Алгоритмічна конструкція розгалуження. Умовні оператори в мові *Delphi: If…Then, If…Then…Else.* Оператор вибору *Case.* Виконання програм з розгалуженнями в покроковому режимі. Вкладені оператори розгалуження

Алгоритмічні конструкції одно-, дво- і багатоальтернативних розгалужень

У низці випадків одноальтернативие розгалуження («робити — не робити») може виявитися недостатнім: в алгоритмі може бути поставлене завдання вибору однієї з двох альтернатив. Подібні алгоритмічні структури реалізуються двоальтернативним оператором **IF**, що має таку структуру:

```
if   логічний вираз then
   оператор
else
   оператор;
```

Тут логічний вираз — це булева змінна, константа або логічний вираз, який являє собою операцію або кілька операцій порівняння, об'єднаних операторами булевої алгебри.

Оператори, що стоять після ключових слів **THEN** і **ELSE**, можуть бути простими операторами або складеними (тобто групою операторів, об'єднаною операторними дужками begin і end).

Як приклад наведемо фрагмент програми, що обчислює корінь квадратного рівняння після перевірки значення дискримінанта (змінні *a, b, c* — коефіцієнти квадратного рівняння, x_1 і x_2 — його корені; усі використані змінні дійсного типу, вони описані на початку процедури або в розділі інтерфейсних змінних).

```
if    b*b - 4*a*c >=0  then
begin
    Label1.caption:=FloattoStr((-b+ sqrt(b*b - 4*a*c))/2/a);
    Label2.caption:=FloattoStr((-b+ sqrt(b*b - 4*a*c))/2/a);
end
else
begin
    Label1.caption := 'Немає дійсних коренів';
    Label2.caption := '  ';
 end;
```

Іноді в алгоритмах недостатньо й двох альтернатив, тоді використовується оператор багатоальтернативного розгалуження **CASE**, який має таку структуру:

```
case перемикач of
    значення 1 перемикача: оператор;
    значення 2 перемикача: оператор;
    … .
end;
```

або

```
case перемикач of
    значення 1 перемикача: оператор;
    значення 2 перемикача: оператор;
    … .
else
   оператор
end;
```

Вираз-перемикач у цьому операторі може бути змінною або виразом перелічного (порядкового) типу, наприклад цілого, літерного. Тип перемикача має збігатися з типом його значень, які стоять перед двокрапкою в альтернативах оператора **CASE**. Частина **ELSE** не обов'язкова — вона вписується в оператор у випадку, коли значення виразу-перемикача не збігається з жодним із зарезервованих значень. Таким чином, якщо вираз-перемикач не дорівнює жодному з цих значень і є частина **ELSE**, то виконується оператор, що йде за **ELSE**. Якщо ж частину **ELSE** в оператор **CASE** не включено, то **CASE** не виконує ніяких дій.

Оператори, що стоять після двокрапки в альтернативах **CASE** (так само як і в операторі **IF**), можуть бути простими або складеними (тобто групою операторів, об'єднаних операторними дужками **begin** і **end**).

Розглянемо простий приклад. На формі містяться кнопка, компонент для введення тексту **Editl** і рядок виведення **Label1**. Натисканням кнопки потрібно ввести номер місяця невисокосного року й вивести в **Label1** кількість днів у цьому місяці.

Метод натискання кнопки матиме такий вигляд:

```
procedure TForm1.Button1Click(Sender: TObject);
var
   mesyac: integer;
begin
   mesyac:=strtoint(edit1.Text);
   case msyac of
     1,3,5,7,8,10,12: label1.caption:= '31';
     4,6,9,11: label1.caption:='30';
     2:label1.caption:='28';
   else
     label1.caption:='Введено неможливий номер місяця'
   end;
end;
```

У цьому прикладі певним значенням змінної-перемикача **misyac** відповідають однакові дії. Щоб не повторювати однакові оператори при різних альтернативах, у деяких рядках значення перемикача просто перераховані через кому. Замість зазначених операторів можуть стояти інші оператори розгалуження. Наприклад,

модифікуємо наведений вище фрагмент для будь-якого року — як високосного, так і невисокосного. Нагадаємо, що високосним є рік, номер якого без остачі ділиться на 4 і не має в числовому записі двох останніх нулів.

Тепер на формі в нас з'явиться ще один компонент для введення тексту **Edit2** (для введення номера року), а метод обробника кнопки зміниться так:

```
procedure TForm1.Button1Click(Sender: TObject);
var
   mesyac, god: integer;
begin
   mesyac:=strtoint(edit1.Text);
   god:=strtoint(edit2.Text);
   case mesyac of
       1,3,5,7,8,10,12: label1.caption: = '31';
       4,6,9,11: label1.caption:='30';
       2: if (god mod 4 o 0) or (god mod 100 = 0) then
            label1.caption:='28'
          else
            label1.caption:='29';
    else
       label1.caption:='Введено неможливий номер місяця'
   end;
end;
```

ТЕОРЕТИЧНІ ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

1. Якими є особливості використання оператора IF?
2. Як вставити в оператор IF більш ніж один оператор?
3. Яке призначення має оператор багатоальтернативного розгалуження CASE? У яких випадках його використовують?
4. Нарисуйте структурну схему для реалізації трьох альтернатив.
5. Наведіть приклади використання чотирьох альтернатив за допомогою операторів IF і CASE.

Виконання програм із розгалуженнями в покроковому режимі. Вкладені оператори розгалуження

Розглянемо виконання умовних операторів у програмуванні тесту на фреймах.

Приклад 6. Створюємо основну форму і два фрейми (створення фрейму **New → Frame**) — їх наведено на рис. 8.4.

Виносимо на форму компонент Frames, у діалоговому вікні обираємо Frame2 і Frame3 (у них наскрізна нумерація з формами, тому якби в нас були дві форми, то фрейми одержали б номери 3 і 4). Потрапивши на форму, фрейми змінюють імена: вони стають Frame21 і Frame31 (тобто 2-й фрейм на 1-й формі і 3-й фрейм на 1-й формі).

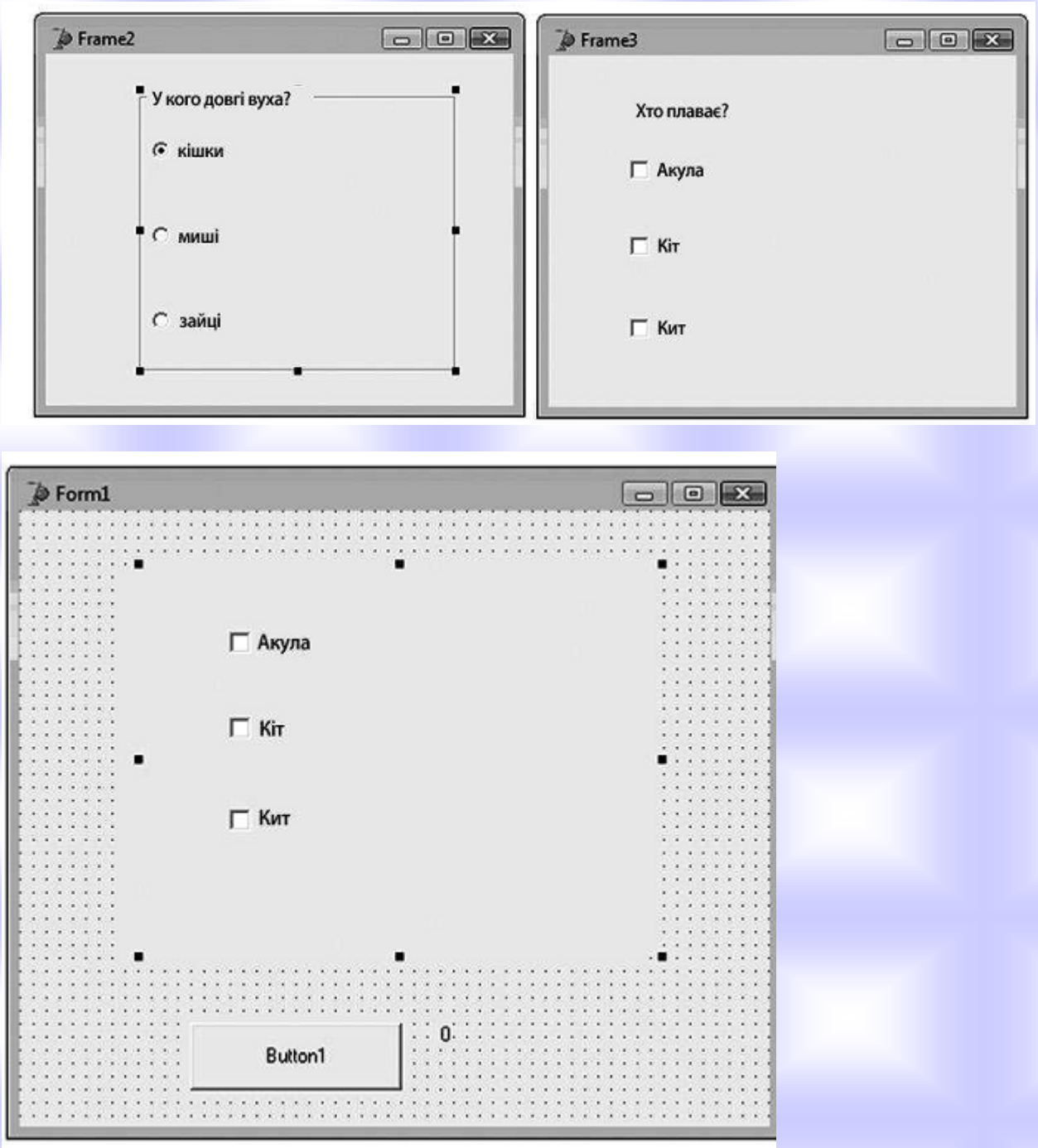

Рис. 8.4. Два фрейми та основна форма з фреймами

Тепер програмуємо тест у методі натискання кнопки:

```
var
    Forml: TForm1; n: integer=0;
implementation
{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
begin
  if frame21.Visible then
    begin
      if frame21.radiogroup1.itemindex=2 then
        n:=n+l;
```

```
        frame21.Visible:=false;
        frame31.Visible:=true;
       end
       else
      begin
         if frame31.checkbox1.checked and
frame31.checkbox3.checked then
           n:=n+l;
         frame31.Visible:=false;
       end;
       Label1.Caption:=inttostr(n);
end;
```

Необхідно звернути увагу на присвоювання значення типізованій константі. Для того щоб це було дозволено, потрібно поставити «галочку» в настройках **Project →Options →Assignable typed constant**. Ще одна особливість програми — компоненти фреймів із форми «не видно», їх слід іменувати на повне ім'я, наприклад: **frame21.radiogroupl.itemindex**.

Розглянемо приклад використання фреймів та умовних операторів.

Приклад 7. Готуємо фрейм, поміщуємо в нього Image. Виносимо на форму компонент frames, вставляємо в нього підготований фрейм (рис. 8.5). Оскільки він менший за форму, автоматично з'являються скролери (властивість **AutoScroll** можна вимкнути — **false**).

Зображення у віконці можна переміщувати програмно. Наприклад, розташуємо на формі кнопки «Ліворуч», «Праворуч», «Угору» й «Униз». Так, кнопка «Ліворуч» зсуває рисунок на 5 пікселів:

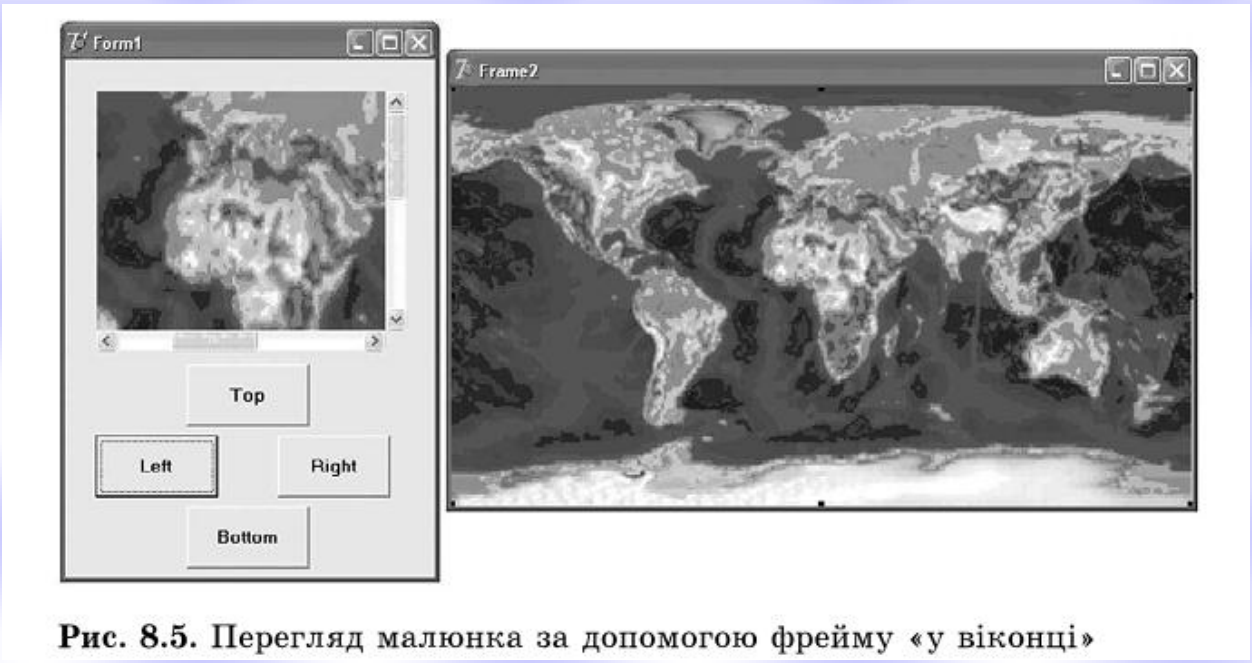

Рис. 8.5. Перегляд малюнка за допомогою фрейму «у віконці»

```
procedure TForm1.Button1Click(Sender: TObject);
begin
   frame21.imagel.Left:=frame21.imagel.Left-5;
end;
```

Використовуючи подібний підхід, можна реалізувати переміщення і в інших напрямках. Така програма має певні недоліки: натискаючи кнопки,

можна «засунути» рисунок за край вікна. Це можна виправити кількома умовними операторами, встановлюючи «неприступність» кнопок. Наприклад, під час руху малюнка праворуч виконується така програма:

```
frame21.imagel.Left:=frame21.imagel.Left+5;
if frame21.imagel.Left>0 then
    Button2.Enabled:=false;
Button1.Enabled:=true;
```

Однак тепер не забудьте «ввімкнути» потрібну кнопку, коли рисунок відсувається від краю,— «ліва» кнопка вмикає «праву», і навпаки. У наведеному фрагменті програми кнопка **Button2** робить доступною **Button1** (рядок **Buttonl.Enabled:=true**;).

Повні умови зміщення малюнка в трьох інших кнопках вам належить записати самостійно (відмінності там будуть у властивостях top і left та в напрямках руху).

ТЕОРЕТИЧНІ ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

1. У чому полягає особливість використання frame?
2. Які дії необхідно виконати для можливості присвоїти значення типізованій константі?
3. Як змістити рисунок праворуч на 5 пікселів?
4. Як змістити рисунок угору на 7 пікселів?