, а також повторне звертання до підпрограми SUM для обчислення суми S елементів масиву d .

5. У самій підпрограмі додати вивод на екран значення S .

6. Виконати програму, зафіксувати результати.

6.ПРОГРАМУВАННЯ ЗАДАЧ ПО УПОРЯДКУВАННЮ МАСИВІВ З ВИКОРИСТАННЯМ ЗОВНІШНЬОЇ ПІДПРОГРАМИ

Підпрограма для упорядкування одномірному масиву $A(i)$ по убиванню:

```

SUB YP(a())
  n = LBOUND(a,1)
  m = UBOUND(a,1)
1  f = 1
  FOR i =n+1 TO m-1
  IF a(i) < a(i+1) THEN
  SWAP a(i), a(i+1)
  f = 2
  END IF
  NEXT
  IF f = 2 THEN GOTO 1
END SUB

```

Вказівки до вивчення розділу

1. Використовуючи приведену вище підпрограму, скласти зведення про браковані труби.

ВХІДНІ ДАНІ

№ ПЛАВКИ	ОГЛЯНУТО, ШТ.	ЗАБРАКОВАНО, ШТ.
1075	12	1
1080	18	2
1075	10	2
1065	8	1
1070	15	2
1075	13	1
1080	11	1

Зведення розташувати в порядку зростання номера плавки. Зведення про плавку з тим самим номером повинні бути просумовані.

При введенні даних у програму використовувати наступні умовні позначки:

А - масив, що складається з номерів плавко;

В- масив, що містить зведення про кількість оглянутих труб;

С - масив, що містить зведення про кількість забракованих труб;

А1, В1, С1 - робочі масиви.

2. Розглянути наступну програму для вище приведеної задачі і виконати її.

```
CLS
```

```
INPUT n
```

```
DIM a(n), b(n), c(n), a1(n), b1(n), c1(n)
```

```
DATA 1075,12,1
```

```
DATA 1080,18,2
```

```
DATA 1075,10,2
```

```
DATA 1065,8,1
```

```
DATA 1070,15,2
```

```
DATA 1075,13,1
```

```

DATA 1080,11,1
FOR i = 1 TO n : READ a(i), b(i), c(i) : NEXT
REM Підсумовування зведень по плавках з однаковими номерами
FOR i = 1 TO n
p = b(i) : r = c(i)

FOR j = i + 1 TO n
IF a(i) = a(j) THEN 1
GOTO 2
1 a(j) = 0: p = p + b(j): b(j) = 0: r = r + c(j): c(j) = 0 ' Обнуління
2 NEXT
b(i) = p : c(i) = r : NEXT i ' Кінець підсумовування
REM Видалення обнулених елементів масивів А, В, С
' і формування масивів А1, В1, С1
k = 0
FOR i = 1 TO n
IF a(i) <> 0 THEN 3
GOTO 4
3 k = k + 1: a1(k) = a(i): b1(k) = b(i): c1(k) = c(i)
4 NEXT ' Кінець видалення і формування
REM переприсвоювання масивів і змінних
FOR i = 1 TO k: a(i) = a1(i): b(i) = b1(i): c(i) = c1(i): NEXT
n = k ' Кінець переприсвоювання
PRINT "N плавки ", "          оглянуто, шт.", "          забраковано,
шт."
PRINT
CALL YPNP(a(),b(),c())
FOR i = 1 TO k
PRINT a(i), , b(i), , c(i)
NEXT

SUB YPNP(a(),b(),c())
n = LBOUND(a,1)
m = UBOUND(a,1)
1 f = 1

```



```

FOR i =n+1 TO m-1
IF a(i) < a(i+1) THEN
SWAP a(i), a(i+1)
SWAP b(i), b(i+1)
SWAP c(i), c(i+1)
f = 2
END IF
NEXT
IF f = 2 THEN GOTO 1
END SUB

```

3. Виконати індивідуальне завдання.

7. РІШЕННЯ НЕЛІНІЙНИХ РІВНЯНЬ МЕТОДОМ ПРОСТОЇ ІТЕРАЦІЇ

Рішення нелінійних рівнянь виду $f(x) = 0$ полягає в пошуку одного або декількох коренів. Передбачається, що кожний із коренів відділен (локалізован) на відрізку.

Роздивимося **метод простої ітерації** для уточнення кореня з заданим ступенем точності ϵ . Нехай корінь відділен на відрізку $[a,b]$.

Ідея методу: рівняння $f(x) = 0$ приводиться до виду $x = j(x)$ і корінь шукається як границя послідовності:

$$x_1 = j(x_0),$$

$$x_2 = j(x_1),$$

$$x_3 = j(x_2),$$

.....

Умова збіжності методу: функція $j(x)$ вибирається таким чином, щоб виконувалася умова

$$q = |j'(x)| \leq 1 \quad \text{на інтервалі } [a,b]$$

Якщо умова збіжності не виконується, то підбирається інша ітераційна формула.

Умовою виходу з процесу ітерації, є досягнення заданого ступеня точності ϵ

$$|x_i - x_{i-1}| < \epsilon$$

За нульове наближення береться будь-яке число з відрізка $[a,b]$.

Вказівки до вивчення розділу

1. Скласти блок - схему і програму для уточнення кореня нелінійного рівняння **виду $5x^3 - 20x + 3 = 0$** з точністю до $\epsilon = 10^{-4}$, корінь відділен на відріжку $[0,1]$.

Приведемо рівняння до ітераційного виду. Це можна зробити декількома засобами:

a) $x = x + (5x^3 - 20x + 3)$

$$q = |\phi'(x)| = |15x^2 - 19x| > 1 \text{ на } [0,1]$$

b) $x = (5x^3 + 3) / 20$

$$q = |\phi'(x)| = |15x^2 / 20| < 1 \text{ на } [0,1]$$

З отриманих розрахунків випливає, що потрібно використовувати ітераційну формулу **виду $x = (5x^3 + 3) / 20$** .

Умова закінчення ітераційного процесу: $|x_i - x_{i-1}| < \epsilon$

Програма:

CLS

DEF fnf(x)=(5 * x ^ 3 + 3) / 20

x0 = 0

eps = 0.0001

x = x0

50 y=fnf(x)

IF abs(y-x)> eps THEN x=y: GOTO 50

PRINT "Корінь рівняння ="; y

END

8. РІШЕННЯ НЕЛІНІЙНИХ РІВНЯНЬ МЕТОДОМ ПОЛОВИННОГО РОЗПОДІЛУ

Роздивимося метод половинного розподілу для уточнення кореня нелінійного рівняння виду $f(x) = 0$ з заданим ступенем точності ϵ . Нехай корінь відділен на відрізку $[a,b]$.

Метод половинного розподілу складається в послідовному розпо-

ділі відрізка $[a,b]$ навпіл:
$$C = \frac{A+B}{2}$$

Умова закінчення обчислювального процесу
$$\frac{|B-A|}{2} < \epsilon$$

Вказівки до вивчення розділу

1. Скласти блок схему і програму для уточнення з точністю до $\epsilon = 10^{-4}$ кореня нелінійного рівняння $5x^3 - 20x + 3 = 0$, відділеного на відрізку $[0,1]$.

Програма:

```
CLS
DEF fny(x) = 5 * x ^ 3 - 20 * x + 3
INPUT a0, b0, eps
a=a0:b=b0
10 c=(a+b)/2
IF ABS (b-a) / 2) < eps THEN PRINT "c="; c: end
IF fny(a) * fny(c) < 0 THEN b = c ELSE a = c
GOTO 10
END
```

9. СТВОРЕННЯ БІБЛІОТЕКИ КОРИСТУВАЧА

Призначення бібліотек - заховати від сторонніх текст своїх програм. Відтранлювавши їх в .EXE файли, ви зможете зберегти "фірмові" сек-

рети і в той же час дати можливість іншим особам користуватися вашими програмами.

Всі модулі, які ви поміщаєте у бібліотеки, мусять бути добре відлагодженими, щоб не містили ніяких помилок, і трансльовані в .EXE файли.

Створення бібліотеки користувача складається із наступних етапів:

1. Скопіювати модулі, які поміщатимуться в бібліотеку;
2. За допомогою команди Load File завантажити копії модулів в оточення;
3. Вибрати команду Make Library . В діалоговий бокс, що з'явиться, ввести ім'я бібліотеки й закінчити діалог.

В пакеті QB є невеличка бібліотека з ім'ям QB.LIB, яка містить стандартні програми обробки переривань і таке інше.

Можливо також створювати бібліотеки за допомогою оточенні DOS, використовуючи програму LINK з відповідними опціями.

Виконання цієї роботи потребує забезпечення повного доступу користувача до диска, із якого завантажується оболонка Quick Basic. Впливає також встановити опції **SET PATHS** для вказівки шляхів доступу до файлів, використовуваним при створенні бібліотеки, тобто повинно бути:

Файл	шлях доступу
Executable files (.exe,.com)	C:\QB
Include files (.bi,.bas)	D:\
Library files (.lib,.qlb)	C:\QB; D:\
Help file (.hlp)	C:\QB

Примітки: завантаження забезпечує лаборант.

Вказівки до вивчення розділу

1. Переконалися у вірності зазначених шляхів доступу до файлів.

2. Ввести в оперативну пам'ять головний модуль, що складається з операторів опису підпрограм

```
DECLARE SUB MA(A(),N,M)
```

```
DECLARE SUB UPR (B(),N)
```

Ввести в оперативну пам'ять підпрограму MA обчислення максимального елемента одномірного масиву A

```
SUB MA(A(),N,M)
```

```
DIM A(N)
```

```
M=A(1)
```

```
FOR i=2 TO N
```

```
IF M<=A(i) THEN M=A(i)
```

```
END SUB
```

Ввести в оперативну пам'ять підпрограму UPR упорядкування одномірного масиву B по зростанню

```
SUB UPR(B(),N)
```

```
DIM M(N)
```

```
FOR i=1 TO N-1
```

```
P=B(i):K=1
```

```
FOR J=i+1 TO N
```

```
IF B(J)>=P THEN 20
```

```
P=B(J):K=J:B(K)=B(i):B(i)=P
```

```
20 NEXT J
```

```
NEXT i
```

```
END SUB
```

3. Записати головний модуль разом із підпрограмами як єдиний текстовий файл на диску E: з ім'ям ROM.

4. Використовуючи ALT, RUN, MAKE LIBRARY, створити бібліотеку з тим же ім'ям ROM, ввівши ім'я у відповідь на запит FILE NAME. При цьому на екрані повинно з'явитися:

```
DECLARE SUB MA(A(),N,M)
```

```
DECLARE SUB UPR(B(),N)
```

5. Створити на диску E файл-заголовок із тим же ім'ям ROM, але з розширенням .bi .

Для цього:

а) вийти з QB у VC;

б) скопіювати (F5) файл ROM.bas під ім'ям ROM.bi на той же диск D:

6. Виконати редагування (F4) файла ROM.bi, удалив із нього усе , крім двох рядків :

```
DECLARE SUB MA(A(),N,M)
DECLARE SUB UPR(B(),N)
```

7. Використовуючи ESC, SAVE, зберегти ці рядки на диску D:.

8. Перевірити наявність таких файлів:

```
ROM. bas } на диску D:
ROM. bi }
```

```
ROM.QBJ } на диску C:
ROM. LIB: }
ROM.QLB }
```

9. У випадку відсутності якогось із цих файлів повторити всю процедуру створення бібліотеки.

10.ВИКОРИСТАННЯ БІБЛІОТЕКИ КОРИСТУВАЧА

Передбачається використання конкретної підпрограми з бібліотеки, створеної користувачем.

Початкове завантаження здійснюється заздалегідь.

\$INCLUDE:'ім'я файла-заголовок бібліотеки' - рядок, що включається у програму, якщо в ній здійснюється звертання до підпрограм бібліотеки. Звертання до підпрограм здійснюється за допомогою оператора CALL.

Вказівки до вивчення розділу

1. Вийти з VC у DOS.

2. Завантажити оболонку QB і бібліотеку з ім'ям ROM в оперативну пам'ять за маршрутом:

```
C:\QB\qb_/LROM
```

3. Ввести програму

```
$INCLUDE:"ROM.bi"
INPUT N
DIM A(N)
```

```

FOR i=1 TO N
INPUT A(i)
NEXT i
CALL MA(A(),N,M)
PRINT "MAX=";M
END

```

4. Виконати програму для вхідного масиву: 27.6, 32.4, 31.8, 43.5, 18.7, 30.4, 29.8, 42.8, 26.3

5. Скласти і виконати програму для упорядкування вихідного масиву по зростанню з використанням відповідної підпрограми зі створеної бібліотеки.

11.СТВОРЕННЯ І ВИКОРИСТАННЯ ФАЙЛА ДАНИХ ПОСЛІДОВНОГО ДОСТУПУ

Оператор відкриття файла:

OPEN "ім'я файла. dat" FOR INPUT/OUTPUT AS #N,

де N - номер файла;

INPUT - указується при зчитуванні з файла;

OUTPUT - указується при записі у файл.

Оператор пересилки у файл послідовного доступу:

PRINT #N, список імен полей

Роздільники в списку:

;- для єдиного запису;

"," - для окремих полей.

Оператор зчитування даних із файла послідовного доступу:

INPUT #N, список змінних

Оператор закриття файла: **CLOSE #N**

Макет файла

Ім'я файла	Довжина запису	Число записів
1	2	3
SAT1	35	5

1	2	3
Імена полей у записі	Призначення полей	Максималь-ний розмір
FAM\$	Прізвище	15
DLG\$	Посада	10
STG	Стаж	2
ZRP	Зарплата	6

Вказівки до вивчення розділу

1. Відповідно до макета скласти і виконати програму для створення файла послідовного доступу з записами без поділу полей:

ІВАНОВ А.П.	інженер	20	400
АСТАХОВ С.Н.	лаборант	15	150
СИДОРЕНКО В.В.	технік	3	140
ВАРЛАМОВА Е.А.	інженер	10	300
УСОВА П. О.	лаборант	8	140

2. Зберегти програму на диску.

3. Очистити оперативну пам'ять.

4. Скласти і виконати програму зчитування записів із файла послідовного доступу.

5. Зберегти програму на диску.

6. Очистити оперативну пам'ять.

7. Викликати в оперативну пам'ять програму створення файла послідовного доступу, відредагувати її для одержання записів із поділом полів.

8. Виконати програму, очистити оперативну пам'ять.

9. Викликати в оперативну пам'ять програму зчитування з файла послідовного доступу, відредагувати її з метою одержання даних окремих полей.

10. Виконати програму.

11 Вносячи відповідні зміни у програму, вивести на екран: а) прізвище і посаду співробітника, що має максимальний стаж роботи; б) дані для співробітника з конкретним прізвищем.

12. СТВОРЕННЯ І ВИКОРИСТАННЯ ФАЙЛА ДАНИХ ДОВІЛЬНОГО ДОСТУПУ

Оператор відкриття файла:

OPEN "ім'я файла. dat" AS #N LEN = L,

де N - номер файла ;

L - довжина запису.

Оператор опису полей:

FIELD #N, L1 AS змінна 1, L2 AS змінна2,...

Оператор запису файла в буфер:

LSET змінна 1 = значення :LSET змінна 2 = значення:...

Функції перекладу числових змінних у символну форму:

MKI\$ - для цілих чисел;

MKSS\$ - для дійсних чисел;

MKD\$ - для дійсних чисел подвійної точності.

Функції перекладу символних змінних у числову форму:

CVI - для цілих чисел;

CVS - для дійсних чисел;

CTD - для дійсних чисел підвищеної точності.

Оператор запису файла даних на диск:

PUT #N

Оператор закриття файла:

CLOSE #N

Оператор зчитування з диска:

GET #N, [M]

де N - номер файла;

M - номер запису (якщо відсутній, то зчитується слідуючий запис).

Функція визначення номера поточного запису у файлі:

LOC(N),

де N - номер файла.

Вказівки до вивчення розділу

1. В відповідності з макетом і завданням із лабораторної роботи №11, але з ім'ям файла PRZ, скласти і виконати програму створення файла довільного доступу.

2. Записати програму на диск.

3. Скласти і виконати програму зчитування записів із наявного файла довільного доступу з ім'ям PRZ, що відповідає макету з лабораторної роботи №11 .

4. Вносячи відповідні зміни в програму, вивести на екран:

а) значення полей "Прізвище" і "Зарплата" для всіх записів;

б) першу і третю запис;

в) всю інформацію для "співробітників із стажем більш 5 років";

г) організувати вихід із циклу після зчитування 3-х записів і вивести номер поточного запису.

13. ГРАФІЧНІ МОЖЛИВОСТІ МОВИ QUICK BASIC. ПОБУДОВА ЛІНІЙ, ПРЯМОКУТНИКІВ

Завдання режиму вивода SCREEN N,

де N=0 - для текстового виводу;

N=1 , N=2, N=9, N=12- для графічного виводу.

COLOR N1,N2

де N1- номер кольору переднього плану

N2- номер кольору фону

Оператор завдання точки

PSET (№1, №2), №3

де №1 - номер стовпчика } - координати точки

№2 - номер рядка

№3 - номер кольору.

Оператор побудови відрізка прямої

LINE (№1, №2) - (№3, №4), №5

де №1, №2 - координати початку відрізка (якщо відсутні, то відрізок починається в останній точці, заданої попереднім оператором);

№3, №4 - координати кінця відрізка;

№5 - номер кольору лінії (якщо відсутній, то колір білий).

Оператор побудови прямокутника

LINE (№1, №2) - (№3, №4), №5, B

де №1, №2 - координати точки верхнього лівого кута прямокутника;

№3, №4 - координати точки правого нижнього кута прямокутника;

№5 - номер кольору лінії (якщо відсутній, то колір білий);

B - ознака побудови прямокутника.

Оператор побудови пофарбованого прямокутника

LINE (№1, №2) -(№3, №4), №5 , BF

Вказівки до вивчення розділу

1. Відобразити червону точку в центрі екрана.
2. Провести лінію білого кольору між точками (55, 75) і (80, 60).
3. Побудувати прямокутник, зафарбувавши його червоним кольором, провести дві білі діагоналі.
4. Зобразити 2 перехресних прямокутника, нижній - блакитний, верхній - червоний.
5. Побудувати ломану лінію (секцію) із 2-х відрізків білого кольору.
6. На основі виконаного в пункті 5, побудувати ломану лінію, використовуючи оператор циклу, із секціями різного кольору.
7. Внести зміни в програму з метою повторення отриманого малюнка п'ять разів по рядках.

14. ПОБУДОВА КОЛ, ЕЛІПСІВ, ДУГ

Оператор побудови кол, дуг, еліпсів:

CIRCLE (x, y), R, N, an, ak, k

де x,y - координати центру кола;

R - радіус кола;

N - номер кольору;

an і ak - початок і кінець дуги, що задаються в радіанах.

Якщо $a_n < a_k$, то дуга вичерчується проти годинної стрілки, у протилежному випадку - по годинній стрілці.

Якщо a_n і a_k задати зі знаком "-", то будуть проведені радіуси, тобто одержимо сектор.

k - коефіцієнт стиску, задається у виді звичайного дробу. Якщо $k < 1$, то одержуємо еліпс, витягнутий по горизонталі, якщо $k > 1$, то одержуємо еліпс витягнутий по вертикалі.

Оператор заповнення цвітом заданої області:

PAINT (x, y) [,n] [,n1],

де x, y - координати будь-якої точки області, що зафарбовується;

n - номер кольору фарбування;

$n1$ - номер кольору межі.

Оператор завдання положення на екрані точки наступного виводу

LOCATE n1, n2,

де $n1$ - номер рядка (0-24);

$n2$ - номер стовпчика (0-40).

Вказівки до вивчення розділу

1. Відобразити в центрі екрана коло радіусом $R=50$. Область, обмежену колом, зафарбувати блакитним цвітом.

2. Зобразити на екрані дуги кол.

3. Зобразити 2 еліпса з загальним центром: один - витягнутий по горизонталі, інший - по вертикалі.

4. Зобразити коло радіуса $R = 20$, що переміщається по діагоналі екрану знизу нагору.

5. Відобразити на екрані багатократну побудову концентричних кол з переменно зростаючим та зменшуваним радіусом.

6. Використовуючи приведену програму, побудувати кругову діаграму розподілу успішності в групі з 25 чоловік:

відмінно - 15%

добре - 45%

задовільно - 35%

незадовільно - 5%.

Програма:

```

INPUT N
DIM P(N)
SCREEN 1:CLS
i=0
FOR i= 1 TO N
INPUT P(i)
T=T+P(i):NEXT I
S=0:R=50
FOR i= 1 TO N
E=6.2832*P(i)/T+S
M=(S+E)/2
CIRCLE (160,100), R,3,-S,-E+0.0001
PAINT (160+COS(M)*R*0.75,100-SIN(M)*R*0.75), i MOD 4,3
LOCATE (100-SIN(M)*(R-8))/8,(160+COS(M)*(R+16))/8
PRINT P(i)
S=E
NEXT i
END

```

15. ВИКОРИСТАННЯ ОПЕРАТОРА DRAW

Оператор креслення ліній

DRAW “підкоманди”

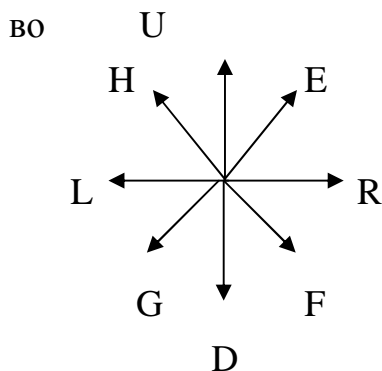
Підкоманди записуються без розділювальних знаків.

Підкоманди:

Un - переміщення на n позицій нагору

Dn - переміщення на n позицій униз

Rn - переміщення на n позицій упра-



Ln - переміщення на n позицій уліво

En - переміщення на n позицій
нагору – праворуч по діагоналі

Fn - переміщення на n позицій
униз - праворуч по діагоналі

Gn - переміщення на n позицій

униз – ліворуч по діагоналі
Nn - переміщення на n позицій
нагору – ліворуч по діагоналі

B - лінію не проводити при наступних переміщеннях;

Mn1,n2 - перехід у задану точку, де n1,n2 - абсолютні координати точки;

N - повернення в поточну точку;

Ак - поворот проти годинної стрілки всіх наступних переміщень і проведених ліній. К – задає кут повороту

0 - 0° 2 - 180°

1 - 90° 3 - 270°

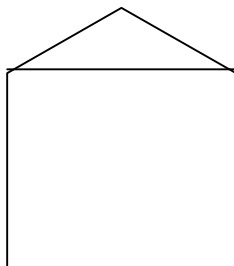
Sk - зменшення або збільшення довжини ліній. К – масштабний коефіцієнт

1 - $\frac{1}{4}$ 8 - 2

4 - 1 36 - 9 і так далі.

Вказівки до вивчення розділу

1. За допомогою оператора DRAW намалювати зображення



Програма:

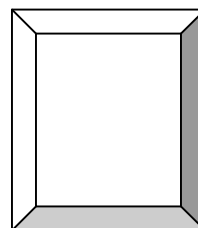
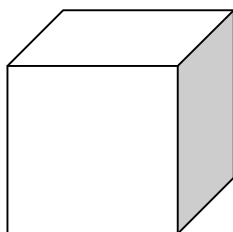
```
SCREEN 12
```

```
PSET (200, 200)
```

```
DRAW " e100 f100 l200 d200 r200 u200"
```

```
END
```

2. За допомогою оператора DRAW намалювати:



16. ВИКОРИСТАННЯ ОПЕРАТОРІВ GET, PUT.

“ПОЖВАВЛЕННЯ” ЗОБРАЖЕННЯ

Оператор для запам'ятовування зображення в пам'яті:

GET (x1,y1)-(x2,y2), ім'я масиву,

де (x1,y1), (x2,y2) - координати прямокутника, що окреслює зображення.

Масив, ім'я якого зазначено, береже інформацію про точки зображення. Він повинний бути попередньо описаний.

Оператор відтворення зображення:

PUT (x,y), ім'я масиву,

де (x,y) - координати точки, що визначає місце відтворення зображення (повинна відповідати лівому верхньому куту прямокутника).

Повторне використання оператора **PUT** стирає зображення.

Для “пожвавлення” використовують оператор **PUT** із поперемінним відтворенням і стиранням зображень сусідніх положень, що імітують динаміку руху.

Для музичного супроводу використовується оператор:

PLAY “підкоманди”

Підкоманди записуються без розділювальних знаків. Підкоманди позначення нот другої октави:

до - C	соль - G	дієз +
ре - D	ля - A	бемоль -
мі - E	сі - B	
фа - F	пауза - P	

Тривалість нот і паузи позначаються числом, зазначеним після ноти

1 - ціла	8 – восьма
2 - половина	і так далі
4 – чверть	

Вказівки до вивчення розділу

1. Ввести і виконати програму, що дає зображення "чоловічка"

CLS

SCREEN 1

COLOR 0,1

PSET (155,98)

PSET (165,98)

```

CIRCLE (160,97),8,3,4.2,5.3
CIRCLE (160,100),10,3
LINE (167,120)-(174,130)
CIRCLE (160,130),20,3,0,6.283,9/3
LINE (162,148)-(169.175)
LINE (153,120)-(143,130)
LINE (158,147)-(151,175)
END

```

3.Внести доповнення в програму для забезпечення прямування рук нагору й униз із музичним супроводом.

17. ПОБУДОВА ГРАФІКІВ ФУНКЦІЙ ЗАСОБАМИ QBasic 4.5

Такі задачі, як вибор початку координат, масштабу зображення графіка, способу відображення і кроку зміни аргументу приводять до наступного узагальнюючого запису графіка оператора, що малює, PSET:

$$\text{Pset (} x * C + A , -y * D + B \text{),}$$

де A і B - величини, що визначають зрушення початку координат графіка

C і D - масштабні коефіцієнти зображення графіка.

Наприклад, графік функції $\text{SIN}(x)$ можна одержати за допомогою наступної програми, у якій $A = 320$, $Y = 240$, $C = 40$, $D = 100$

```

SCREEN 12                                ‘ переклад екрана в графічний режим
FOR x = -6.28 TO 6.28 STEP .01
PSET (x * 40 + 320, -SIN(x) * 100 + 240)
NEXT x
LINE (0, 240)-(640, 240)                ‘ побудова осей координат
LINE (320, 0)-(320, 480)
LOCATE 1, 2                              ‘ позиціонування заголовка екрана
PRINT "y = sin(x)"
LOCATE 16, 80                            ‘ позиціонування підпису до осі X
PRINT "x"

```



```

LOCATE 17, 1          ‘ позиціонування розмітки до осі X
PRINT "-6.28"
LOCATE 17, 20
PRINT "-3.14"
LOCATE 16, 40
PRINT "0"
LOCATE 17, 60
PRINT "3.14"

```

```

LOCATE 1, 37          ‘ позиціонування підпису до осі B
PRINT "y"

```

```

LOCATE 13, 37        ‘ позиціонування розмітки до осі B
PRINT "0.1"
LOCATE 8, 37
PRINT "0.3"
LOCATE 3, 37
PRINT "0.5"
LOCATE 18, 36
PRINT "-0.1"
LOCATE 23, 36
PRINT "-0.3"
LOCATE 28, 36
PRINT "-0.5"
END

```

Оформлення графіка функції складається з зображення осей координат і написів, що пояснюють. Нижче приведена програма, у якій додатково зображуються і позначаються координатні осі.

Спростити процедуру розрахунку масштабних коефіцієнтів і величину зсуву осей координат відповідно до координатної області розташування графіка й обраний графічний режим екрана дозволяє оператор: **WINDOW (xn, yn) - (xk, yk)**, де x_n, y_n і x_k, y_k - координати будь-яких діагональних вершин вікна, що відкривається, у рамках якого може бути розміщений графік. Вибираються ці координати таким чином, щоб максимальні і мінімальні зна-

чення графіка по будь-яких координатах були видні в межах вікна, що розкривається, тобто як мінімум $x_n \geq x_{\min}$, $y_n \geq y_{\min}$ і $x_k \leq x_{\max}$, $y_k \leq y_{\max}$. Оператор **WINDOW** (x_n, y_n) - (x_k, y_k) дозволяє вводити координати в реальних розмірних одиницях, і не прив'язувати розрахунки до розмірності графічного екрана, вираженого в крапках (наприклад для графічного режиму SCREEN 12 - 640x480).

```
xn = -6.28
xk = 6.28
yn = SIN(xn / 4) - .2
yk = SIN(xk / 4) + .2
PRINT yn, yk
SCREEN 12
WINDOW (xn, yn)-(xk, yk)
FOR x = xn TO xk STEP .01
y = SIN(x)
PSET (x, y)
'PRINT y
NEXT x
LINE (xn, 0)-(xk, 0)
LINE (0, yk)-(0, yn)
LOCATE 1, 2
PRINT "y = sin(x)"
FOR x = xn TO xk STEP 1.57
LINE (x, 0)-(x, .03)
NEXT x
FOR y = yn TO yk STEP .2
LINE (0, y)-(.05, y)
NEXT y
LOCATE 16, 80
PRINT "x"
```

```
LOCATE 16, 1
PRINT "-6.28"
LOCATE 16, 20
PRINT "-3.14"
LOCATE 16, 40
PRINT "0"
LOCATE 16, 60
PRINT "3.14"
LOCATE 1, 37
PRINT "y"
LOCATE 13, 37
PRINT "0.1"
LOCATE 8, 37
PRINT "0.3"
LOCATE 3, 37
PRINT "0.5"
LOCATE 16, 60
PRINT "3.14"

LOCATE 18, 36
PRINT "-0.1"
LOCATE 23, 36
PRINT "-0.3"
LOCATE 28, 36
PRINT "-0.5"
LOCATE 16, 60
PRINT "3.14"

END
```

НАВЧАЛЬНЕ ВИДАННЯ

Швачич Генадій Григорович

Пасинков Володимир Миколайович

Гуляєва Олена Анатолієвна

Єфанова Ліна Михайлівна

Скороход Борис Макаров ич

Романова Наталія Сергійовна

Кузьменко Вячеслав Віталійович

АЛГОРИТМІЧНА МОВА QBASIC

Навчальний посібник

Частина 2

Тем.план 2004,поз.159