

УДК 004.42

М.А. Бондаренко, В.А. Жилін, Д.П. Панасенко

Українська інженерно-педагогічна академія, Харків

ТЕСТУВАННЯ КЛАСІВ І ПРИКЛАД ЇХ ВИКОРИСТАННЯ У КОМПІЛЯТОРНИЙ ПІДСИСТЕМІ СИНТАКСИЧНОГО АНАЛІЗУ АРИФМЕТИЧНИХ ВИРАЗІВ

В роботі наведено приклад використання нових класів у системі об'єктно-орієнтованого програмування, що можуть бути застосовані при розробці підсистеми синтаксичного аналізу арифметичних виразів. Розробка прикладу здійснена засобами системи об'єктно-орієнтованого програмування Delphi 6.

Ключові слова: класи даних, об'єктно-орієнтоване програмування, підсистеми синтаксичного аналізу.

Вступ

Розвиток комп'ютерних технологій програмування привів до створення мов програмування високого рівня, що є системами, які позбавляють програміста необхідності опрацьовувати програмні додатки у машинних кодах або асемблері. Проте природно виникла потреба створення компіляторів, що перетворюють програмні коди мов високого рівня у машинні коди.

Зазвичай компілятор складається з декількох підсистем, зокрема, підсистеми лексичного аналізу і підсистеми синтаксичного аналізу. Задача цих підсистем – аналіз тексту програми, пошук помилок на етапі компіляції, підготовка до генерації машинного коду.

Прикладний додаток, який було розроблено [1], є моделлю справжньої підсистеми синтаксичного аналізу. Він відображає основні функції синтаксичного аналізу арифметичних виразів. Розробку класів здійснено засобами системи об'єктно-орієнтованого програмування Delphi 6. [2 – 4]

Метою даної статті є розробка прикладу використання нових класів у системі об'єктно-орієнтованого програмування, що можуть бути застосовані при розробці підсистеми синтаксичного аналізу арифметичних виразів.

Отримані результати повинні підтвердити правильність розробленої програмної реалізації [1] запропонованих класів.

Розробка моделі підсистеми синтаксичного аналізу арифметичних виразів

Для розробки повної системи синтаксичного аналізу потрібно узгодження з іншими підсистемами.

На рис. 1 наведено вікно додатку.

У верхній частині вікна розташовано текстове поле Edit1, в яке будуть введені арифметичні вирази. Командна кнопка «Новий вираз» призначена для очистки полів вводу та поля відображення результату аналізу.

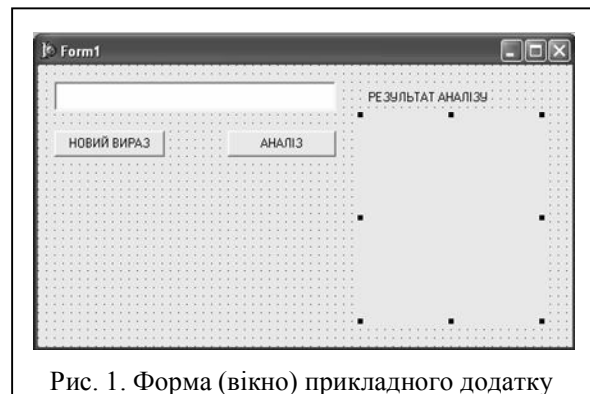


Рис. 1. Форма (вікно) прикладного додатку

Командна кнопка «Аналіз» призначена для запуску процедури синтаксичного аналізу арифметичного виразу.

Поле (мітка Label1) під надписом «Результат аналізу» призначена для відображення результату аналізу.

Програмний код додатку наведено нижче.

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants,
  Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ClassAV, ClassTempTable;
type
  TForm1 = class(TForm)
    Edit1: TEdit;
    Button1: TButton;
    Label1: TLabel;
    Button2: TButton;
    Label2: TLabel;
  procedure Button1Click(Sender: TObject);
  procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
  AV:TAV;
```

```

T:TTempTable;
implementation
{$R *.dfm}
// Аналіз та перетворення виразу
procedure TForm1.Button1Click(Sender: TOb-
ject);
var
i,k,r,x:integer;
s:string;
begin
AV:=TAV.Create; // Утворення об'єкту класу
"Арифметичний вираз"
T:=TTempTable.Create; // Утворення об'єкту
класу "Допоміжня таблиця"
AV.Content:=edit1.Text;
// Цикл перевірки та спрощення виразу
repeat
x:=AV.Analyze; // Перевірка виразу
if x<0 then
begin
Showmessage('ПОМИЛКА У
АРИФМЕТИЧНОМУ ВИРАЗІ');
exit; end;
AV.Simplify(T); // Спрощення виразу
until x=0;
// Відображення отриманих операцій
for i:=1 to T.Count do
begin
s:=T.Get(i);
for k:=1 to length(s) do
begin
if ord(s[k])<127 then La-
bel1.Caption:=Label1.Caption+s[k]
else begin
// Допоміжня змінна
r:= ord(s[k])-ord('A');
La-
bel1.Caption:=Label1.Caption+'R'+IntToStr(r);
end; end;
Label1.Caption:=Label1.Caption+#13;
end;
s:=AV.Content;
for k:=1 to length(s) do
begin
if ord(s[k])<127 then La-
bel1.Caption:=Label1.Caption+s[k]
else begin
// Допоміжня змінна
r:= ord(s[k])-ord('A');
La-
bel1.Caption:=Label1.Caption+'R'+IntToStr(r);
end; end; end;
// Новий вираз
procedure TForm1.Button2Click(Sender: TOb-
ject);
begin
edit1.Clear;
label1.Caption="";
end; end.

```

Тестування властивостей та методів класів

Тестування властивостей та методів розробле-них класів проводилося в декілька етапів. Нагадає-мо, що синтаксичний аналіз здійснюється після лек-сичної обробки арифметичного виразу. Тобто у ви-разі вилучені пропуски, а кожна змінна або констан-та замінена на відповідну лексему. В нашому при-кладі замість лексем ми будемо використовувати латинські букви.

1. На першому етапі тестування перевірено ре-акцію підсистеми на помилку в арифметичному ви-разі, тобто тестування методу Analyze.

Після запуску додатку було введено помилко-вий вираз, у якому кількість закриваючих дужок більша за кількість відкриваючих (рис. 2).



Рис. 2. Введення помилкового виразу

Далі було натиснуто кнопку «Аналіз». Отрима-но відповідне повідомлення (рис. 3).

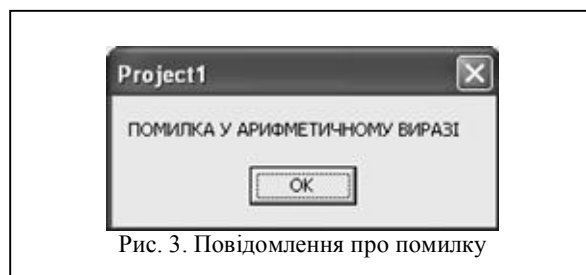


Рис. 3. Повідомлення про помилку

Наступні перевірки виразів з помилковим ба-лансом дужок також показали, що метод Analyze виявляє помилку.

Далі було перевірено пари операцій.

Нагадаємо, що у арифметичному виразі не до-пускається, щоб дві операції були поруч (тобто ++, */ і т. п.).

Для цієї перевірки було натиснуто кнопку «Но-вий вираз» і введено помилковий вираз (рис. 4).

Після натискання кнопки «Аналіз» також було отримано повідомлення про помилку (рис. 4). На-ступні перевірки виразів з помилковими парами операцій також показали, що метод Analyze виявляє цю помилку.

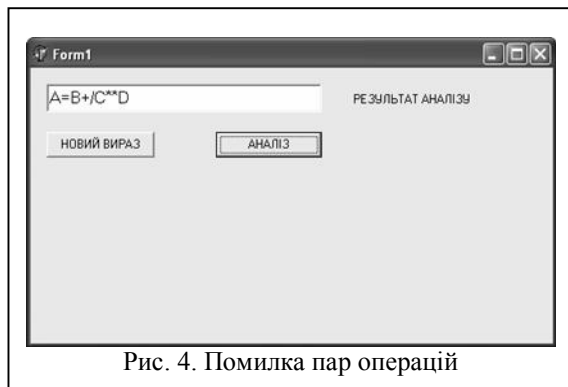


Рис. 4. Помилка пар операцій

Таким чином, робимо висновок, що метод Analyze класу «Арифметичний вираз» працює у відповідності до вимог.

2. На другому етапі тестування перевірено роботу методу Simplify (спрощення виразу) для виразу без дужок, але з операціями з різними пріоритетами. Було введено такий арифметичний вираз (рис. 5).

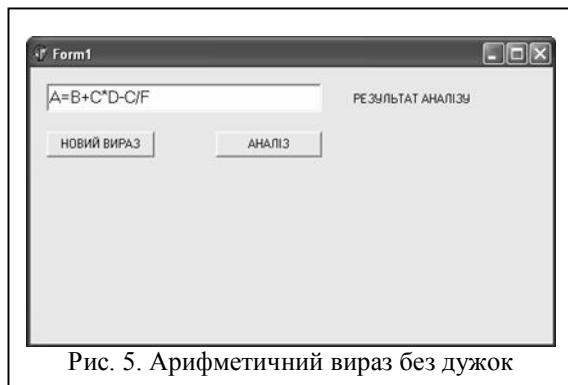


Рис. 5. Арифметичний вираз без дужок

Після натискання кнопки «Аналіз» отримано такий результат (рис. 6).

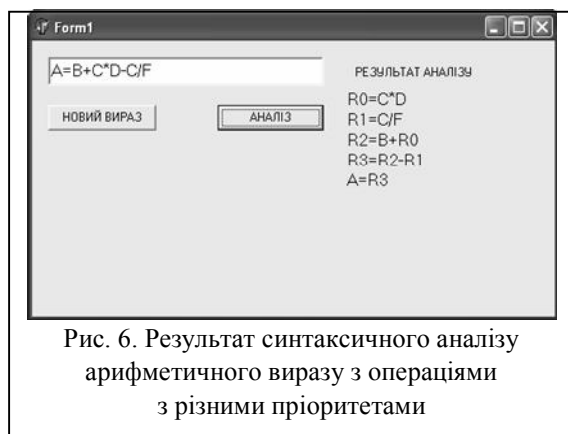


Рис. 6. Результат синтаксичного аналізу арифметичного виразу з операціями з різними пріоритетами

Як бачимо, результат відповідає очікуванням. Спочатку виконані операції з більш високим пріоритетом (множення та ділення), а потім – додавання і віднімання.

Наступні перевірки також показали правильність синтаксичного аналізу для таких виразів (рис. 7).



Рис. 7. Результат синтаксичного аналізу арифметичного виразу без дужок

Таким чином, можна зробити висновок про те, що метод Simplify (спрощення виразу) класу «Арифметичний вираз» для виразу без дужок працює відповідно до вимог.

3. На третьому етапі було перевірено роботу методу Simplify для виразів з дужками. Було отримано такий результат, який наведено на рис. 8.

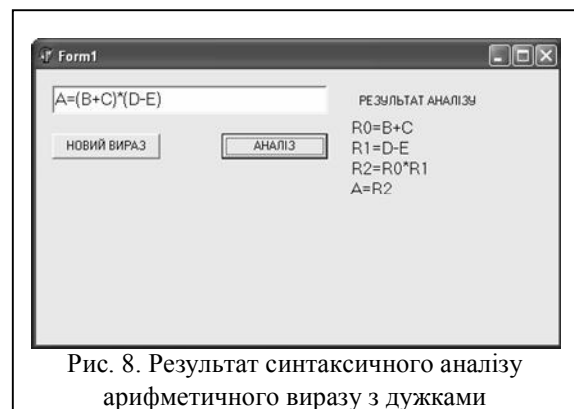


Рис. 8. Результат синтаксичного аналізу арифметичного виразу з дужками

Як бачимо, результат відповідає очікуванням – спочатку виконані операції в дужках і вже потім – множення.

Далі було перевірено результат синтаксичного аналізу арифметичного виразу з дужками, в яких містяться вирази з операціями різних пріоритетів. Отримано такий результат (рис. 9).

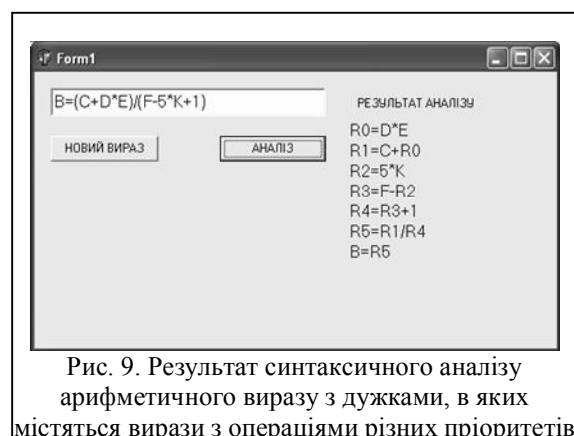


Рис. 9. Результат синтаксичного аналізу арифметичного виразу з дужками, в яких містяться вирази з операціями різних пріоритетів

Як бачимо, результат відповідає очікуванням.

4. На останньому етапі перевірено роботу синтаксичного аналізу для арифметичного виразу з вкладеними дужками. Отримано такий результат (рис. 10).

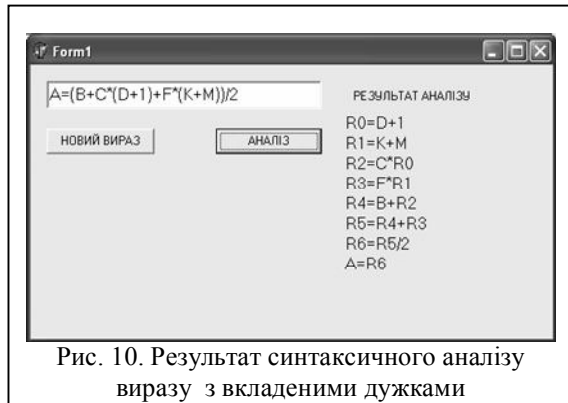


Рис. 10. Результат синтаксичного аналізу виразу з вкладеними дужками

Отримано правильний результат – спочатку виконані операції у внутрішніх скобках, потім – у зовнішніх.

Ще один приклад синтаксичного аналізу досить складного арифметичного виразу (рис. 11).

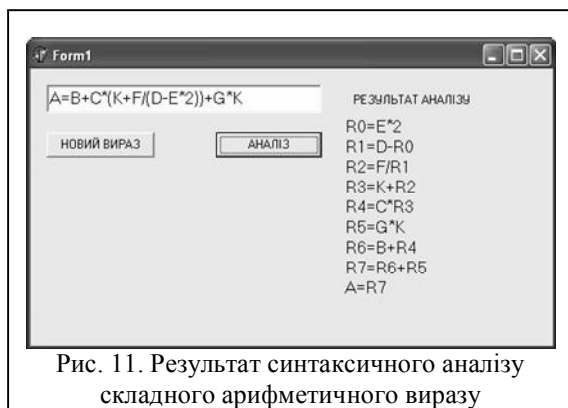


Рис. 11. Результат синтаксичного аналізу складного арифметичного виразу

Таким чином, наведені приклади демонструють, що властивості і методи нових класів працюють у відповідності до вимог та об'єктної моделі.

Основні результати роботи

Основним результатом роботи є те, що на підставі вимог до функціонування підсистеми синтаксичного аналізу компілятору розроблено прикладний додаток, який тестує властивості і методи нових класів — «Арифметичний вираз» і «Допоміжна таблиця», а також демонструє їх можливості для розробки підсистеми синтаксичного аналізу арифметичних виразів.

Отримані результати підтверджують правильність програмної реалізації запропонованих класів.

Список літератури

1. Бондаренко М.А. Розробка класів для підсистем синтаксичного аналізу трансляторів з алгоритмічних мов високого рівня / М.А. Бондаренко, В.А. Жилин, Д.П. Панасенко // Системи обробки інформації. – Х.: XV ПС, 2010. – Вип. 5 (86). – С. 20-24.
2. Бондаренко М.А. Програмування у середовищі Delphi / М.А. Бондаренко. – Х.: ФОП Лібуркіна Л.М., 2007. – 600 с.
3. Культин Н. Программирование в Delphi 5 / Н. Культин. – СПб.: Питер, 2000. – 464 с.
4. Фаронов В.В. Delphi 5: Учебный курс / В.В. Фаронов. – М.: Нолидж, 2000. – 608 с.

Надійшла до редколегії 27.04.2010

Рецензент: д-р техн. наук, проф. В.А. Краснобаєв, Харківський національний технічний університет сільського господарства ім. П. Василенка, Харків.

ТЕСТИРОВАНИЕ КЛАССОВ И ПРИМЕР ИХ ИСПОЛЬЗОВАНИЯ В КОМПИЛЯЦИОННОЙ ПОДСИСТЕМЕ СИНТАКСИЧЕСКОГО АНАЛИЗА АРИФМЕТИЧЕСКИХ ВЫРАЖЕНИЙ

Н.А. Бондаренко, В.А. Жилин, Д.П. Панасенко

В работе приведен пример использования новых классов в системе объектно-ориентированного программирования, которые могут быть применены при разработке подсистемы синтаксического анализа арифметических выражений. Разработка примера осуществлена средствами системы ООП Delphi6.

Ключевые слова: классы данных, объектно-ориентированное программирование, подсистемы синтаксического анализа.

TESTING OF CLASSES AND EXAMPLE OF THEIR USE IS IN THE COMPILING SUBSYSTEM OF SYNTACTIC ANALYSIS OF ARITHMETIC EXPRESSIONS

N.A. Bondarenko, V.A. Zhilin, D.P. Panasenko

The example of the use of new classes is in-process resulted in the system of the object-oriented programming, which can be applied at development of subsystem of syntactic analysis of arithmetic expressions. Development of example is carried out facilities of the system of OOP Delphi6.

Keywords: classes of information, object-oriented programming, subsystems of syntactic analysis.