МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний технічний університет
«Харківський політехнічний інститут»

МЕТОДИЧНІ ВКАЗІВКИ
до лабораторної роботи
«Організація галуження у програмах мовою Delphi»

з курсу «Програмування»
для студентів напряму 6.040302 – Інформатика
(спеціалізація «Соціальна інформатика»)

Затверджено редакційно-видавничою
радою університету,
протокол № 3 від 28.12.09.

Харків НТУ «ХПІ» 2010

Методичні вказівки до лабораторної роботи «Організація галуження у програмах мовою Delphi» з курсу «Програмування» для студентів напряму 6.040302 – Інформатика (спеціалізація «Соціальна інформатика») / Уклад. М. І. Безменов. – Х. : НТУ «ХПІ», 2010. – 17 с.

Укладач М. І. Безменов

Рецензент Л. М. Любчик

Кафедра системного аналізу і управління

Мета роботи

Освоєння методики організації керування процесом обчислень за допомогою операторів, що перевіряють умову.

1. ТЕОРЕТИЧНІ ОСНОВИ

1.1. Умовний оператор

Досить часто в програмі необхідно виконувати деякі дії в тому випадку, коли є істинною деяка умова. Найпростішим оператором, що використовується у цьому випадку, є оператор **if** (оператор розгалуження, умовний оператор):

```
if вираз_умова then оператор
```

Якщо вираз_умова істинний (`True`), то виконується оператор, розташований після службового слова **then**, а потім – наступний оператор; якщо значенням виразу_умови є `False`, то оператор, що стоїть після **then**, пропускається (див. рис. 1.1, *а*). Як вираз_умова може бути записаний довільний логічний вираз.

Наведена конструкція зручна у випадку, коли деяка дія повинна виконуватися тільки при позитивній відповіді на запитання і не виконуватися, якщо отримано негативну відповідь. Якщо ж при негативній відповіді необхідно виконати іншу дію, то оператор **if** доведеться повторити з умовою, протилежною умові, що перевіряється в першій конструкції:

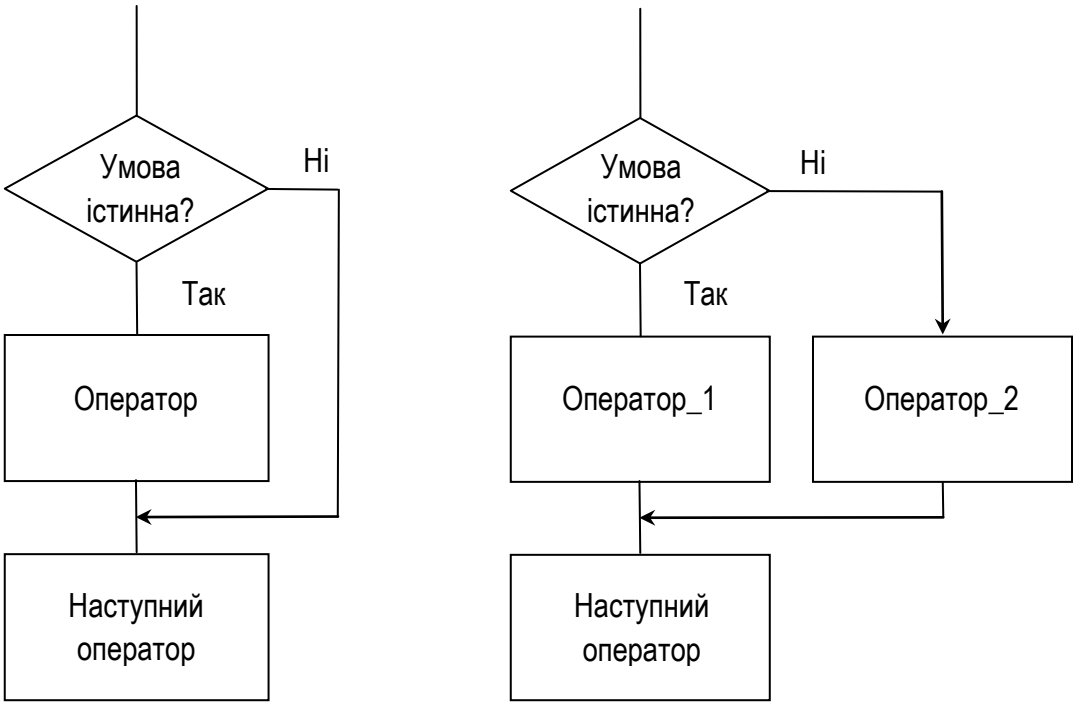
```
if вираз_умова then оператор_1;
if протилежна_умова then оператор_2
```

Більш загальною конструкцією є конструкція, ідея якої полягає в тому, щоб здійснювати перевірку не того, потрібно або не потрібно виконувати деяку дію, а того, яка із двох дій повинна бути виконана (див. рис. 1.1, *б*). При цьому обидві дії не виконуються ніколи.

Повна форма оператора **if** така:

```
if вираз_умова then оператор_1
else оператор_2
```

Оператори оператор, оператор_1 і оператор_2 можуть бути як простими операторами, так і складеними операторами.



а *б*

Рис. 1.1. Можливі схеми дії оператора **if**

Складеним оператором називається послідовність операторів, укладена у операторні дужки, тобто записана між службовими словами **begin** та **end**:

```
begin
   оператори
end
```

Іноді деяку дію потрібно виконувати тільки при хибності виразу_умови. Тоді після службового слова **then** вказують порожній оператор, позначуваний відсутністю будь-якого оператора між службовими словами **then** та **else**:

```
if вираз_умова then
else оператор
```

У цьому випадку більш розумно представити умову, що перевіряється, у вигляді протилежної умови:

```
if протилежна_умова then оператор
```

Наприклад, еквівалентними за одержуваним результатом є такі дві конструкції:

```
if x < x1 then else z := x1;
```

та

```
if x >= x1 then z := x1;
```

Обидва оператори (оператор_1 і оператор_2), що фігурують в операторі **if**, можуть бути будь-якими операторами, у тому числі операторами, що перевіряють умову. Тому, якщо необхідно вибрати одну з декількох взаємовиключних альтернатив, використовують вкладений оператор **if**, формат якого наступний:

```
if вираз_умова_1 then оператор_1
else
  if вираз_умова_2 then оператор_2
  else оператор_3
```

або

```
if вираз_умова_1 then
  if вираз_умова_2 then оператор_1
  else оператор_2
else оператор_3
```

Відповідність між службовими словами **then** та **else** установлюється в такий спосіб: кожному **then** відповідає найближче наступне службове слово **else**, не задіяне при встановленні відповідності з іншим **then**.

1.2. Деякі особливості запису умов

Вище відзначалося, що вираз_умова може бути будь-яким булевим виразом. Поряд з арифметичними операціями, у цьому виразі використовуються операції порівняння (або відношення) тобто операції <, >, <=, >=, =, <>, а також логічні операції **not** (логічне заперечення, НЕ), **and** (кон'юнкція, ТА), **or** (диз'юнкція, АБО), **xor** (ВИКЛЮЧАЮЧЕ АБО, операція незбігу).

Значення виразу_умови обчислюється з урахуванням пріоритетів операцій. Для перелічених операцій пріоритетні рівні наведені у табл. 1.1.

Таблиця 1.1 – Пріоритети деяких операцій

Вид операцій	Операції	Пріоритет
Унарні	–, +, **not**	1 (вищий)
Мультиплікативні	*, /, **div**, **mod**, **and**, **shl**, **shr**	2
Адитивні	+, –, **or**, **xor**	3
Відношення	=, <>, <, >, <=,	4 (нижчий)

Зауважимо, що

- унарні операції мають найвищий пріоритет серед перелічених операцій (арифметичних, порівняння й логічних) і виконуються справа наліво;
- всі інші операції з перелічених у табл. 1.1 виконуються зліва направо;
- операції відношення мають найнижчий пріоритет.

У разі необхідності зміни порядку обчислень використовуються круглі дужки.

Результат виконання логічних операцій визначається табл. 1.1.

Таблиця 1.1 – Результат виконання логічних операцій

Значення операндів		Результат			
a	b	**not** a	a **and** b	a **or** b	a **xor** b
False	False	True	False	False	False
False	True	True	False	True	True
True	False	False	False	True	True
True	True	False	True	True	False

Досить часто в складних логічних виразах виникає помилка у використанні операцій відношення. Наприклад, для перевірки приналежності деякого числа *v* інтервалу [*a*, *b*] записують такий вираз:

```
a <= v and v <= b
```

Правильною у цьому разі є така конструкція:

```
(a <= v) and (v <= b)
```

1.3. Оператор вибору

Оператор вибору (перемикач) використовується для організації розгалуження по багатьом альтернативам. Він має такий формат:

```
case селектор of
        константа_1: оператор_1;
        константа_2: оператор_2;
                  ...
        константа_N: оператор_N;
end;
```

Особливості складових частин оператора **case**:

- селектор – будь-який вираз порядкового типу (наприклад, `Integer`, `Char`, перелічений тип, але не `Real`);
- константа_1, константа_2, …, константа_N – константи, які може приймати селектор і які ідентифікують альтернативи вибору;
- константа_1, константа_2, …, константа_N можуть бути виразами над константами;
- в альтернативі вибору можна зазначити більше одного значення селектора (їх у такому разі перелічують через кому);
- замість констант у альтернативі вибору можна вказати відрізок (ряд послідовних значень) – його задають зазначеннями початку і кінця, розділеними двома крапками (наприклад: `7..27`);
- в одній альтернативі вибору можна перелічити декілька констант та відрізків з їх поділом комою;
- якщо серед альтернатив перелічені не всі можливі значення селектора, то при одержанні селектором такого значення оператор **case** ніби пропускається;
- всі значення, що не ввійшли в жодну з альтернатив, можуть бути згруповані у пункті (альтернативі) **else**; наприклад:

```
case n + 1 of
      -2: оператор_1;
      -1, 2, 5..9, 13: оператор_2;
      else оператор_else;
end;
```

- перед **else** в операторі **case** крапка з комою може ставитися, а може не ставитися;
- альтернатива **else** завжди записується останньою;
- у будь-якій з альтернатив можна записати (виконати) тільки один оператор (простий або складений).

Оператор **case** виконується в такий спосіб.

У першу чергу здійснюється обчислення значення селектора, що послідовно порівнюється з константами, зазначеними в альтернативах. При першому ж збігу селектора з однієї з констант керування передається операторові, що стоїть за відповідною константою, після чого здійснюється вихід з оператора **case**. Якщо значення селектора не збігається з жодною з констант, то виконується оператор, розташований в альтернативі **else**, а при її відсутності в не виконується жоден з операторів (рис. 1.2).

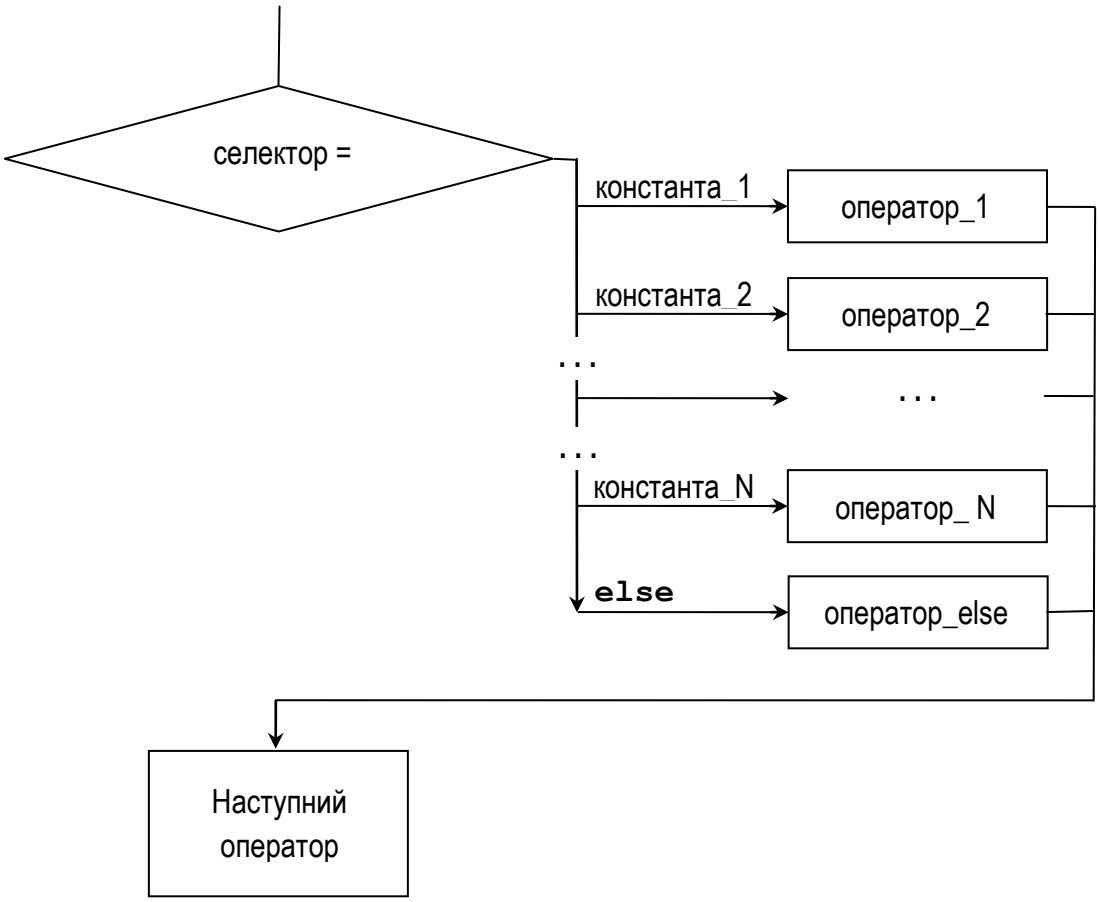


Рис. 1.2. Схема роботи оператора **case**

2. ПРИКЛАДИ ПРОГРАМ

Приклад 1. Дано дійсні числа *a*, *b*, що визначають на числовій осі інтервал (*a*, *b*). Чи правда, що випадкове число *w* ($2a \le w < 2b$) належить інтервалу (*a*, *b*)?

Розмістимо на формі два редактори Edit1 та Edit2 для введення чисел *a* та *b* відповідно, багаторядковий редактор Memo1 для виведення результату та кнопку Button1. В такому разі задачу розв'язує такий оброблювач події OnClick кнопки Button1:

```
procedure TForm1.Button1Click(Sender: TObject);
var
  a, b: Real;
  w: Real;
begin
  DecimalSeparator := '.';
  a := StrToFloat(Edit1.Text);
  b := StrToFloat(Edit2.Text);
```

```
  Randomize;
  w := 2 * Random * ( b -  a) +  a;
  if (w > a) and (w < b)  // Перевірка приналежності інтервалу
    then Memo1.Lines.Add('Yes')
    else Memo1.Lines.Add('No')
end;
```

Приклад 2. Дано ціле число k, $|k|<1000$. Перевірити для нього справедливість правила подільності на 3. **Примітка.** Ціле число ділиться на 3 тоді і тільки тоді, коли сума його цифр ділиться на 3.

Розмістимо на формі однорядковий редактор Edit1 для введення числа, багаторядковий редактор Memo1 для виведення результату та кнопку Button1. В такому разі код оброблювача події OnClick кнопки Button1 може бути таким:

```
procedure TForm1.Button1Click(Sender: Object);
vary
   n: Integer;
begin
   n := Station(Edit1.Text);
   if ( n mod 3 = 0) and           // Якщо число ділиться на 3
       ((n mod 10 +                          // Остання цифра
         n div 10 mod 10 +              // Передостання цифра
         n div 100 mod 10) mod 3 <> 0)  // Третя цифра з кінця
       or ( n mod 3 <> 0) and   // Якщо число не ділиться на 3
       ((n mod 10 + n div 10 mod 10 +
         n div 100 mod 10) mod 3 = 0)
      then Memo1.Lines.Add('Правило не виконується')
      else Memo1.Lines.Add('Правило виконується');
end;
```

Зауваження до програмної реалізації:

1. Оскільки правило подільності цілого числа на 3 виконується для всіх чисел, програма повинна завжди видавати повідомлення `Правило виконується` (у супротивному випадку має місце помилка).
2. Замість цифр розглядаються остачі від ділення числа на 10 (при цьому зберігається знак).
3. За допомогою ділення числа k на 10 та на 100 здійснюється відсікання однієї або двох останніх його цифр відповідно.
4. Якщо число має менш трьох цифр, то результат ділення на 100 (або на 10) буде дорівнювати нулю і, як слідство, сума цифр не зміниться.

Приклад 3. Дано дійсні числа x_1, y_1, R_1, x_2, y_2, R_2, що визначають відповідно координати центрів і радіуси двох кіл на координатній площині ($R_1, R_2 \geq 0$). Чи перетинаються ці кола?

Розмістимо на формі однорядкові редактори EditX1, EditY1, EditR1, EditX2, EditY2, EditR2 для введення чисел, багаторядковий редактор Memo1 для виведення результату та кнопку Button1. В такому разі код оброблювача події OnClick кнопки Button1 може бути таким:

```
procedure TForm1.Button1Click(Sender: TObject);
var
  x1, y1, R1, x2, y2, R2 : Integer;
begin
  DecimalSeparator := '.';
  x1 := StrToFloat(EditX1.Text);
  y1 := StrToFloat(EditY1.Text);
  R1 := StrToFloat(EditR1.Text);
  x2 := StrToFloat(EditX2.Text);
  y2 := StrToFloat(EditY2.Text);
  R2 := StrToFloat(EditR2.Text);
  if ((x1 - x2) * (x1 - x2) + (y1 - y2) * (y1 - y2) <=
      (R1 + R2) * (R1 + R2)) then Memo1.Lines.Add('Yes')
  else Memo1.Lines.Add('No');
end;
```

Приклад 4. Дано натуральне число, що не перевищує 10, яке задає числове значення оцінки. Вивести відповідну оцінку, позначену літерами: 10 – A, 9 – B, …, 6 – E, 5 і 4 – FX, 3, 2, 1 – F.

Розмістимо на формі однорядковий редактор Edit1 для введення числа, багаторядковий редактор Memo1 для виведення результату та кнопку Button1. В такому разі код оброблювача події OnClick кнопки Button1 може бути таким:

```
procedure TForm1.Button1Click(Sender: TObject);
var
  n: Integer;
begin
  n := StrToInt(Edit1.Text);
  Memo1.Lines.Add(IntToStr(n) + ' is equivalent');
  case n of                          // Вибір одного з варіантів
    10:    Memo1.Lines.Add('A');
     9:    Memo1.Lines.Add('B');
     8:    Memo1.Lines.Add('C');
     7:    Memo1.Lines.Add('D');
```

```
      6:    Memo1.Lines.Add('E');
      5,4:  Memo1.Lines.Add('FX');
      1..3: Memo1.Lines.Add('F');
      else  Memo1.Lines.Add('Error');
  end;
end;
```

3. ЗАВДАННЯ НА ЛАБОРАТОРНУ РОБОТУ

За час, відведений для виконання лабораторної роботи (2 академічні години), студент повинен:

1. Розробити алгоритм розв'язання задачі, запропонованої для програмування.
2. Здійснити проектування форми для функціонування розроблюваної програми.
3. Здійснити програмну реалізацію розробленого алгоритму.
4. Здійснити відлагодження програми, виправивши синтаксичні та логічні помилки.
5. Підібрати тестові дані для перевірки програми, включаючи виняткові випадки.
6. Оформити звіт до лабораторної роботи.
7. Відповісти на контрольні запитання.
8. Здати викладачу працездатну програму з демонстрацією її роботи на декількох варіантах вихідних даних.

4. ВАРІАНТИ ЗАДАЧ

1. Дано два дійсних числа. Вивести перше з них, якщо воно більше другого, і обидва числа, якщо це не так.
2. Дано три дійсних числа. Вибрати ті з них, які належать інтервалу (1, 3).
3. Дано дійсні значення *x*, *y*. Обчислити:

$$y = \begin{cases} x - y & \text{при } x > y, \\ y - x + 2 & \text{у супротивному випадку.} \end{cases}$$

4. Дано дійсні числа *a*, *b*, *c*, *d*. Якщо $a \le b \le c \le d$, то кожне число замінити найбільшим з них; якщо $a > b > c > d$, числа залишити без зміни; у супротивному випадку всі числа заміняються їхніми квадратами.
5. Якщо сума трьох попарно різних дійсних чисел *x*, *y*, *z* менше одиниці, то найменше із цих трьох чисел замінити напівсумою двох інших; у супротив-

ному випадку замінити менше з x та y напівсумою двох значень, що залишилися.

6. Дано дійсні значення x, y. Обчислити:

$$z = \begin{cases} \operatorname{arctg}\dfrac{x+y}{1-xy} & \text{при } xy < 1, \\ -\pi + \operatorname{arctg}\dfrac{x+y}{1-xy} & \text{при } xy > 1,\ x < -1, \\ -\pi + \dfrac{x+y}{1-xy} & \text{при } x < 0,\ xy > 1, \\ 1{,}57 & \text{в інших випадках.} \end{cases}$$

7. Дано дійсні додатні числа a, b, c. З'ясувати, чи існує трикутник з довжинами сторін a, b, c. Якщо трикутник існує, то відповісти на запитання – чи є він гострокутним?
8. Дано дійсні числа x_1, x_2, x_3, y_1, y_2, y_3. Чи належить початок координат трикутнику з вершинами (x_1, y_1), (x_2, y_2), (x_3, y_3)?
9. Дано дійсні числа x, y. Визначити, чи належить точка з координатами x, y заштрихованій частині площини, включаючи її межі (рис. 1.3, *а*–1.3, *л*).
10. Дано дійсні числа x, y. Визначити, чи належить точка з координатами x, y складеній з декількох незв'язаних ділянок заштрихованій частині площини, включаючи межі (рис. 1.4, *а*–1.4, *в*).
11. Дано додатні дійсні числа a, b, c, d. Чи можна прямокутник зі сторонами a і b повністю помістити усередині прямокутника зі сторонами c, d? Розв'язати задачу для випадку, коли сторони одного прямокутника можуть бути тільки паралельними або перпендикулярними сторонам іншого прямокутника;.
12. Дано дійсне число h. З'ясувати, чи має рівняння $ax^2 + bx + c = 0$ дійсні корені, якщо

$$a = \sqrt{\frac{|\sin 8h| + 17}{1 - \sin 4h \cos(h^2 + 18)^2}},$$

$$b = 1 - \sqrt{\frac{3}{3 + |\operatorname{tg} ah^2 - \sin ah|}},$$

$$c = ah^2 \sin bh + bh^3 \cos ah.$$

Якщо дійсні корені існують, то знайти їх. У супротивному випадку відповіддю повинне служити повідомлення, про відсутність дійсних коренів.

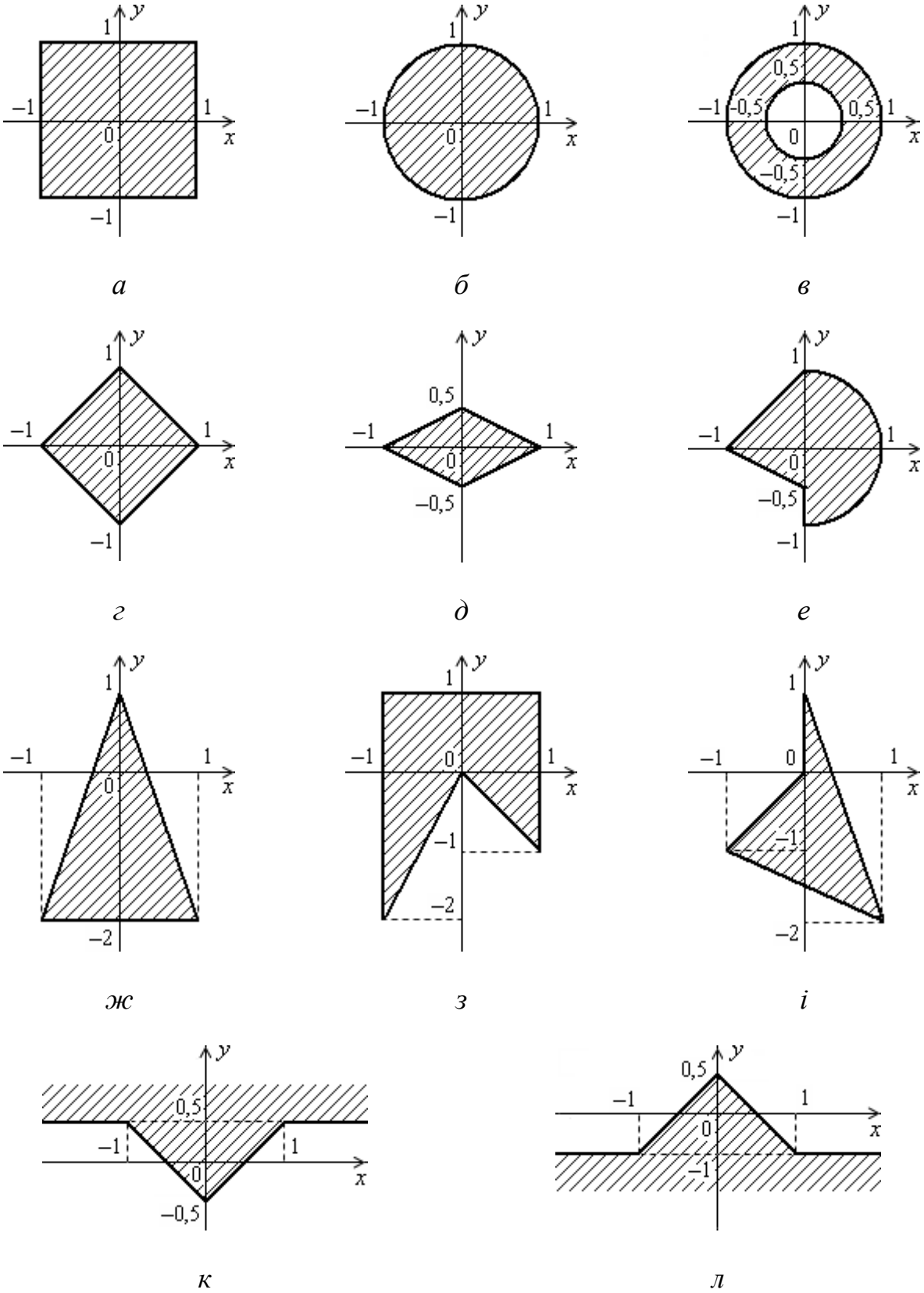


Рис. 1.3. Зв’язані області

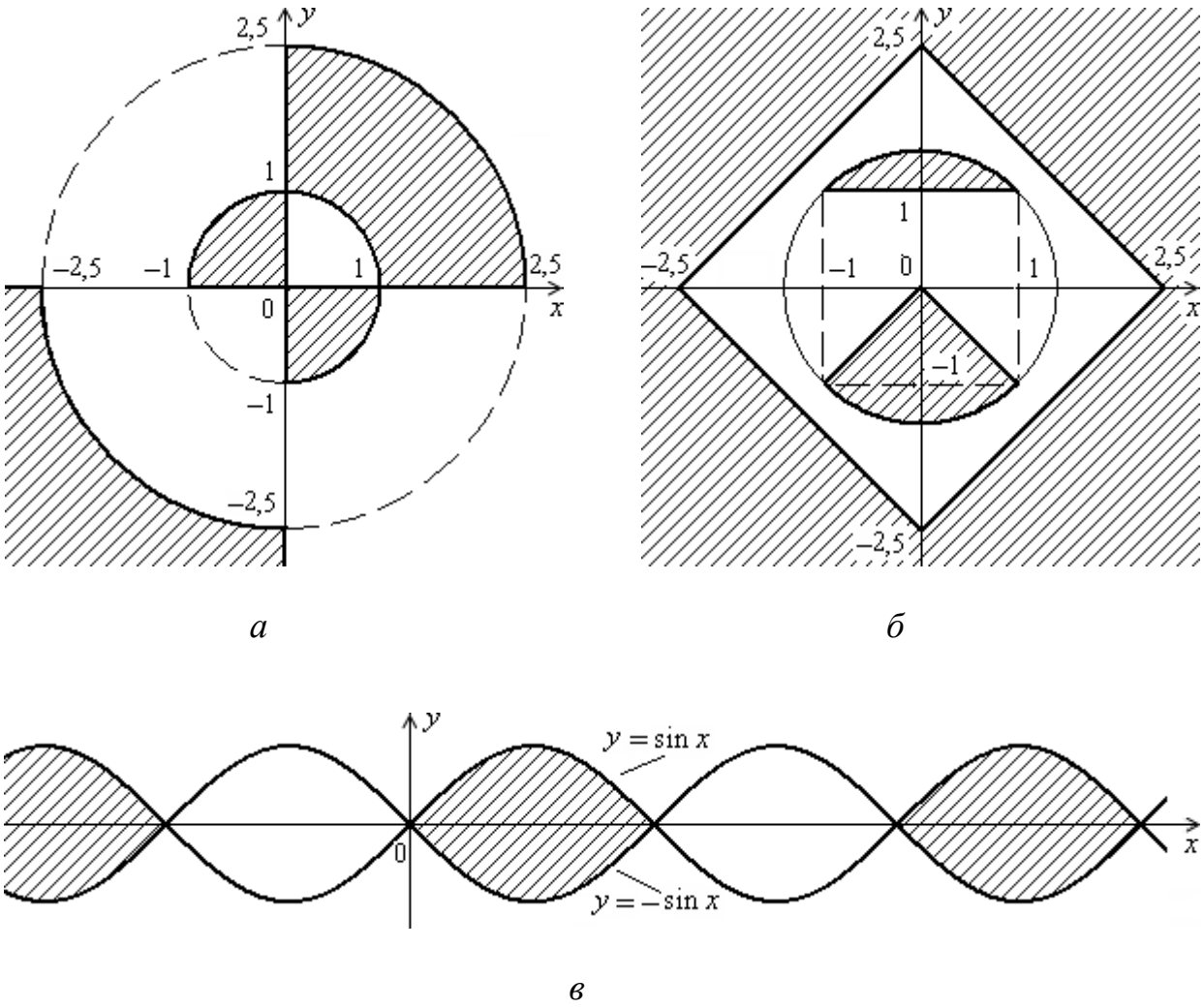


Рис. 1.3. Незв’язані області

13. Дано дійсні числа a_1, b_1, c_1, a_2, b_2, c_2, що задають коефіцієнти системи двох лінійних алгебраїчних рівнянь

$$a_1x + b_1y + c_1 = 0,$$

$$a_2x + b_2y + c_2 = 0.$$

Перевірити виконання співвідношення $|a_1b_2 - a_2b_1| \geq 0.0001$ і при виконанні цього співвідношення знайти розв’язок системи рівнянь. **Примітка.** При виконанні зазначеної умови система рівнянь гарантовано має розв’язок.

14. Дано додатні дійсні числа *a*, *b*, *c*, *d*. Чи можна прямокутник зі сторонами *a* і *b* повністю помістити усередині прямокутника зі сторонами *c*, *d*? Розв’язати задачу для загального випадку взаємного розташування прямокутників.

15. Дано додатні дійсні числа a, b, c, що визначають розміри цегли, і додатні дійсні числа d і h, які визначають розміри прямокутного отвору. Чи пройде цегла розміру $a \times b \times c$ в прямокутний отвір розміру $d \times h$? Розв'язати задачу для довільного розташування цегли.
16. Дано натуральні числа k, l, p, q. Вважаючи, що в парах чисел (k, l) і (p, q) перше число є номером рядка, а друге – номером стовпця прямокутної таблиці, визначити, чи є у комірок (k, l) і (p, q) таблиці хоча б одна спільна сторона.
17. З пункту A в пункт B із швидкістю v_0 км/год виїхав автомобіль. Одночасно назустріч йому з пункту B виїхав другий автомобіль. До їх зустрічі перший автомобіль їхав з постійною швидкістю, а другий s_1 км їхав із швидкістю v_1 км/год і s_2 км – зі швидкістю v_2 км/год. Скільки годин рухалися автомобілі до їх зустрічі і яку частку загальної відстані між пунктами A і B проїхав кожний з них?
18. Пара носків коштує 1.05 грн, в'язка з 12 пар коштує 10.25 грн, а коробка з 12 в'язками коштує 114 грн. Покупцеві потрібно n пар носків. Визначити кількість коробок $n_к$, в'язок $n_в$ і пар $n_п$ носків, які повинен купити покупець, щоб потрібна кількість пар обійшлася дешевше за все.
19. Східний календар характерний 60-річним циклом. Кожний з циклів розбитий на 12-річні підцикли, які позначаються п'ятьма кольорами – зелений, червоний, жовтий, білий і чорний. Роки кожного підциклу носять назви тварин – пацюка, корови, тигра, зайця, дракона, змії, коня, вівці, мавпи, курки, собаки та свині. Початок одного із циклів (рік зеленого пацюка) був у 1924 році. Дано ціле число, що задає деякий рік. Вивести його назву за східним календарем.

5. КОНТРОЛЬНІ ЗАПИТАННЯ

1. Які оператори застосовуються для організації розгалуження?
2. У яких форматах може бути використаний оператор **`if`**?
3. Як установлюється відповідність між службовими словами **`else`** та **`then`** в операторі розгалуження?
4. Що являє собою складений оператор?
5. Чи правильним є твердження, що в програмах, написаних мовою Delphi, в операторі **`if`** перед службовим словом **`else`** ставиться крапка з комою?
6. Як записується порожній оператор і як він використовується в операторі **`if`**?

7. Яка різниця між **if** n = 7 **then** та **if** (n = 7) **then**?
8. Коли доцільно застосовувати оператор **case**?
9. Чи є обов'язковим перелічення у альтернативах оператора **case** всіх можливих значень селектора?
10. Яка дія виконується після того, як буде виконаний оператор, що записаний у альтернативі оператора **case**?
11. Що треба зробити, щоб у альтернативі оператора **case** виконувалося декілька дій (простих операторів)?
12. Чи може виникнути ситуація, коли жоден з операторів в альтернативах оператора **case** не буде виконаний?

СПИСОК ЛІТЕРАТУРИ

1. Безменов, М. І. Турбо Паскаль 7.0 : навч. посіб. / М. І. Безменов. – Х. : НТУ «ХПІ»; Парус™, 2005. – 240 с.
2. Кэнту, М. Delphi 7 : Для профессионалов / М. Кэнту – СПб. : Питер, 2004. – 1101 с.
3. Архангельский, А. Я. Программирование в Delphi 6 / А. Я. Архангельский. – М. : БИНОМ, 2002. – 1120 с.
4. Дарахвелидзе, П. Г. Программирование в Delphi 7 / П. Г. Дарахвелидзе, Е. П. Марков. – СПб. : БХВ-Петербург, 2003. – 784 с.
5. Культин, Н. Б. Основы программирования в Delphi 7 / Н. Б. Культин. – СПб.: БХВ-Петербург, 2003. – 608 с.
6. Пестриков, В. М. Delphi на примерах / В. М. Пестриков, А. Н. Маслобоев. – СПб. : БХВ-Петербург, 2005. – 496 с.
7. Ремкеев, А. А. Курс Delphi для начинающих. Полигон нестандартных задач / А. А. Ремкеев, С. В. Федотова. – М. : СОЛОН-Пресс, 2006. – 360 с.
8. Митчелл, К. Керман. Программирование и отладка в Delphi™ : учебный курс / Митчелл К. Керман. – М. : Вильямс, 2004. – 720 с.
9. Парижский, С. М. Delphi : Только практика / С. М. Парижский. – К. : МК-Пресс, 2005. – 208 с.
10. Культин, Н. Б. Основы программирования в Delphi 2007 / Н. Б. Культин. – СПб. : БХВ-Петербург, 2008. – 480 с.

Навчальне видання

Методичні вказівки
до лабораторної роботи
«Організація галуження у програмах мовою Delphi»
з курсу «Програмування» для студентів напряму 6.040302 – Інформатика
(спеціалізація «Соціальна інформатика»)

Укладач БЕЗМЕНОВ Микола Іванович

Відповідальний за випуск О. С. Куценко
Роботу до видання рекомендував О. В. Горелий

За авторською редакцією

План 2010 р., поз. 12 / 42-10

Підп. до друку 15.03.2010 р. Формат 60 × 84 $^{1}/_{16}$. Папір офісний.
Riso-друк. Гарнітура Таймс. Ум. друк. арк. 0,9. Наклад 50 прим.
Зам. № 63. Ціна договірна.

Видавничий центр НТУ «ХПІ».
Свідоцтво про державну реєстрацію ДК № 3657 від 24.12.2009 р.
61002, Харків, вул. Фрунзе, 21

Друкарня НТУ «ХПІ», 61002, Харків, вул. Фрунзе, 21