

Застосування Delphi для реалізації методу проектів при навчанні інформатики

Метод проектів, запропонований у 1919 році в місті Дальтон (США) американським педагогом Е. Паркхарстом з метою індивідуалізації процесу навчання, використовувався в Радянському Союзі в 20-30 ті роки ХХ століття. Бурхливий розвиток інформаційних технологій сприяє відродженню цього методу навчання в теперішній час. Все більше українських педагогів-практиків використовують цей метод в своїй викладацькій діяльності, зокрема, М.Е. Єгорова [1] та інші.

Метод проектів привертає увагу вчителів своєю орієнтацією на самостійну пізнавальну діяльність учнів – індивідуальну, парну, групову, яку учні виконують протягом певного відрізка часу. Він надає значні можливості для: 1) формування дослідницьких умінь: збирати необхідну інформацію, висувати гіпотезу, робити висновки і умовиводи, використовувати для роботи з інформацією нові інформаційні технології; 2) навчання самостійного мислення і діяльності, системного підходу у самоорганізації, груповій взаємодії; 3) підвищення мотивації навчання та самонавчання: актуалізація незатребуваних знань та стимуляція надбання нових; 4) реалізації методики випереджуючого навчання; 5) формування особистісних якостей учнів: взаєморозуміння, взаємоповаги, відповідальності; 6) організації творчої діяльності учнів; 7) структурування знань учнів шляхом встановлення міжпредметних зв'язків; 8) вироблення навичок ефективного використання комп'ютера в своїй подальшій професійній діяльності; 9) формування проектної культури (знань, умінь визначення потреб і можливостей діяльності при виконанні проекту, висування спектру ідей виконання проекту, вибір оптимальної ідеї, дослідження цієї ідеї, планування, організація і виконання роботи з реалізації проекту, включаючи набуття додаткових знань та умінь, оцінка і його презентація); 10) формування інформаційної культури (знання, уміння використовувати методи збирання, зберігання, опрацювання та використання інформації з різних джерел).

В інформатиці метод проектів в основному використовується в двох напрямках: телекомунікаційні проекти та проекти створення комп'ютерних моделей реальних систем. Практика показує ефективність застосування при навчанні інформатики програмних проектів моделювання функціонування реальних систем. Так, наприклад, помічено, що максимальні знання із навчаючої програми отримують розробники цієї системи, а не учні, для яких

ця система призначена. В ході розробки моделі системи або навчаючої програми відбувається “перетворення знань, що використовуються, вони повертаються різними своїми аспектами і становляться більш глибокими” [3, С.86].

Під проектом розуміють задачу, що ясно описана, має кінцевий результат, практичну значимість і структурний зміст.

Приклад. Розробити програмний проект для контролю знань, в якому передбачити:

- 1) реалізацію виду контролю, коли учень вибирає правильну відповідь на питання із наданих;
- 2) випадковий вибір питань;
- 3) підрахування кількості питань, на які отримані правильні відповіді, кількості питань, на які отримані неправильні відповіді, кількості набраних балів, кількості втрачених балів;
- 4) запис результатів тестування групи осіб до таблиці;
- 5) подання на круговій діаграмі даних, що містяться у виділеному в таблиці фрагменті, для візуального порівняння;
- б) сортування даних в таблиці, підрахування суми, середнього значення даних, виділених в таблиці.

Структурність змістової частини проекту передбачає можливість виділення етапів його реалізації та їх задач (підзадач), які в більшості випадків розв’язуються за допомогою різного програмного забезпечення, із застосуванням різних формальних мов описування об’єктів та процесів. Так в наведеному прикладі розв’язування задач 1)-3) доцільніше здійснювати деякою мовою програмування (наприклад, Pascal), а задачі 4)-6) зручніше розв’язувати в середовищі електронної таблиці (наприклад, Excel). Тим самим учні повинні володіти різними формальними мовами двох програмних засобів, які в більшості випадків неможливо поєднати в єдиному програмному проекті, в одному вікні програми, як це можна зробити в середовищі візуального програмування, наприклад, Delphi.

В основі Delphi лежить простіша для школярів мова програмування Object Pascal, і, відповідно, витримується мовна лінія Н.Вірта, дотримання якої дозволяє прищеплювати учням програмістську культуру. Delphi – це інтегроване середовище розробки програм, яке об’єднує в собі механізми програмування з іншими засобами подання і опрацювання даних: базами даних, електронними таблицями, графічним, текстовим редакторами, містить множину готових компонентів для підтримки Web-програмування тощо.

В основі створення власних програмних проектів засобами системи візуального програмування Delphi лежить комбіноване використання різних компонент – візуальних (кнопок, панелей, таблиць, діаграм тощо) і невізуальних (таймер, діалоги відкриття і збереження файлів тощо) об’єктів з власними властивостями і методами опрацювання, з яких, як будинок з цеглин, будуються

проекти – користувацькі програми. Поєднуються ці “цеглини” – компоненти в одній програмі за допомогою єдиної формальної мови програмування Object Pascal.

Розглянемо реалізацію наведеного прикладу програмного проекту для контролю знань, єдина форма (вікно) якого подана на рис.1.

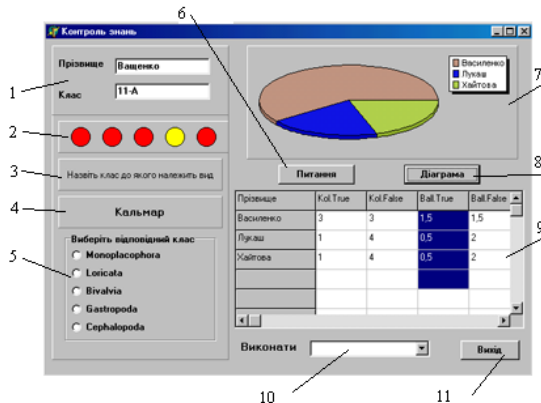


Рис.1. Форма програмного проекту для контролю знань і опрацювання результатів.

Позначення до рис.1:

- 1 – блок реєстрації учасника тестування; передбачається введення прізвища, та номера класу;
- 2 – блок для візуального подання результатів тестування одного учасника; для тестування одній людині пропонується п'ять запитань, при правильній відповіді на питання круг замальовується жовтим кольором, при неправильній відповіді – червоним кольором;
- 3-4 – панелі для подання запитання;
- 5 – блок вибору відповіді на питання;
- 6 – кнопка включення режиму тестування після реєстрації учасника;
- 7 – діаграма для графічного подання виділених в таблиці (9) даних про результати тестування кількох учасників;
- 8 – кнопка, за натисненням якої відбувається малювання діаграми;
- 9 – таблиця, кожний рядок якої містить результат тестування окремого учасника (прізвище, кількість питань, на які дані правильні відповіді, кількість питань, на які дані неправильні відповіді, кількість набраних балів, кількість втрачених балів);
- 10 – компонента для вибору дії над виділеними в таблиці (9) даними (упорядкувати дані, знайти суму, середнє значення тощо);
- 11 – кнопка виходу з програми.

З рис.1 видно, що робота блоків 2-6 спрямована на розв'язування задач 1)-3) завдання, а робота блоків 7-10 – на розв'язування задач 4)-6). Різні за напрямком задачі об'єднані в одному вікні програми за допомогою єдиної формальної мови – мови програмування Object Pascal Delphi. Розглянемо фрагменти програмного коду, що ілюструють окремі підходи до реалізації зазначеного проекту.

Реалізація першої задачі, яка передбачає виведення питання і вибір правильної відповіді тим, хто проходить тестування, здійснюється в блоках 3-5 проекту. Загальне питання, наприклад, “Назвіть клас, до якого належить вид”, виводиться на панель (3) і залишається незмінним протягом роботи програми. Назва окремого виду виводиться на панель (4) і змінюється випадковим чином. Для вибору відповіді на питання застосовується компонента *RadioGroup*, яка містить набір радіокнопок з можливими варіантами відповіді. Вибір цієї компоненти обґрунтовується можливістю реалізації вибору тільки однієї відповіді в один момент часу.

Назви класів для питання і варіанти правильних відповідей можна спочатку зчитувати з файла ('Zawd1.txt'), записувати до відповідних масивів (Nazva, Otv), а потім випадковим чином вибирати з масиву і виводити на панель (4).

Фрагмент програмного коду:

```
Var {Описання змінних}
kol:integer; {кількість питань}
Nazva:array[1..100] of string;
Otv:array [1..100] of 1..5;
procedure wwod;
var i:integer; f:textfile;
begin
assignfile(f,'Zawd1.txt');
reset(f); {Відкриття файлу для зчитування}
readln(f,kol); {Зчитування інформації із файлу}
for i:=1 to kol do begin
  readln(f,Nazva[i]);
  readln(f,Otv[i]);
end;
Closefile(f); {Закриття файлу}
end;
```

Для реалізації випадкового вибору питання або частини питання можна скористатися функцією *random*, для запобігання повторень питань можна позначати в окремому масиві *Dop* номери невибраних питань за 0, а вибраних за 1. При цьому необхідно обнуляти масив при запуску програми на виконання та у випадку відсутності невибраних питань.

Фрагмент програмного коду:

```
Var x, kol:integer; { x - випадковий номер питання; kol - кількість питань}
Dop:array[1..100] of 0..1;
procedure Proov; {обнулення масиву Dop у випадку вибору всіх питань}
var i,f:integer;
```

```

begin
  f:=0;
  for i:=1 to kol do if Dop[i]=1 then f:=f+1;
  if f=kol then for i:=1 to kol do Dop[i]:=0;
end;
procedure Slych; { випадковий вибір питання }
  Label m1;
  begin
    m1: x:=random(kol); { вибирається випадковий номер питання }
    if Dop[x]=0 then begin { якщо питання не вибране, }
      Dop[x]:=1 ; { то воно вибирається і висвітлюється на панелі }
      Form1.Panel3.Caption:=Nazva[x];
      End {в іншому випадку здійснюється перевірка на наявність
невибраних питань і вибирається нове}
      else begin Prov; goto m1; end;
  end;

```

Аналіз відповідей на питання пов'язує програмним кодом на Object Pascal компоненту *RadioGroup* з компонентами виведення простих графічних зображень *Shape*. В залежності від того, яка кнопка групи радіокнопок була вибрана, тобто якого значення набуває властивість *ItemIndex* компоненти *RadioGroup*, цілій глобальній змінній "номер копки" *nbut* присвоюється відповідне значення.

```

Procedure TForm1.RadioGroup1Click(Sender: TObject);
begin
  Case RadioGroup1.ItemIndex of
    0: nbut:=1;
    1: nbut:=2;
    2: nbut:=3;
    3: nbut:=4;
    4: nbut:=5;
  end;
  analiz; nzadan:=nzadan+1;Slych;{аналіз результату відповіді, підрахування
кількості виконаних завдань, виведення наступного питання }
  .....
end;

```

Якщо номер вибраної кнопки *nbut* збігається з правильною відповіддю на питання *otv[x]*, де *x* – випадково вибраний номер питання, то компонента *Shape*, що відповідає певному номеру завдання, замальовується жовтим кольором, в іншому випадку – червоним.

```

procedure analiz;
begin
  if nbut=otv[x] then begin
    case nzadan of {замальовування компонент Shape жовтим кольором}
      1: Form1.Shape1.Brush.Color:=clYellow;
      .....
      5: Form1.Shape5.Brush.Color:=clYellow;
    end;
    otvTrue:=otvTrue+1; {підрахування кількості правильних відповідей}
    BallTrue:=BallTrue+0.5; {підрахування кількості набраних балів}
  end
  else begin
    case nzadan of {замальовування компонент Shape красним кольором}

```

```

1: Form1.Shape1.Brush.Color:=clRed;
.....
5: Form1.Shape5.Brush.Color:=clRed;
end;
otvFalse:=otvFalse+1; { підрахування кількості неправильних відповідей }
BallFalse:=BallFalse+0.5; { підрахування кількості втрачених балів }
end;

```

end;

Перехід до виконання наступних завдань 4)-6), пов'язаних з опрацюванням даних в таблиці, здійснюється через запис до компоненти *StringGrid* (таблиця рядкових величин) результатів реєстрації і тестування окремого учасника після відповіді на останнє п'яте запитання, тобто після натиснення відповідної радіокнопки.

```
procedure TForm1.RadioGroup1Click(Sender: TObject);
```

```
begin
```

```
.....
If nzadan =5 then Zapis; { Якщо номер завдання дорівнював 5, то занести дані в таблицю }
```

```
end;
```

```
Procedure Zapis; { Запис даних до таблиці }
```

```
Form1.StringGrid1.Cells[0,0]:= 'Прізвище'; { Формування заголовків таблиці }
```

```
Form1.StringGrid1.Cells[1,0]:= 'Kol.True';
```

```
Form1.StringGrid1.Cells[2,0]:= 'Kol.False';
```

```
Form1.StringGrid1.Cells[3,0]:= 'Ball.True';
```

```
Form1.StringGrid1.Cells[4,0]:= 'Ball.False';
```

```
Form1.StringGrid1.Cells[0,Uch]:=Form1.Edit1.Text; { Запис даних реєстрації до таблиці, тобто введеного в компоненту Edit1 прізвища особи }
```

```
{ Запис даних про кількість правильних і неправильних відповідей, про кількість набраних і втрачених балів до таблиці }
```

```
Form1.StringGrid1.Cells[1,Uch]:=IntToStr(otvTrue);
```

```
Form1.StringGrid1.Cells[2,Uch]:=IntToStr(otvFalse);
```

```
Form1.StringGrid1.Cells[3,Uch]:=FloatToStr(BallTrue);
```

```
Form1.StringGrid1.Cells[4,Uch]:=FloatToStr(BallFalse);
```

```
Uch:=Uch+1;
```

```
End;
```

В наведеному фрагменті програмного коду змінна *Uch* означає номер особи, що проходить тестування, а тим самим і номер рядка в таблиці, куди записуються дані тестування чергового учасника. Цей номер обнуляється при запуску програми на виконання і нарощується на одиницю при тестуванні наступної особи. Доступ до окремої комірки таблиці здійснюється через властивість *Cells[x,y]* компоненти – таблиці рядків *StringGrid*. У властивості *Cells[x,y]* параметр *x* означає номер стовпчика, а параметр *y* – номер рядка таблиці, що дещо відрізняється від прийнятого позначення в звичайній матриці, де першим вказується номер рядка, а другим параметром – номер стовпчика.

Виділення даних в таблиці у процесі роботи програми здійснюється клавішами управління курсором при натисненій клавіші *Shift*. Про наявність виділеної області даних сигналізує подія *onSelectCell*.

```
Var a,b,c,d:integer; fl:boolean;
```

```

procedure TForm1.StringGrid1SelectCell(Sender: TObject; ACol,
  ARow: Integer; var CanSelect: Boolean);
begin
  a:=StringGrid1.Selection.Left;
  b:=StringGrid1.Selection.Top;
  c:=StringGrid1.Selection.Right;
  d:=StringGrid1.Selection.Bottom;
  fl1:=CanSelect;
end;

```

В параметрах *Left*, *Right*, *Top*, *Bottom* властивості *Selection* (виділення) фіксуються розміри виділеної області: номер стовпчика зліва, номер стовпчика справа, номери верхнього і нижнього рядків. Змінна логічного типу *CanSelect* набуває значення *True*, якщо в таблиці виділена деяка область даних.

Виконання певної операції опрацювання даних виділеної області (розв'язування задачі 6) відбувається після вибору цієї операції в компоненті *ComboBox*, тобто коли трапляється подія *onChange*. Наступний фрагмент програмного коду показує реалізацію знаходження суми даних в комірках виділеної області таблиці.

```

procedure TForm1.ComboBox1Change(Sender: TObject);
var i,j:integer;s:real;
begin

```

```

  If (ComboBox1.Text='Суму')and Fl1 then Begin {Якщо вибрана операція
  знаходження суми і виділена область даних, тобто Fl1=True, то знаходимо суму
  даних }

```

```

    s:=0;
    for i:=a To c do
      for j:=b To d do
        if StringGrid1.Cells[i,j]<>" then {Якщо комірка непорожня, то її значення
        додається до суми}
          s:=s+StrToFloat(StringGrid1.Cells[i,j]);
        End;

```

```

end;

```

Розв'язування задачі 5) здійснюється за допомогою використання стандартної компоненти *Chart*, яка дозволяє будувати діаграми різних типів. Тип діаграми задається як окрема серія (наприклад, *Series[0]*), що вибирається через інспектор об'єктів (*Gradient/Series/Add*). Визначення в *Chart* кількох серій дозволяє будувати сумісні діаграми. Перший тип діаграми у списку вибраних діаграм має індекс 0. Передбачаючи для розв'язування поставленої задачі роботу тільки з однією круговою діаграмою достатньо визначити тільки *Series[0]*. В наступному фрагменті програмного коду показано, як побудувати кругову діаграму за виділеним в таблиці стовпчиком за натисненням кнопки "Діаграма". Така діаграма дозволить, наприклад, візуально порівняти успішність виконання тестів учасниками тестування.

```

procedure TForm1.Button1Click(Sender: TObject);
var i:integer;Cr1:array[1..10]of TColor; {Cr1- масив для задання кольорів }
begin
  Cr1[1]:=RGB(199,150,133);Cr1[2]:=RGB(11,22,234);Cr1[3]:=RGB(180,209,78);
  .....

```

```

Crl[9]:=RGB(112,212,199);Crl[10]:=RGB(238,116,199);
if fl1 then begin {Якщо є виділений фрагмент в таблиці,}
  Chart1.Series[0].Clear; {то очищуємо діаграму від попередніх даних.}
  if (a=c) then {Якщо виділеним є стовпець, то дані комірок виділяються у
  вигляді окремих секторів на круговій діаграмі}
  for i:=b to d do begin
    if StringGrid1.Cells[a,i]<>" then Begin
      Chart1.SeriesList.Series[0].Add(StrToFloat(StringGrid1.Cells[a,i]),
      StringGrid1.Cells[0,i],Crl[i]);
    end;
  end
end;
end;
end;

```

Параметри оператора *Chart1.SeriesList.Series[0].Add(a,b,c)* мають такий смисл: перший параметр означає чергове значення, що додається до діаграми (наприклад, перетворене з рядкової величини в дійсне число значення комірки таблиці *StrToFloat(StringGrid1.Cells[a,i])*), другий параметр задає відповідний напис в легенді діаграми (в даному випадку – це прізвище учасника тестування, записане в нульовому стовпчику таблиці *StringGrid1.Cells[0,i]*), третій параметр задає колір малювання відповідного сектора діаграми, який вибирається з масиву *Crl*.

Не менш цікавими для застосування в проектах є поєднання програмування з використанням мультимедійних засобів, засобів роботи з базами даних. Реалізований в Delphi механізм Borland Database Engine (BDE) забезпечує для програми, написаній на Pascal, простий і зручний доступ до баз даних незалежно від їх архітектур: локальної, файл-серверної, клієнт-серверної. Наявність в Delphi багатих засобів дозволяє значно розширити коло тем проєктів, що надаються учням для виконання.

Окрім цього, ознайомлення учнів з правилами програмування операцій опрацювання таблично поданої інформації, наприклад, про результати тестування групи учасників корисно тим, що сприяє глибшому розумінню методів подання і опрацювання інформації, які застосовуються в сучасному прикладному програмному забезпеченні. А наявність єдиної формалізованої мови (мови програмування) при виконанні програмного проєкту дозволяє не витратити час на засвоєння нової мови формалізації, а зосередитись на вивченні особливостей предмету проєктування, розширити коло задач для розв'язування, задовольнити творчі наміри учнів в одній програмі.

ЛІТЕРАТУРА

1. Єгорова М.Е. Створення умов особистісного зростання та розвитку учнів на уроках інформатики в умовах Довгінцівського ліцею на підставі використання особистісно зорієнтованої педагогічної парадигми через запровадження творчо-ігрових елементів // Інформатика та комп'ютерно-орієнтовані технології

- навчання: Збірник наукових праць Всеукраїнської науково-практичної конференції (м. Хмельницький, 16-18 травня 2001 року)/ Редкол. – К.: Педагогічна Думка. – 2001. С. 88-90.
2. Лернер И.Я. Дидактические основы методов обучения. – М.: Педагогика, 1981. – 186 с.