

Тема: Атрибути, змінні, присвоювання та стандартні методи.

Мета: Познайомитися із типами даних, описом змінних та власними типами даних, навчитись використовувати результати підрахунків для зміни властивостей відеокомпонентів для отримання візуальних ефектів від реалізації заданого алгоритму.

1. Поняття змінної та константи. Поняття ідентифікатора. Атрибути форм та елементів управління в програмі.

Поняття змінної та константи.

Ідея позначати проміжні та кінцеві результати підрахунків літерами, а потім використовувати їх в ролі змінних виникла в умах математиків ще у древні часи. Очевидно, що у перших мовах програмування, які в основному використовувались для прискорення формульних обрахунків, механізм змінних та констант був повністю запозичений (у подальшому розвинений) із математики.

Як і в інших мовах програмування, в мові Pascal (та в її послідовниці Delphi) дані поділяються на константи та змінні. В програмі вони визначаються ідентифікаторами (іменами), за якими до них можна звертатися для отримання поточних значень та вводити їх до складу арифметичних, логічних чи рядкових виразів.

Константами називаються елементи даних, значення яких встановлені в описовій частині програми і в подальшому не змінюються.

Наприклад, якщо у програмі використовується значення кількості учнів у класі, то краще за все його задати константою. При визначенні середнього балу успішності учнів класу ця кількість не змінює свого значення. Якщо в клас прибуде один учень, достатньо збільшити константу на одиницю, і всі розрахунки будуть відбуватися з новим числовим значенням. Це значно простіше, ніж переглядати всю програму, змінюючи кількість, задану числом, на нове значення.

Константи описуються у спеціальному розділі, який розпочинається із зарезервованого слова `const`, за таким форматом:

`Const`

`<ідентифікатор> = <значення константи>;`

Наприклад:

`Const`

`MySchool = 'Солонянська СЗШ №1';`

`MinTemp = -273;`

`Inch = 2.54;`

Тип користувацьких констант розпізнається компілятором без попереднього опису.

В Pascal є ряд констант, до значень яких можна звертатись без попереднього визначення, наприклад, `Pi`. Їх називають зарезервованими константами.

Змінними називаються величини, які можуть змінювати своє значення в процесі виконання програми. Кожна змінна, як і константа, належить до певного типу даних.

Тип даних повинен бути заздалегідь відомим компілятору до початку трансляції виразів і операторів із використанням цих змінних. Окрім того, за

відомим типом даних компілятор виділяє під змінні в оперативній пам'яті необхідне місце – різну кількість байтів для різних типів даних.

Змінні описуються у спеціальному розділі, який починається із зарезервованого слова `var`, за таким форматом:

```
Var  
    <ідентифікатор> : <тип>;
```

Наприклад:

```
Var  
    a,b,c: integer;  
    x:     real;  
    s:     string;
```

Окрім констант та змінних, існують так звані типізовані константи, значення яких можуть змінюватись у ході виконання програми. Для них слово «константа» лише вказує на те, що дані цього типу описуються у розділі `const`, а слово «типізована» - на те, що цей тип даних повинен позначатися, як у змінних.

Формат опису типізованих констант може бути реалізований в одному із наступних варіантів:

```
Const  
    <ідентифікатор> : <тип> = <значення>;
```

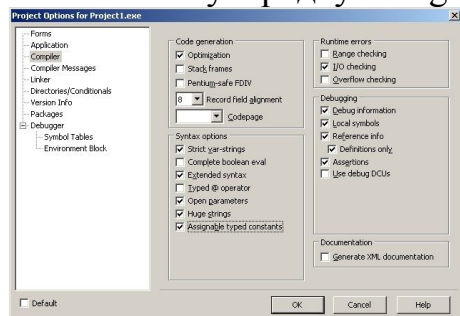
Або:

```
Var  
    <ідентифікатор> : <тип> = <значення>;
```

Наприклад:

```
Var  
    R: real = 57.29;  
Const  
    Language: string = 'Delphi';
```

У процесі програмування типізовану константу можна використовувати як змінну із заздалегідь визначеним значенням, але для цього потрібно виконати одне налаштування в інтерфейсі Delphi. Заходимо через головне меню в пункт `Project – Options`. У закладці `Compiler` для дій із типізованими константами потрібно поставити галочку в рядку `Assignable typed constants`:



Поняття ідентифікатора.

Для позначення змінних та постійних величин, різноманітних процедур та функцій використовуються ідентифікатори – імена, які встановлюють відповідність між об'єктом та деяким набором символів (латинських літер, знаків підкреслювання та цифр; цифра не може бути першим символом ідентифікатора).

Можливості програміста при виборі імен для своїх змінних досить широкі. Не можна тільки використовувати в ролі ідентифікаторів зарезервовані слова, які компілятор сприймає як базові директиви чи оператори програми (наприклад, не можна назвати змінну словами `begin` чи `end`); не бажано використовувати також і стандартні ідентифікатори.

Стандартні ідентифікатори введені авторами мови програмування для позначення типів даних, процедур чи функцій, наприклад: `integer`, `sin`, `read`, `writeln` тощо.

Ідентифікатори користувача вводяться програмістом для позначення констант, змінних, процедур чи функцій користувача у програмі. При цьому ідентифікатори повинні бути унікальними: в кожному програмному блоці один і той же ідентифікатор не може бути використаний для позначення більш ніж однієї змінної, константи, функції чи процедури. Дублювання ідентифікаторів викликає помилку на етапі компіляції програми.

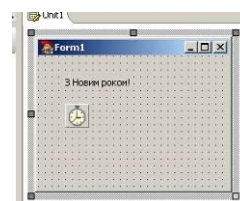
В ідентифікатор на мові програмування Delphi не можуть входити пробіли, спеціальні символи та літери будь-який алфавітів, за винятком англійського (латинського).

Правильно підібрані ідентифікатори значно полегшують читання та розуміння програми, а також зменшують ймовірність появи помилок при внесенні до неї змін.

Атрибути форм та елементів управління у програмі.

При візуальному проектуванні форм програміст виносить на поле форми необхідні йому відеокомпоненти. Робиться це простим перетягуванням необхідного відеокомпоненту із інструментальної лінійки на форму. Вид відеокомпоненту можна змінювати, змінюючи його властивості через інспектор об'єктів. Властивості форм і відеокомпонентів (або, як ще кажуть, їхні атрибути), по суті, є змінними і в такій якості вони можуть бути використані у програмі. Їх основна відмінність від змінних користувача – нерозривний зв'язок із об'єктом, якому вони належать, і розроблений авторами Delphi метод, який визначає, що зміниться у поведінці об'єкта при зміні атрибуту.

Наприклад, на формі розміщені два компоненти – напис (`label`) та таймер (`timer`). Запишемо обробник події форми `OnMouseDown` (цей фрагмент програми буде виконуватись при натискуванні лівої кнопки мишки у той час, коли вона перебуває над формою):



```
procedure TForm1.FormMouseDown(Sender: TObject; Button: TMouseButton; Shift: TShiftState;
X, Y: Integer);
var
  inter: integer;
  capt: string;
begin
  timer1.Interval := 100;
  label1.Caption := 'З Новим роком!';
  inter := 2000;
  capt := 'Привіт!';
end;
```

У наведеній процедурі виконуються чотири операції присвоювання. Й якщо зміна властивостей компонентів `timer1` та `label1` очевидні (починають швидше відбуватись події таймера, змінюється напис на кнопці), то зміна значень змінних ні до яких видимих наслідків не призводить. У цьому й полягає відмінність змінних від властивостей відеокомпонентів. Змінні застосовуються лише для розрахунків і вимагають написання процедур для їх виведення чи використання у програмі (для зміни графічного зображення, мультиплікації тощо). Для властивостей же відеокомпонентів режим їх використання передбачений власне у самому середовищі Delphi. Досить змінити ці властивості, щоб отримати видимий результат.

2. Типи даних. Оператор присвоювання.

Типи даних.

Опис типу даних вказує компілятору на те, як зберігати та опрацьовувати інформацію у програмі, які процедури та функції можуть бути застосовані для даного типу.

В Delphi в оголошенні змінної необхідно вказувати її тип. Типи даних поділяються на стандартні та створювані користувачем. Стандартні типи уже визначені у мові програмування, а змінну такого типу можна оголосити, просто вказавши ім'я цього типу. Створювані типи програмісту доводиться конструювати власноруч у відповідності із призначенням програми.

Типи даних, визначені користувачем, задаються:

- В розділі опису типів програми.
- В модулі (unit), у розділі `interface` або `implementation`.
- У зовнішньому модулі, який підключається (unit).
- Всередині процедур.
- Всередині функцій.

Оголошення типів діють в межах того блоку, в якому вони розміщені. Поза цим блоком посилатись на ці типи не можна. Всередині ж вони замінюють (перекривають) зовнішні типи з тими самими іменами. Оголошені типи даних можна застосовувати в межах області їх видимості.

Оголошення типів є для компілятора правилами, які він повинен «запам'ятати» на той випадок, якщо зустрине у програмі посилання на той чи інший тип. Саме по собі оголошення типу не приносить до програми ніяких змін. Тип змінної обмежує її значення та ті операції, які можна виконувати з цими значеннями.

Оголошення типів і змінних схематично можна уявити наступним чином:

```
type1 = type_definition1; // кожному новому типу присвоюється ім'я, потім він
// визначається через уже існуючі типи.
type2 = type_definition2; // В одному розділі type можна оголосити декілька
// типів. Найбільш просте визначення типу складається із імені типу,
// оголошеного раніше
type3 = type1;
Нові змінні оголошуються у var:
Var
Var1 :type_definitions;
```

// Кожній новій змінній присвоюється спочатку ім'я, а потім – тип.

Var2, var3: type_definition4;

Var4: type1;

В одному розділі var можна оголошувати декілька змінних. Їм можна присвоювати один і той же тип. При цьому тип може бути сконструйований заздалегідь або ж побудований безпосередньо у var.

До основних типів даних відносяться:

- Прості.
- Рядкові – для збереження послідовності символів.
- Структуровані.
- Вказівники – для посилань на змінні заданих типів.
- Процедурні – для звернення до процедур та функцій як до змінних.
- Варіантні – для збереження в одній змінній даних різних типів.

Як правило, ідентифікатори типів (виділені для назви змінні) використовуються тільки визначення нових типів або ж оголошення змінних. Допускається використання імені типу як функції для перетворення типів.

Багато із визначених типів мають дуже складні структуру і можуть займати в пам'яті багато місця. При цьому елементи таких типів створені скоріш за все для представлення значень у певному логічному порядку, ніж для того, щоб зменшити місце, яке виділяється у пам'яті.

Структура типів у Delphi:

- Simple (прості):
 - Ordinal (порядкові типи)
 - Integer (цілий)
 - Character (літеральні типи)
 - Boolean (логічний)
 - Enumerated (перераховуваний)
 - Subrange (діапазон)
 - Real (дійсний)
 - Tdatetime (дата – час)
- Structured (структуровані, складені):
 - Array (масив)
 - String (рядковий тип)
 - List та stringlist (списки)
 - Set (множини)
 - Record (записи)
 - File (файлові типи)
 - Textfile (текстові файли)
 - Class (класи)
 - Class reference (посилання на класи)
 - Interface (інтерфейси)
 - Pointer (вказівники)
 - Variant (варіанти)
 - Procedural (процедурний)

○ Type identifier

До цього переліку не внесені типи, які підтримуються у Delphi, проте є для цієї мови «зовнішніми», запозиченими із інших програм та додатків. Наприклад, тип `bitmap`, призначений для зберігання малюнків, а також типи, пов'язані із датою та часом `DateTime`. Багато типів даних (`TTable`, `TQuery` тощо) введено для підтримки баз даних. Взагалі, типів даних дуже багато, і з кожною новою версією Delphi їх кількість зростає. Знати всі типи даних для програмування в цьому середовищі зовсім не обов'язково. Потрібно лише добре орієнтуватися тільки в основних типах.

Оператор присвоювання.

Присвоювання – механізм у програмування, який дозволяє змінювати значення змінних або властивостей об'єктів шляхом заміни старих значень новими. Загальний синтаксис операції присвоювання:

<вираз зліва> := <вираз справа>;

У ролі виразу зліва можуть використовуватися змінна, типізована константа або властивість об'єкту.

Вираз справа повинен позначати величину, яка буде присвоєна об'єкту даних. Це може бути змінна, типізована константа, значення властивості об'єкта або ж допустимий вираз (арифметичний, логічний, рядковий) на їх основі.

В ролі оператора присвоювання у Delphi виступає знак «:=». За зовнішнім виглядом він нагадує математичний знак «=», що може призвести до помилкового висновку, нібито оператор присвоювання є його точна копія. У такому випадку запис $x=x+1$ з математичної точки зору є повною нісенітницею; проте в програмування це широко застосовуваний оператор $x:=x+1$.

Істотною відмінністю оператора присвоювання є те, що справа від знаку присвоювання знаходиться «старі» значення змінної x , над яким буде виконана певна послідовність дій. Результат цих дій (результат виконання математичних операцій) необхідно занести в ту ж комірку x . Наприклад, якщо в програмі до цього оператора в змінній x зберігалось значення 5, то після виконання оператора в ній буде значення 6.

3. Теоретичні питання для самоконтролю.

1. Дайте визначення поняттю «константа». В чому полягає особливість її використання?
2. Що являють собою змінні? Наведіть приклади їх опису.
3. Дайте визначення поняттю «ідентифікатор».
4. Чим відрізняються стандартні та користувацькі ідентифікатори?
5. В чому полягає відмінність змінних від властивостей об'єктів?
6. Типи даних – призначення та місце їх опису.
7. Перерахуйте типи даних, які відносяться до простих типів.
8. Перерахуйте типи даних, які відносяться до структурованих (складених) типів.
9. Для чого призначена операція присвоювання?