

УДК 377:004

**Базурін Віталій Миколайович**

кандидат педагогічних наук, доцент кафедри професійної освіти та комп'ютерних технологій  
Глухівський національний педагогічний університет імені Олександра Довженка, м. Глухів, Україна  
u-3700@ukr.net

## МЕТОДИКА НАВЧАННЯ УЧНІВ РОЗВ'ЯЗУВАННЯ ЗАДАЧ З ОПРАЦЮВАННЯ МАСИВІВ У СЕРЕДОВИЩІ ВІЗУАЛЬНОГО ПРОГРАМУВАННЯ DELPHI

**Анотація.** Візуальне середовище програмування Delphi надає широкі можливості щодо наочного відображення масивів. Існує низка компонентів екранної форми Delphi, за допомогою яких можна наочно відобразити масив на формі. Наразі існують відмінності опрацювання масивів мовою Паскаль і в середовищі програмування Delphi. У статті розкрито зазначені особливості й описано елементи методики навчання учнів розв'язування навчальних задач з використанням візуальних компонентів Delphi, зокрема послідовність і логіку викладення навчального матеріалу з опрацювання масивів з використанням компонентів TStringGrid і TMemo.

**Ключові слова:** методика; програмування; масив; Delphi.

### 1. ВСТУП

**Постановка проблеми.** Перехід до інформаційного суспільства зумовлює необхідність оволодіння учнями елементами алгоритмізації та програмування. Програмою з інформатики для загальноосвітньої школи академічного рівня передбачено вивчення елементів програмування в 11-му класі, а саме: розробка програм у візуальному середовищі програмування Delphi. Науковцями розроблено також програму курсу за вибором «Основи візуального програмування». Отже, сучасний стан розвитку інформаційно-комунікаційних технологій враховано під час розробки програми з інформатики.

Водночас недостатньо вивченими залишаються питання методики навчання учнів опрацювання масивів. Більшість наукових праць розглядає типові задачі на опрацювання масивів (пошук елементів у масивах, сортування масивів тощо). Візуальне середовище Delphi надає інші, більш широкі можливості опрацювання масивів, введення і виведення даних у масив тощо.

**Аналіз останніх досліджень і публікацій.** Проблемам розроблення структури і змісту шкільного курсу інформатики присвячені дослідження М. І. Жалдака, Ю. С. Рамського, Н. В. Морзе, В. Д. Руденка, В. В. Лапінського та інших науковців.

Розв'язання проблеми навчання учнів програмування відображено у працях М. І. Жалдака, Т. П. Караванової, Ю. С. Рамського, Н. В. Морзе та інших.

Останнім часом усе більшої актуальності набуває навчання учнів програмування у сучасних середовищах програмування, таких як Delphi, C++ Builder, Visual Basic.NET тощо.

**Метою статті** є обґрунтування й описання елементів методики навчання учнів розв'язування навчальних задач із використанням візуальних компонентів системи програмування Delphi.

### 2. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

#### 2.1. Визначення основних понять

Масив – це структура даних, яка характеризується іменем та індексом.

Програмування – процес створення програм або програмного забезпечення з використанням мов програмування, тобто процес реалізації одного або кількох взаємопов'язаних алгоритмів на певній мові програмування [10].

Мова програмування – це формальна знакова система, призначена для написання програм [12].

Комп'ютерна програма – набір інструкцій у вигляді слів, цифр, кодів, схем, символів чи у будь-якому іншому вигляді, виражених у формі, придатній для зчитування комп'ютером, які приводять його в дію для досягнення певної мети або результату [8].

Додаток – програма, призначена для виконання певних завдань і розрахована на безпосередню (опосередковану інтерфейсом програми) взаємодію користувача з комп'ютером [34].

Форма – візуальний компонент, що реалізується як вікно Windows [13].

Класи – програмно описані спеціальні типи, які містять поля, методи і властивості [11, с. 230].

Об'єкт – окремий екземпляр класу [34].

Метод – це інкапсульовані в об'єкті класу процедури і функції [11, с. 233].

Властивості – спосіб подання даних і доступу до них у об'єктах класу.

## 2.2. Особливості інтерфейсу програм, створених у середовищі Delphi

Шкільною програмою з інформатики рекомендовано вибирати для вивчення програмування у школі середовище візуального програмування Delphi.

На відміну від Паскаля, у середовищі Delphi учень створює повноцінний додаток, видима частина якого представлена у вигляді форми. Форма може містити значну кількість компонентів. З їх допомогою можна більш наочно подати вхідні дані і результат роботи програми. Зупинимось на тих аспектах, які пов'язані з опрацюванням масивів.

Розв'язуючи будь-яку задачу на опрацювання масивів, учень зустрічається з проблемою: а як швидко побачити результат роботи програми? Правильно чи неправильно програма сортує елементи масиву? Чи справді в масиві немає елемента із заданим значенням? Бажано, щоб учень бачив перед собою і сам масив (у його початковому стані), й, одночасно – результат роботи програми (відсортований масив або кількість шуканих елементів). У цьому випадку учень має можливість візуально перевірити правильність програми.

Візуальне програмне середовище Delphi має низку компонентів, які доцільно використати для демонстрування результатів роботи програми з опрацювання масивів. До цих компонентів відносяться багаторядкове поле (Memo), текстове поле з форматуванням (RichEdit) і таблиця (StringGrid). Розглянемо можливості, які вони забезпечують для відображення масиву.

Багаторядкове поле (Memo) (рис. 1) має вигляд прямокутника, у якому можна розміщувати текст (зазвичай, у рядок). Перевагами цього компонента є набір специфічних методів, які дозволяють додавати у компонент Memo рядок, видаляти рядок, зберігати вміст цього компонента у файл тощо.

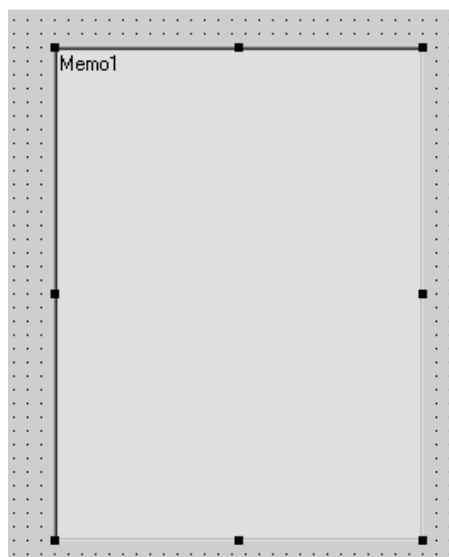


Рис.1. Загальний вигляд багаторядкового поля (Мемо)

Цей компонент має низку специфічних властивостей. Стандартні властивості не розглядатимемо.

Розглянемо основні методи об'єкта Lines. До них відносяться:

Destroy – знищити усі рядки (очистити вміст рядків);

Add – додати символ;

AddStrings – додати рядок внизу поля;

Clear – очистити поле;

Delete – вилучити об'єкт Мемо;

Insert – вставити рядок після вказаного;

LoadFromFile – завантажити вміст поля з файла;

SaveToFile – зберегти вміст поля у файл.

Цей компонент оперує символьними і рядковими даними, тому всі числові змінні доцільно перетворювати в рядкові. Для цього, зазвичай, використовуються процедури val, IntToStr, StrToInt, str. Їх призначення:

Str(x:б:2, s) – перетворення дійсного числа у рядкову змінну;

Val(s,x,cod) – перетворення рядкової змінної в дійсну;

s:=IntToStr(y) – перетворення цілого числа у рядок;

y:=StrToInt(s) – перетворення рядка в ціле число.

Наступний компонент – таблиця (StringGrid) (рис. 2). З його допомогою можна виводити на екран і зберігати у пам'яті комп'ютера під час роботи програми одновимірний і двовимірний масиви. Ця таблиця має рядок заголовків стовпчиків угорі і стовпчик заголовків рядків зліва. Наявність або відсутність зазначених заголовків встановлюються за допомогою властивостей FixedCols (кількість заголовків стовпчиків) і FixedRows (кількість заголовків рядків). За допомогою властивостей ColCount і RowCount можна змінити відповідно кількість стовпчиків і кількість рядків у таблиці.

Для того щоб надати таблиці зручного вигляду для відображення одновимірного масиву, слід встановити значення властивостей FixedCols і FixedRows рівними нулю, значення властивості RowCount – 1, а властивості ColCount – таким самим, як і кількість елементів одновимірного масиву. Для відображення двовимірного масиву слід також змінити значення властивості RowCount на потрібне число.

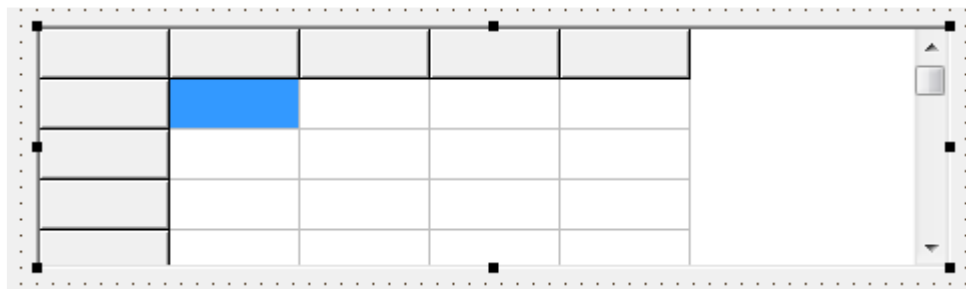


Рис. 2. Загальний вигляд таблиці (StringGrid)

Нумерація рядків і стовпчиків починається з нуля. Ця особливість є однією з відмінностей в описанні масивів мовою Паскаль і в середовищі Delphi, тому створює певні незручності під час виведення у компонент StringGrid елементів масиву, оскільки в Object Pascal прийнято нумерувати елементи масиву, починаючи з 1. Тому під час виведення масиву в таблицю слід передбачити цей випадок.

Типова процедура для виведення елементів масиву дійсних чисел в однорядковий компонент StringGrid має такий вигляд:

```
For i:=1 to n do
Begin
Str(m[i]:8:2,s);
StringGrid1.Cells[i-1,0]:=s;
End;
```

Отже, середовище візуального програмування надає широкі можливості для унаочнення виведення елементів масиву. Відповідно послідовність подання навчального матеріалу має певні відмінності від тієї, яка застосовується під час вивчення мови програмування Паскаль.

### 2.3. Методика вивчення способів опрацювання масивів і використання з цією метою компонентів форм

Нині існує два підходи до навчання візуального програмування [4]. Відповідно до першого з них, спочатку вивчається мова Паскаль (Паскаль, Object Pascal або аналогічна), а вже потім, після засвоєння правил створення програм, – візуальне середовище. Другий підхід передбачає, що вивчення програмування обмежується навчанням розроблення програм у візуальному середовищі програмування. У цьому випадку основні алгоритмічні конструкції вивчаються після вивчення візуальних компонентів форм.

Шкільною програмою з інформатики академічного рівня [34] передбачено застосування другого підходу до вивчення програмування. Той же самий підхід застосовується під час вивчення курсу за вибором «Основи візуального програмування» у 10–11-их класах середньої школі [34].

У своєму дослідженні ми будемо спиратися на ідеї даного підходу.

На нашу думку, ще до пояснення опрацювання масивів, слід ознайомити учнів з компонентами Memo і StringGrid, їх властивостями і методами. Наразі слід дотримуватися принципів «від загального до часткового» і «від простого до складного». Учні вже мають уявлення про найчастіше використовувані компоненти для введення-виведення даних: TLabel, TEdit, TLabeledEdit та інші. Тому варто згадати з учнями особливості цих компонентів, їхні основні властивості. Доцільно задати учням запитання типу: «Скільки змінних можна ввести за один раз у компонент TEdit?», «Як ввести кілька змінних за допомогою компонента TEdit?», «Які відмінності між

компонентами TEdit і TLabelEdit?», «Як вивести 4–5 змінних на форму так, щоб їх було видно одночасно?».

Після цього вводяться поняття компонентів TMemo і TStringGrid. Перший із них є багаторядковим текстовим полем, другий – таблицею. Після цього можна приступати до вивчення кожного компонента окремо, принагідно наголосивши на тому, що ці компоненти є об'єктами, властивостями яких можна керувати програмно.

Звернення (виклик і впровадження у створювану програму) до компонента TMemo здійснюється з палітри компонентів Standard. Призначення цього компонента – виведення на екран багаторядкового тексту.

Доцільно продемонструвати учням таблицю з **властивостями компонента TMemo** (табл. 1). Слід нагадати учням, що ці властивості не всі, а тільки основні.

Таблиця 1

### Основні властивості компонента TMemo

Назва властивості	Характеристика властивості	Можливі значення
BorderStyle	Зміна обрамлення форми	
Font	Шрифт	
Enabled	Активність компонента	
Height	Висота	
Left	Ліва межа	
Lines	Рядки	
Name	Ім'я компонента	
ReadOnly	Режим «тільки для читання»	
ScrollBars	Наявність смуг прокручування	
Top	Координата верхньої межі	
Visible	Видимість	
Width	Ширина	

Третій стовпчик цієї таблиці незаповнений. Доцільно надати учням можливість самостійно заповнити останній стовпчик цієї таблиці. На нашу думку, ефективнішим за все є робота у візуальному середовищі Delphi. Учні самостійно працюють з вікном Інспектора об'єктів. Виокремлюючи потрібну властивість, учні можуть змінювати її значення. Саме під час роботи в середовищі Delphi учні не тільки ознайомлюються з властивостями компонента TMemo, а й набувають практичних навичок зміни властивостей об'єкта за допомогою Інспектора об'єктів. У процесі роботи з властивостями компонента TMemo в учнів продовжується формування операційного компонента, який надалі генералізуватиметься як спільний для роботи з об'єктами візуального програмування (не тільки в середовищі Delphi).

Далі учнів слід ознайомити з основними **методами** цього компонента. Спочатку доцільно згадати разом з учнями, що таке метод. Складова компонента TMemo – рядок (об'єкт Lines). До цього об'єкта застосовуються такі основні методи:

```

Memo1.Lines.Add('Мій рядок'); // Рядок вставляється в кінці поля Memo1
Memo1.Lines.Insert(3, 'Мій рядок'); // Рядок вставляється у полі Memo1 перед
третім рядком
Memo1.Lines.Delete(3); // видалення третього рядка у полі Memo1

```

Далі учням пояснюється, як додати рядок або символ до компонента. Пояснення слід проілюструвати конкретними прикладами. Наприклад, щоб додати фразу «Доброго ранку», слід виконати команду:

```
Memo1.Lines.Add('Доброго ранку');
```

Для того щоб видалити рядок, слід скористатися методом Memo1.Lines.Clear.

Для того щоб очистити компонент Memo1, слід використати метод Memo1.Lines.Clear;

Після цього слід нагадати учням, що змінні, які додаються до компонента TMemo, необхідно перетворювати у рядковий тип. Доцільно провести аналогію з компонентом типу TEdit і запропонувати учням заповнити таблицю, у якій вказати спільні якості і відмінності між компонентами TMemo і TEdit.

Далі слід наголосити на недоліках використання компонента TMemo. Як з його допомогою вивести на екран таблицю? Як вивести на екран значення аргумента і функції? У компонент TMemo недоцільно виводити таблицю, оскільки важко розробити програму для її подальшого опрацювання. Для цього можна використати інший компонент – TStringGrid.

Після цього можна приступати до вивчення компонента TStringGrid, постійно проводячи аналогії з компонентом TMemo.

Компонент TStringGrid є таблицею. Упровадження компонента в програму виконується зверненням до відповідної позиції в палітрі Additional. Даний компонент призначений для введення / виведення таблиці, яка може складатися з одного стовпчика, одного рядка або багатьох рядків і стовпчиків.

Повідомивши учням призначення цього компонента, доцільно продемонструвати їм таблицю із записом основних властивостей компонента (табл. 2) і запропонувати їм самостійно заповнити третій стовпчик таблиці, користуючись візуальним середовищем Delphi. Під час виконання зазначених дій, як уже було зазначено, формується операційний компонент уміння змінювати властивості компонента у вікні Інспектора об'єктів.

Слід наголосити, що компонент має як стандартні властивості, так і спеціальні.

Таблиця 2

### Властивості Options компонента StringGrid

Назва властивості	Характеристика властивості	Можливі значення
goFixedVertLine	Наявність вертикальних роздільних ліній між фіксованими комірками	
goFixedHorzLine	Наявність горизонтальних роздільних ліній між фіксованими комірками	
goVertLine	Наявність вертикальних роздільних ліній між звичайними комірками	
goHorzLine	Наявність горизонтальних роздільних ліній між звичайними комірками	
goRangeSelect	Можливість виділити діапазон комірок	
goDrawFocusSelected	Зафарбовування комірки з фокусом введення	
goRowSizing	Можливість змінювати висоту рядків мишкою	
goColSizing	Можливість змінювати ширину	

	стовпчиків мишкою	
goRowMoving	Можливість змінювати номер рядка (переміщувати рядок мишкою)	
goColMoving	Можливість переміщувати номер стовпчика (переміщувати його мишкою)	
goEditing	Можливість редагувати вміст комірки з клавіатури	
goTabs	При значенні True фокус зміщується на наступну комірку в таблиці, при значенні False – на наступний компонент	
goRowSelect	Виділяється увесь рядок з фокусованою коміркою	
goAlwaysShowEditor	При значенні True вміст комірки при отриманні фокуса відразу доступний для редагування, при значенні False необхідно натиснути спочатку на Enter або F2	
goThumbTracking	При значенні True переміщення бігунка прокрутки мишкою викликає переміщення комірок, при значенні False комірки переміщуються тільки при відпусканні бігунка	

Необхідно пояснити, що нумерація рядків і стовпчиків починається з 0. Особливу увагу слід акцентувати на властивостях, які визначають рядок і стовпчик заголовків: FixedCols і FixedRows. Тобто, для того щоб таблиця не мала заголовків рядків і заголовків стовпчиків, значення цих властивостей слід встановити рівними нулю. Однорядкова таблиця має значення властивості RowCount рівним 1. Таблиця, яка складається з одного стовпчика, має значення властивості ColCount рівним 1.

Нумерація комірок у компоненті TStringGrid має вигляд: StringGrid1.Cells[i, j], де i – номер стовпчика, j – номер рядка.

Отримати координати комірки, на якій встановлено курсор, можна за допомогою команди:

```
StringGrid1.MouseToCell(X, Y, ACol, ARow); //
```

Змінна S зчитується з комірки за допомогою команди:

```
S:=StringGrid1.Cells[i, j];
```

Записати у комірку таблиці значення змінної S можна за допомогою команди:

```
StringGrid1.Cells[i, j]:=S.
```

Окремо слід звернути увагу учнів на те, що дані в цих компонентах мають рядковий тип даних (String). Тому при поміщенні чисел у багаторядкове поле або таблицю необхідно перетворювати їх у рядок. Зчитуючи дані з компонентів Memo і StringGrid, їх необхідно перетворювати у дійсні або цілі числа.

Для зчитування даних з комірки слід використати команду:

```
S:=StringGrid1.Cells[1,4];
```

```
Val(s,t,cod);
```

Пристаюючи до навчання учнів сортування масивів, необхідно дотримуватися такої логічної послідовності викладу навчального матеріалу: нагадати поняття масиву → пояснити основні задачі з опрацювання масивів → сформулювати поняття про пошук елемента у масиві → сформулювати уявлення про методи пошуку елементів у масиві → навести алгоритм програми пошуку елементів у масивів на Паскалі → навести приклади виведення результатів пошуку елемента у масиві у компонент на формі.

Наступний етап – вивчення масиву. Масив – це структура даних, яка характеризується іменем та індексом. Необхідно навести приклади масивів і запропонувати учням навести свої приклади масивів. Масив оголошується у вигляді глобальної або локальної змінної за допомогою команди:

```
Var m:array[1..5] of real;
```

Далі слід провести аналогію з життям і запропонувати учням визначити, які саме задачі виникають під час опрацювання масивів: наповнення масиву значеннями, сортування елементів масиву, пошук елемента в масиві, виведення масиву на екран.

Для прикладу можна продемонструвати роботу програми, яка заповнює масив випадковими числами, виконує пошук елемента у масиві, очищає всі поля (рис. 3). Екранна форма містить такі компоненти: TStringGrid, TLabelEdit, TButton.

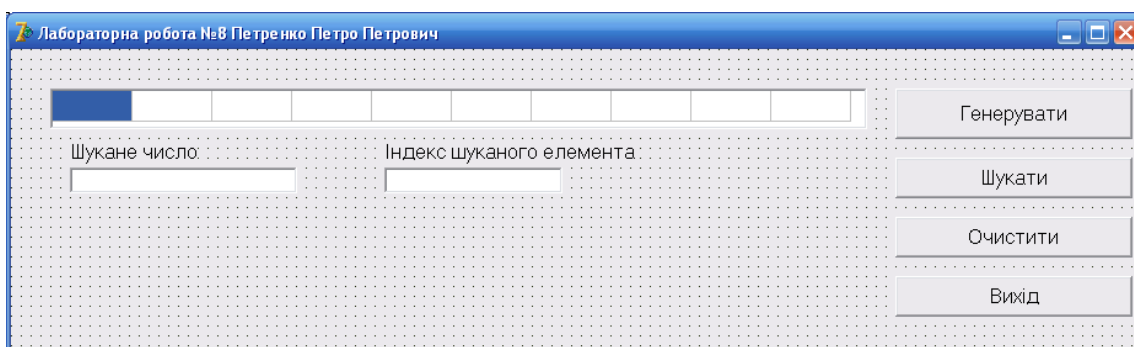


Рис. 3. Зовнішній вигляд форми екранної програми для опрацювання масиву

Найпростіша з цих задач – наповнення елементів масиву і виведення масиву в однорядковий компонент TStringGrid. Запускається вона за допомогою кнопки «Генерувати». Для цього необхідно створити окрему процедуру, яка міститиме такий код:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  randomize;
  for i:=0 to n do
  begin
    m[i]:=random(100);
    StringGrid1.Cells[i,0]:=inttostr(m[i]);
  end;
```

Ця підпрограма заповнює масив випадковими числами і виводить його вміст у компонент StringGrid1.

Наступна процедура виконує пошук числа, заданого користувачем, у масиві:

```
procedure TForm1.Button2Click(Sender: TObject);
begin
```



```

k:=false;
s:=labelededit1.Text;
x:=strtoint(s);
for i:=0 to n do
begin
s:=stringgrid1.Cells[i,0];
m[i]:=strtoint(s);
if m[i]=x then
begin
k:=true;
ip:=i;
end;
end;
if k then
begin
labelededit2.Text:=inttostr(ip);
labelededit2.Color:=clgreen;
end
else
begin
labelededit2.Color:=clred;
labelededit2.Text:='';
end;
end;
end;

```

Якщо число знайдено, то у відповідному полі виводиться індекс елемента, який дорівнює заданому числу. Якщо шуканий елемент відсутній у масиві, то поле LabeledEdit2 забарвлюється у червоний колір.

Для того щоб очистити масив і компонент StringGrid1, служить наступна підпрограма:

```

procedure TForm1.Button3Click(Sender: TObject);
begin
labelededit1.Text:='';
labelededit2.Text:='';
labelededit2.color:=clwhite;
for i:=0 to n do
begin
StringGrid1.Cells[i,0]:='';
end;
end;

```

Завершити вивчення теми доцільно детальним аналізом запропонованої програми. Для цього учням можна запропонувати скласти коментарі до програми і дописати їх у код програми. Після засвоєння основних правил використання компонента TStringGrid під час опрацювання масиву можна приступати до вивчення алгоритмів сортування масивів і алгоритмів пошуку даних у масиві.

### 3. ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

У результаті дослідження встановлено, що розв'язування задач на опрацювання масивів у середовищі візуального програмування Delphi має певні особливості. Середовище візуального програмування Delphi має низку компонентів, за допомогою яких можна виводити масив на екран і зберігати масив у формі. З використанням компонентів TStringGrid і TMemo учні отримують можливості для унаочнення задачі з опрацювання масивів. Так, під час сортування масивів можна в одному компоненті розміщувати несортований масив, а в іншому – сортований масив. У компонентах TStringGrid і TMemo всі дані зберігаються у рядковому форматі. Це вимагає застосування функцій і процедур перетворення змінних.

Відповідно логіці вивчення задач на опрацювання масивів доцільно побудувати так: поняття масиву → основні задачі з опрацювання масивів → поняття про пошук елемента в масиві → методи пошуку елемента в масиві → алгоритм програми пошуку елементів у масиві на Паскалі → приклади виведення результатів пошуку елемента в масиві у компонент на формі.

Вказані особливості вивчення опрацювання масивів у Delphi доцільно, на нашу думку, врахувати у процесі вивчення масивів.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Завадський І. О. Основи візуального програмування / І. О. Завадський, Р. І. Заболотний. – К. : Видавнична група ВНУ, 2009. – 272 с.
2. Завадський І. О. Програма курсу за вибором «Основи візуального програмування» : Для організації профільного навчання у старших класах загальноосвітніх навчальних закладів / І. О. Завадський [Електронний ресурс]. – Режим доступу : <http://mon.gov.ua>.
3. Інформатика. Навчальна програма для 10–11 класів загальноосвітніх навчальних закладів. Академічний рівень [Електронний ресурс]. – Режим доступу : <http://mon.gov.ua>.
4. Лапінський В. В. Проблема вибору першої мови програмування – сьогоднішнє бачення // Комп'ютер у школі та сім'ї. – 2014. – №1 (113). – С. 14–17.
5. Морзе Н. В. Методика навчання інформатики. [Навч. посібник.] / Н. В. Морзе / За ред. М. І. Жалдака. – У 4 ч. – Ч. 4. Методика навчання основ алгоритмізації та програмування. – Київ : Навчальна книга, 2004. – 368 с.
6. Об'єктно-орієнтоване програмування / Вікіпедія [Електронний ресурс]. – Режим доступу: [http://uk.wikipedia.org/wiki/Об'єктно-орієнтоване\\_програмування](http://uk.wikipedia.org/wiki/Об'єктно-орієнтоване_програмування)
7. Прикладное программное обеспечение / Википедия [электронный ресурс]. – Режим доступа : [https://ru.wikipedia.org/wiki/Прикладное\\_программное\\_обеспечение](https://ru.wikipedia.org/wiki/Прикладное_программное_обеспечение).
8. Про авторське право і суміжні права. Стаття 1. Визначення термінів [Електронний ресурс]. – Режим доступу : [http://kodeksy.com.ua/pro\\_avtors\\_ke\\_pravo\\_i\\_sumizhni\\_prava/statja-1.htm](http://kodeksy.com.ua/pro_avtors_ke_pravo_i_sumizhni_prava/statja-1.htm).
9. Спирін О. М. Порівняльний аналіз програмних технологій операційної системи Windows 8 для навчання програмування / О. М. Спирін, П. Г. Шевчук // Інформаційні технології і засоби навчання [Електронний ресурс]. – 2014. – №2(40). – Режим доступу : <http://journal.iitta.gov.ua/index.php/itlt/article/view/997/759#.VNXsnei9p5I>.
10. Технологии программирования [Электронный ресурс]. – Режим доступа : <http://www.maksakov-sa.ru/TehProgram/index.html>.
11. Фаронов В. В. Delphi 5. Руководство программиста / В. В. Фаронов. – М. : Нолидж, 2001. – 880 с.
12. Языки программирования [Электронный ресурс]. – Режим доступа : <http://life-prog.ru/>.
13. Языки программирования Delphi и Pascal [Электронный ресурс]. – Режим доступа : <http://www.maksakov-sa.ru/TeorDelphi/FormDelphi/index.html>.

*Матеріал надійшов до редакції 07.02.2015 р.*

## МЕТОДИКА ОБУЧЕНИЯ УЧАЩИХСЯ РЕШЕНИЮ ЗАДАЧ ПО ОБРАБОТКЕ МАССИВОВ В СРЕДЕ ВИЗУАЛЬНОГО ПРОГРАММИРОВАНИЯ DELPHI

**Базурин Виталий Николаевич**

кандидат педагогических наук, доцент кафедры профессионального образования и компьютерных технологий

Глуховский национальный педагогический университет имени Александра Довженко, г. Глухов, Украина

*u-3700@ukr.net*

**Аннотация.** Визуальная среда программирования Delphi предоставляет широкие возможности наглядного отображения массивов. Существует ряд компонентов экранной формы Delphi, с помощью которых можно наглядно отобразить массив на форме. Программы по разработке массивов в среде Delphi имеют свои отличия от таких же программ на Паскале. В статье раскрыты эти различия. Также раскрыты особенности

методики обучения учащихся решению задач по обработке массивов с использованием визуальных компонентов Delphi. Раскрыто последовательность и логику изложения учебного материала по разработке массивов с использованием компонентов TStringGrid и TMemo.

**Ключевые слова:** методика; программирование; массив; Delphi.

## METHODS OF TEACHING STUDENTS FOR SOLVING PROBLEMS ON ARRAYS PROCESSING IN THE DELPHI VISUAL PROGRAMMING ENVIRONMENT

**Vitalii M. Bazurin**

Ph.D. (pedagogical sciences), assistant professor of the Department of Vocational Education and Computer Technology

Oleksandr Dovzhenko Hlukhiv National Pedagogical University, Hlukhiv, Ukraine

*email@email.com*

**Abstract.** Delphi visual programming environment provides ample opportunities for visual mapping arrays. There are a number of Delphi screen form components, which help you to visualize the array on the form. Processing arrays programs in Delphi environment have their differences from the same programs in Pascal. The article describes these differences. Also, the features of student learning methods for solving problems of array processing using Delphi visual components are highlighted. It has been exposed sequence and logic of the teaching material on arrays processing using TStringGrid and TMemo components.

**Keywords:** methods; programming; array; Delphi.

### REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Zavadskiy I. O. Fundamentals of visual programming / I.O.Zavadskiy, R.I.Zabolotnyi. – K. : Publishing Group BHV, 2009. – 272 p. (in Ukrainian).
2. Zavadskiy I. O. The program of elective courses "Fundamentals of visual programming": To organize profile education in upper secondary schools [online] / I. O. Zavadskiy. – Available from : <http://mon.gov.ua> (in Ukrainian).
3. Informatics. The curriculum for 10–11 grades of secondary schools. Academic level [online]. – Available from : <http://mon.gov.ua> (in Ukrainian).
4. Morze N. V. Methods of teaching science. [Training. Manual.] / N. V.Morze / Ed. M. I. Zhaldak. – At 4 h. – Paragraph 4. Methods of teaching the basics of algorithms and programming. – Kyiv : Training Book, 2004. – 368 p. (in Ukrainian).
5. Lapinskii V. The choice of the first programming language – today's vision // Computer in school and family. – 2014. – №1 (113). – P. 14–17 (in Ukrainian).
6. Object-oriented programming / Wikipedia [online]. – Available from : [http://uk.wikipedia.org/wiki/Об'єктно-орієнтоване\\_програмування](http://uk.wikipedia.org/wiki/Об'єктно-орієнтоване_програмування) (in Ukrainian).
7. Application software / Wikipedia [online]. – Available from : [https://ru.wikipedia.org/wiki/Прикладное\\_программное\\_обеспечение](https://ru.wikipedia.org/wiki/Прикладное_программное_обеспечение) (in Russian).
8. On Copyright and Related Rights. Article 1. Definitions [online]. – Available from : [http://kodeksy.com.ua/pro\\_avtors\\_ke\\_pravo\\_i\\_sumizhni\\_prava/statja-1.htm](http://kodeksy.com.ua/pro_avtors_ke_pravo_i_sumizhni_prava/statja-1.htm) (in Ukrainian).
9. Spirin O. M. Comparative analysis software technology of Windows 8 for learning programming [online] / O. M. Spirin, P. H. Shevchuk // Agriculture and forestry education. – 2014. – №2 (40). – Available from : <http://journal.iitta.gov.ua/index.php/itlt/article/view/997/759#.VNXsnei9p5I> (in Ukrainian).
10. Technologies programming [online]. – Available from : <http://www.maksakov-sa.ru/TehProgram/index.html> (in Russian).
11. Faronov V. V. Delphi 5. Guidance programmer / V. V. Faronov. – M. : Nolydzh, 2001. – 880 p. (in Russian).
12. Languages of programming [online]. – Available from : <http://life-prog.ru/> (in Russian).
13. Languages of programming Delphi and Pascal [online]. – Available from : <http://www.maksakov-sa.ru/TeorDelphi/FormDelphi/index.html> (in Russian).