

МЕТОДИКА ВИКЛАДАННЯ ДИСЦИПЛІНИ «ОСНОВИ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ПРОГРАМУВАННЯ»

Постановка проблеми. Підготовка бакалаврів спеціальності 6.010104.01 «Професійна освіта. Електроенергетика» включає в себе об'єктно орієнтоване програмування (ООП), яке є основою викладання дисципліни «Основи інформаційних технологій та програмування». Для виконання курсових та дипломних робіт бакалаврів у межах цієї дисципліни необхідно подати матеріал, який включає в себе питання створення інструментальних засобів розробки тестових завдань, або підсистеми складання тестових завдань. Розглядаючи використання тестування, доцільно орієнтуватися на узагальнення досвіду його застосування для всіх дисциплін.

Широке застосування тестів у педагогічній практиці неможливе без існування якісних тестів. Важливу роль в їх створенні відіграють інструментальні засоби розробки тестових завдань. У нашій роботі розглядається методика викладання дисципліни «Основи інформаційних технологій та програмування» й розробка нових класів «Тестове запитання» (ТТQ) та «Тест» (ТTest), які можуть бути використані при організації створення системи тестів. Розробка класів і прикладу здійснюється засобами об'єктно орієнтованої системи Delphi 2010 [2],[3].

Аналіз останніх досліджень і публікацій. У роботі [1] розглядається оновлення змісту дисципліни «Основні напрямки розвитку в комп'ютерній галузі», де головним є об'єктно орієнтоване програмування. Спочатку програмування виникло та розвивалось як процедурне програмування, основою якого є алгоритм. Ускладнення завдань, розв'язуваних на ЕОМ, та збільшення їхніх масштабів, виявили недоліки процедурного підходу, пов'язані з надійністю програм, повторним використанням коду, супроводом тощо. В результаті аналізу цих проблем і методів на початку 90-х років минулого сторіччя з'явилася концепція об'єктно орієнтованого програмування. При розробці зовсім нових систем програмування використання ООП дає безліч переваг [2], [3].

Постановка завдання. У рамках зазначеної проблеми слід розглянути питання методики викладання дисципліни «Основи інформаційних технологій та програмування» і розробку нових класів, які можуть бути використані при розробці підсистеми складання тестів. Тестування, особливо в комп'ютерній формі, вирішує ті питання, що впливають із сучасних тенденцій навчання. Зокрема, за допомогою тестування можна забезпечити ріст самостійного вивчення курсу студентами, що передбачено законом про вищу школу. Тільки тестування забезпечує необхідну систематичність вивчення студентами різних дисциплін.

Виклад основного матеріалу. Аналіз вимог до властивостей нового класу.

Для підготовки бакалаврів у межах цієї дисципліни для спеціальності 6.010104.01 «Професійна освіта. Електроенергетика» необхідно подати матеріал, який включає в себе педагогічне тестування.

Оцінювання має відбуватися на використанні педагогічних тестів, тому що тести дають можливість одержати об'єктивну оцінку якості навчання.

До останнього часу тести використовували переважно ентузіасти. Поставлені задачі комп'ютеризації навчання по-новому порушують питання про тестування і дають йому новий імпульс. Комп'ютеризація навчального процесу практично не може існувати без тестування, вона є його вищою формою.

У цій роботі розглядається розробка нових класів, які можуть бути використані при розробці педагогічного тестування. Наведено приклад такого використання. Розробка класів і прикладу здійснена засобами системи ООП Delphi.

Комп'ютерне тестування відбувається на основі тестів – сукупності запитань і варіантів відповідей. Задачу підготовки тестів має виконувати незалежна підсистема складання тестових завдань.

Існують два основних типи тестових завдань:

– тестові завдання закритого типу – вибір правильної відповіді з декількох варіантів, які надані студенту;

– тестові завдання відкритого типу – студент самостійно формулює відповідь, тобто в постановці завдання немає можливих варіантів відповіді.

Тестові завдання закритого типу складаються з основи (текст завдання з ілюстраціями, графіками, діаграмами тощо) і декількох (зазвичай 4–5) варіантів відповіді. Виконання завдання закритого типу полягає у виборі правильної відповіді з наданих варіантів. Завдання відкритого типу не використовують готові варіанти відповідей. Вони пропонують доповнити або написати відсутнє слово (групу слів, формулу, число, знак тощо), які свідчать про якість знань студента.

Основним типом завдань при комп'ютерному тестуванні є завдання закритого типу. Передбачити всі можливі види написання правильної відповіді при відкритому типі завдання просто неможливо, а отже, неможливо перевірити її за допомогою комп'ютера.

Таким чином, існує декілька типів тестових завдань, які відрізняються один від одного способом формулювання запитання і варіантів відповідей. Загальну класифікацію цих типів наведено на рис. 1.



Рис.1. Класифікація тестових завдань

Таким чином, розробник тесту (викладач) за допомогою підсистеми складання тестових завдань має можливість введення тексту завдання (запитання), варіантів відповідей на нього, вказувати правильну відповідь. Доцільно ввести новий клас «Тестове запитання» (TTQ) з такими властивостями і методами.

Властивості класу «Тестове запитання» – немає.

Методи класу «Тестове запитання»:

1. GetQuestion – читання запитання. Це функція, яка повертає рядкове значення з запитанням.

2. SetQuestion(Q: string) – запис запитання. Це процедура, яка встановлює запитання.

3. GetAnswer(i:integer) – читання *i*-го варіанту відповіді. Це функція, яка повертає рядкове значення з *i*-м варіантом відповіді.

4. SetAnswer(i:integer;A:string; r:boolean; out Er:boolean) – запис *i*-го варіанту відповіді. Ця процедура встановлює значення рядку A як *i*-й варіант відповіді. Параметр r вказує, чи є цей варіант відповіді правильним. Параметр Er, що повертається, вказує, чи було звернення до цього методу помилковим.

5. ShowQuestion(L:TLabel) – відображення запитання в мітці користувача L.

6. ShowAnswer(i:integer;L:TLabel; out Er:boolean) – відображення *i*-го варіанту відповіді в мітці користувача L. Параметр Er вказує, чи було звернення до цього методу помилковим.

Доцільно ввести новий клас «Тест» (TTest) із такими властивостями і методами.

Властивості класу «Тест»:

Count – кількість запитань у тесті. Режим доступу – тільки читання.

Методи класу «Тест»:

1. Add (Q:TTQ) – додавання тестового запитання до тесту.

2. Get (i:integer; out Q:TTQ) – читання *i*-го тестового запитання з тесту.

3. Open (FN:string) – читання тесту з файлу. Тут FN (File Name) – змінна з ім'ям файлу, який містить тест.

4. Save (FN:string) – зберігання тесту в файлі з ім'ям, яке задано у змінній FN.

Тест остаточно зберігається у вигляді окремого файлу, який потім використовується підсистемою тестування. На підставі аналізу вимог до функціонування підсистеми складання тестових завдань побудовано об'єктну модель цієї підсистеми. Визначено два класи – «Тестове запитання» і «Тест», за допомогою яких можна здійснити програмну реалізацію підсистеми складання тестових завдань. Вона моделює основні функції, які має виконувати така підсистема. Зрозуміло, що насправді така підсистема розробляється групою програмістів тривалий час.

На рис. 2 наведено форму (вікно) додатку.

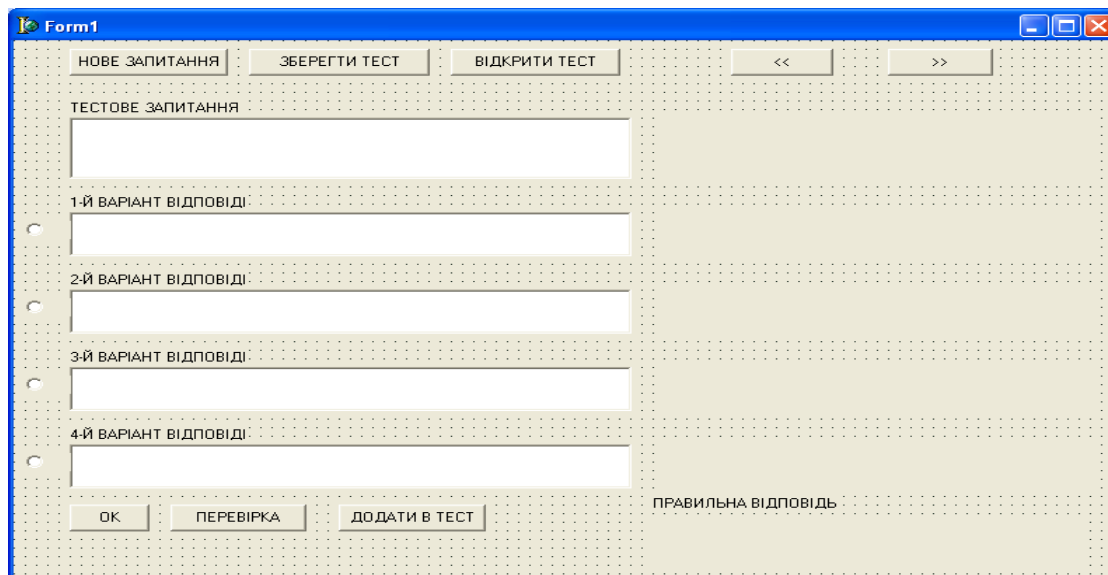


Рис.2. Форма (вікно) прикладного додатку

Призначення компонентів форми.

Поля «Тестове запитання», «1-й варіант відповіді» ... «4-й варіант відповіді» призначені для введення тексту, тестового запитання та варіантів відповідей. Компоненти Options поруч з варіантами відповідей слугують для визначення правильної відповіді.

Командні кнопки

1. «Нове запитання» – очищує відповідні поля форми і утворює новий об'єкт класу «Тестове запитання». Відповідний програмний код:

```
procedure TForm1.Button5Click(Sender: TObject);
```

```
begin
```

```
    NewQuestion;
```

```
TQ:=TTQ.Create;
```

```
end;
```

2. Кнопка «ОК». Ця кнопка призначена для підтвердження розробником тесту того, що тестове завдання остаточно сформовано. Відповідний програмний код:

```
procedure TForm1.Button1Click(Sender: TObject);
var
  Er,r:boolean;
begin
  if Edit1.Text="" then
  begin
    ShowMessage('ВІДСУТНЕ ЗАПИТАННЯ');
    EXIT;
  end;
  if Edit2.Text="" then
  begin
    ShowMessage('ВІДСУТНІЙ 1-Й ВАРІАНТ ВІДПОВІДІ');
    EXIT;
  end;
  if Edit3.Text="" then
  begin
    ShowMessage('ВІДСУТНІЙ 2-Й ВАРІАНТ ВІДПОВІДІ');
    EXIT;
  end;
  if Edit4.Text="" then
  begin
    ShowMessage('ВІДСУТНІЙ 3-Й ВАРІАНТ ВІДПОВІДІ');
    EXIT;
  end;
  if Edit5.Text="" then
  begin
    ShowMessage('ВІДСУТНІЙ 4-Й ВАРІАНТ ВІДПОВІДІ');
    EXIT;
  end;
  if (not RB1.Checked) and (not RB2.Checked) and
    (not RB3.Checked) and (not RB4.Checked) then
  begin
    ShowMessage('НЕ ВИЗНАЧЕНО ПРАВИЛЬНИЙ ВАРІАНТ ВІДПОВІДІ');
    EXIT;
  end;
  TQ.SetQuestion(Edit1.Text);
  TQ.SetAnswer(1,Edit2.Text,RB1.Checked,Er);
  TQ.SetAnswer(2,Edit3.Text,RB2.Checked,Er);
  TQ.SetAnswer(3,Edit4.Text,RB3.Checked,Er);
  TQ.SetAnswer(4,Edit5.Text,RB4.Checked,Er);
  ShowMessage('ТЕСТОВЕ ЗАПИТАННЯ СФОРМОВАНО');
  end;
```

3. Кнопка «Перевірка». Ця функція призначена для незалежної перевірки правильності формування тестового завдання і роботи методів ShowQuestion та ShowAnswer. Відповідний програмний код:

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

```

var
  i, m: integer;
  S:string;
  Er:boolean;
begin
  TQ.ShowQuestion(IQuest);
  TQ.ShowAnswer(1,lAnsw1,Er);
  TQ.ShowAnswer(2,lAnsw2,Er);
  TQ.ShowAnswer(3,lAnsw3,Er);
  TQ.ShowAnswer(4,lAnsw4,Er);
  for i:=1 to 4 do
  begin
    S:=TQ.GetAnswer(i);
    m:=length(S);
    if ord(S[m])=7 then
    begin
      IRigthAnswer.Caption:=copy(S,1,m-1);
      exit;
    end;
  end;
end;

```

4. Кнопка «Додати в тест» – призначена для додавання тестового запитання до тесту.
Відповідний програмний код:

```

procedure TForm1.Button3Click(Sender: TObject);
begin
  Test.Add(TQ);
  ShowMessage('ТЕСТОВЕ ЗАПИТАННЯ ДОДАНО ДО ТЕСТУ');
end;

```

5. Кнопка «Зберегти тест». Призначена для зберігання тесту в окремому файлі. За замовчанням ім'ям файлу є MyTest.txt. Відповідний програмний код:

```

procedure TForm1.Button4Click(Sender: TObject);
var
  s:string;
  f:TextFile;
  i:integer;
begin
  Test.Save('MyTest.txt');
  ShowMessage('ТЕСТ ЗБЕРЕЖЕНО');
end;

```

6. Кнопка «Відкрити тест». Призначена для завантаження тесту з файлу для подальшої роботи з ним. За замовчанням ім'ям файлу є MyTest.txt. Відповідний програмний код:

```

procedure TForm1.Button6Click(Sender: TObject);
begin
  Test.Open('MyTest.txt');
  ShowMessage('ТЕСТ ВІДКРИТО. КІЛЬКІСТЬ ТЕСТОВИХ ЗАПИТАНЬ =
'+IntToStr(Test.Count));
  CurQuest:=1;
  ShowTest;
  end;

```

7. Кнопки «<<<» та «>>>». Призначені для перегляду запитань у тесті. Кнопка «<<<» – попереднє запитання. Кнопка «>>>» – наступне запитання. Відповідний програмний код:

```

procedure TForm1.Button7Click(Sender: TObject);
begin
  CurQuest:=CurQuest-1;
  if CurQuest<=0 then CurQuest:=1;
  ShowTest;
end;
procedure TForm1.Button8Click(Sender: TObject);
begin
  CurQuest:=CurQuest+1;
  if CurQuest>Test.Count then CurQuest:=Test.Count;
  ShowTest;
end;

```

Крім того, розробимо такі допоміжні процедури:

```

procedure TForm1.FormCreate(Sender: TObject);
begin
  TQ:=TTQ.Create;
  Test:=TTest.Create;
  NewQuestion;
end;
procedure TForm1.NewQuestion;
begin
  Edit1.Text:="";
  Edit2.Text:="";
  Edit3.Text:="";
  Edit4.Text:="";
  Edit5.Text:="";
  RB1.Checked:=false;
  RB2.Checked:=false;
  RB3.Checked:=false;
  RB4.Checked:=false;
end;
procedure TForm1.ShowTest;
var
  Er:boolean;
  i,m:integer;
  S:string;
begin
  Test.Get(CurQuest,TQ);
  TQ.ShowQuestion(IQuest);
  TQ.ShowAnswer(1,IAnsw1,Er);
  TQ.ShowAnswer(2,IAnsw2,Er);

```

```

TQ.ShowAnswer(3,1Answ3,Er);
TQ.ShowAnswer(4,1Answ4,Er);
for i:=1 to 4 do
begin
  S:=TQ.GetAnswer(i);
  m:=length(S);
  if ord(S[m])=7 then
  begin
    IRigthAnswer.Caption:=copy(S,1,m-1);
    exit;
  end;
end;
end;
end;

```

Для використання нових класів в оператор uses додано відповідне посилання:

uses Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, ClassTest;

Після запуску цього додатку введено запитання, варіанти відповідей та вказано правильну відповідь (рис. 3).



Рис.3. Введення тестового запитання

Після введення запитання і варіантів відповіді натиснути кнопку «ОК» (підтвердження формування тестового завдання). Отримано відповідне повідомлення (рис. 4).

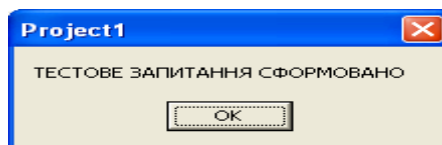


Рис.4. Підтвердження формування тестового запитання

Для незалежної перевірки тестового завдання натиснути кнопку «Перевірка». Отримано результат (рис. 5):

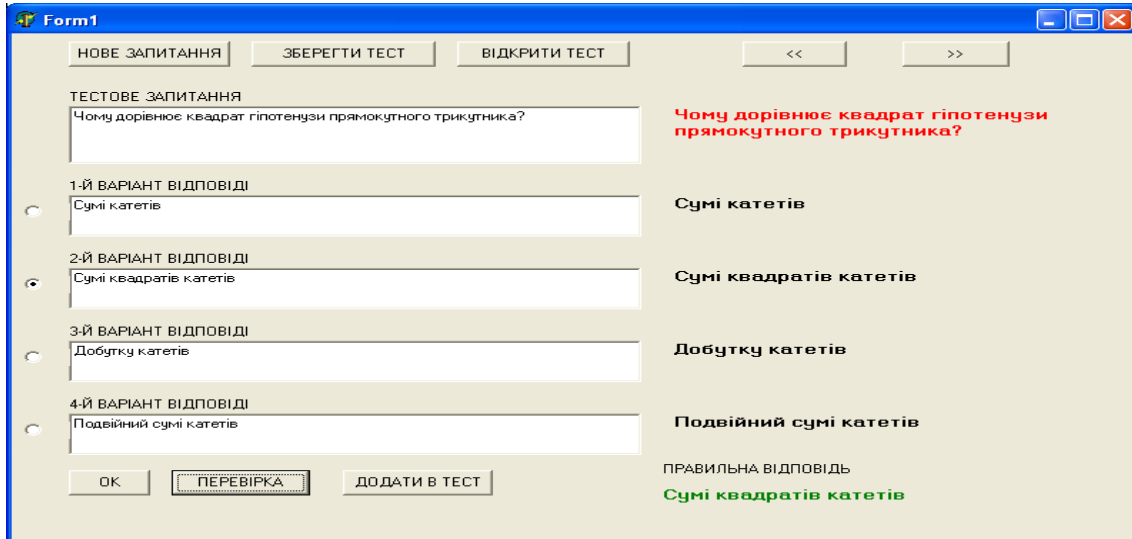


Рис.5. Результат перевірки тестового завдання

Незалежна перевірка підтвердила, що тестове завдання сформовано правильно. При натисканні кнопки «Додати в тест», отримано відповідне повідомлення (рис. 6).

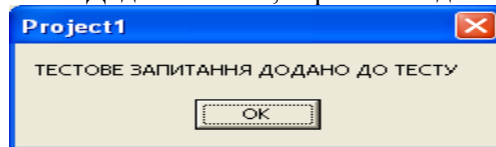


Рис.6. Тестове запитання додано до тесту

Для зберігання тесту (поки що з одного запитання) слід натиснути кнопку «Зберегти тест» та отримати відповідне повідомлення (рис. 7).

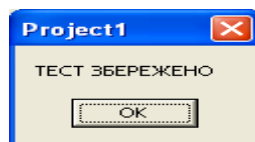


Рис.7. Тест збережено (за замовчанням у файлі MyTest.txt)

Аналогічно необхідно ввести ще три тестових завдання і тест за замовчанням зберегти у файлі MyTest.txt.

Тестування класу «Тест» частково вже відбулося під час збереження тесту у файлі (метод Save). Для перевірки методу Open розроблений додаток треба закрити та завантажити знову. Натиснувши кнопку «Відкрити тест», отримано таке повідомлення (рис.8):

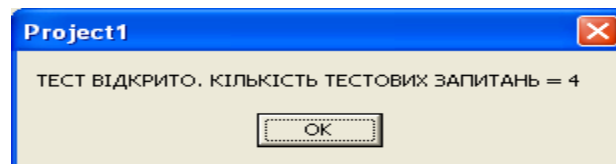


Рис.8. Повідомлення про відкриття тесту

Зміст повідомлення відповідає очікуванням – тест відкрито разом із кількістю запитань у тесті. Слід перевірити запитання, які знаходяться у тесті. Після натискання на кнопку «ОК» цього повідомлення буде отримано перше тестове запитання, варіанти відповідей та правильну відповідь (рис.9).

Рис.9. Перше запитання тесту

Перевірити наступне запитання можна натиснувши на кнопку «>>». Результат буде такий (рис.10):

Рис.10. Друге запитання тесту

Подальша перевірка тесту шляхом довільного натискання кнопки «>>» та кнопки «<<<» показує, що тест сформовано у відповідності з очікуванням (рис. 11).

Рис.11. Четверте запитання тесту

Висновки. Методи класу «тест» працюють без помилок. Тобто розроблено прикладний додаток, який тестує властивості і методи нових класів, а також демонструє їх можливості для розробки підсистеми складання тестових завдань.

Перспективи подальших досліджень. Подальші дослідження будуть спрямовані на вивченні проблем методики викладання дисципліни «Основи інформаційних технологій та програмування» прикладного об'єктно орієнтованого програмування для інженерно-педагогічних спеціальностей із використанням інноваційних методів навчання бакалаврів - майбутніх висококваліфікованих спеціалістів, а також утворення нових класів, які можуть бути використані для розробки системи складання тестових завдань.

Список використаних джерел

1. Бондаренко М.А. Оновлення змісту дисципліни «Основні напрямки розвитку в комп'ютерній галузі» / М. А. Бондаренко // Проблеми інженерно-педагогічної освіти : зб. наук. пр. / Укр. інж.-пед. акад. – Х., 2006. – Вип. 13. – С. 28–35.
2. Бондаренко М. А. Програмування у середовищі Delphi : навч. посіб. для вищ. навч. закладів / М. А. Бондаренко ; Укр. інж.-пед. акад. – Х. : ФОП Лібуркіна Л.М., 2007. – 600 с.
3. Бондаренко М. А. Основи інформаційних технологій та програмування : підруч. для вищ. навч. закладів / М. А. Бондаренко ; Укр. інж.-пед. акад. – Х. : ФОП Павленко О.Г., 2010. – 536 с.

Бондаренко М.А.

Методика викладання дисципліни «Основи інформаційних технологій та програмування»

Розглядається питання методики викладання бакалаврам спеціальності 6.010104.01 «Професійна освіта. Електроенергетика» дисципліни «Основи інформаційних технологій та програмування». На підставі аналізу вимог до функціонування підсистеми складання тестових завдань побудовано об'єктну модель цієї підсистеми. Визначено два класи – «Тестове запитання» і «Тест», за допомогою яких можна здійснити програмну реалізацію підсистеми складання тестових завдань. Здійснено програмну реалізацію класів, які можуть бути використані для розробки підсистеми складання тестових завдань.

Ключові слова: об'єктно орієнтоване програмування, об'єкт, клас, методи класу, властивості класу, прикладний додаток, тестування класу.

Бондаренко Н.А.

Методика преподавания дисциплины «Основы информационных технологий и программирование»

Рассматривается вопрос методики преподавания бакалаврам специальности 6.010104.01 «Профессиональное образование. Электроэнергетика» дисциплины «Основы информационных технологий и программирования». На основании анализа требований к функционированию подсистемы составления тестовых заданий построена объектная модель этой подсистемы. Определенно два класса – «Тестовый вопрос» и «Тест», с помощью которых можно осуществить программную реализацию подсистемы составления тестовых заданий. Осуществлена программная реализация классов, которые могут быть использованы для разработки подсистемы составления тестовых заданий.

Ключевые слова: объектно ориентированное программирование, объект, класс, методы класса, свойства класса, прикладное дополнение, тестирование класса.

M. Bondarenko

Methods of Teaching Discipline «Principles of Information Technologies and Programming»

The issue is considered of teaching methods for the discipline «Principles of Information Technologies and Programming» being taught to the Bachelors of the speciality 6.010104.01

“Vocational Training Electrical Engineering”. An object model of the subsystem of compiling text assignments has been constructed that is based on the demands to this system functioning. The classes – “Text question” and “Test” have been defined that make possible the system programme realization while compiling text assignments. A programme class realization has been made which can be made use of while developing a subsystem for compiling test assignments.

Key words: object oriented programming, object, class, class methods, applied supplement, testing the class.

Стаття надійшла до редакції 09.02.2011 р.