

МЕТОДИКА РОЗРОБКИ КЛАСУ «ТЕСТ» ДЛЯ ПІДСИСТЕМИ УПРАВЛІННЯ ТЕСТОВИМИ ЗАВДАННЯМИ

Постановка проблеми. Підготовка бакалаврів спеціальності 6.010104.36 «Професійне навчання. Комп'ютерні технології в управлінні та навчанні» включає в себе розробку нового класу «Тест» у середовищі об'єктно орієнтованого програмування (ООП) Delphi. Цей клас може бути використаний при створенні автоматизованих систем контролю якості знань.

Істотною особливістю системи Delphi є можливість створення власних класів [1]. Як відомо, в основі ідеології об'єктно орієнтованого програмування лежить підхід, при якому програмування зводиться до опису об'єктів і визначенню їхніх властивостей, методів і подій. Такими об'єктами є, наприклад, стандартні елементи керування в середовищі Delphi – мітки, текстові поля, списки, командні кнопки тощо. Прикладна програма створюється шляхом завдання необхідних об'єктів і визначенням їхньої взаємодії між собою та операційною системою.

Кожен об'єкт, що існує в програмі, має свої властивості. Однак окремі об'єкти можуть мати загальні ознаки, поведження і події. Такі об'єкти поєднуються в класи. Інакше кажучи, клас – це деяка абстракція. У ньому ніколи не задаються конкретні значення всіх полів і властивостей. Навіть методи в деяких випадках можуть не мати жодного рядка програмного коду. Призначення класу – бути шаблоном (зразком) для створення об'єктів цього класу, тобто таких самих структур, як і сам клас, але вже з конкретними значеннями всіх членів класу. Тобто об'єкт – це така сама структура, як і клас, але в якій визначені конкретні значення всіх його компонентів.

Наприклад, усі командні кнопки є об'єктами класу TButton, усі мітки – об'єктами класу TLabel. Клас відрізняється від елемента керування тим, що об'єкти класу можуть реально існувати в програмі без візуального відображення [2], [3], [4].

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. Програмування виникло та розвивалося як процедурне програмування. Процедурний підхід припускає, що основою програми є алгоритм, тобто деяка процедура обробки даних. 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 багаторічного досвіду та аналізу цих проблем і методів на початку 90-х років минулого сторіччя з'явилася концепція об'єктно орієнтованого програмування.

Програми, розроблені старими методами, без використання об'єктно орієнтованого підходу, не є «неправильними». У більшості випадків вони прекрасно працюють, і нічого в них міняти не треба. А от при розробці зовсім нових систем програмування використання ООП може дати безліч переваг [2], [3], [4]. Тому, незважаючи на те, що за останні 10 років об'єктно орієнтовані технології розвивалися досить швидко та стабільно, особливий поштовх їхньому розвитку дала поява Інтернету. Він відкрив перед ООП область зовсім нових розробок, без необхідності використання старих методів і засобів. Сьогодні більшість баз даних та Web-технологій є за своєю природою об'єктно орієнтованими.

Постановка завдання. У рамках зазначеної проблеми слід розглянути питання методики розробки нового класу «Тест» у середовищі об'єктно орієнтованого програмування Delphi. Цей клас може бути використаний при створенні відповідних баз даних і автоматизованих систем контролю якості знань із різних дисциплін навчального плану. Головний елемент будь-якої бази даних – це запис, який містить усю необхідну інформацію про один об'єкт. Сама база даних розглядається як сукупність записів, тобто сукупність інформації про об'єкти.

У рамках зазначеної проблеми слід розглянути питання методики розробки нового класу TTest засобами системи Delphi.

Виклад основного матеріалу. Надалі в роботі ми будемо використовувати таку термінологію.

Тестове завдання (або тестове запитання) – одне завдання або одне запитання, яке студент має виконати або дати відповідь у процесі тестування. В даному випадку терміни «тестове завдання» є більш широким, ніж «тестове запитання», оскільки запитання завжди можна переформулювати як завдання (замість запитання «Чому дорівнює площа прямокутника?» можна дати завдання «Укажіть, чому дорівнює площа прямокутника»). Тому надалі ми будемо використовувати саме термін «тестове завдання», маючи на увазі, що воно може бути сформульовано й у формі запитання.

Тест – набір тестових завдань, які має виконати студент.

При роботі з такою структурою даних, як тест викладач кожен раз має вирішувати питання відкриття і перевірки файлів, які містять тексти тестових запитань і варіантів відповідей на них, розміщення варіантів у випадковому порядку.

Така задача може бути зручно вирішена шляхом створення спеціального класу «Тест», який би взяв на себе всю цю «рутинну роботу». Тобто для відображення завдань з тесту студенту буде достатньо створити новий об'єкт цього класу, задати необхідні властивості і застосувати відповідні методи для відображення інформації.

Об'єкт нового класу «Тест» має містити, по-перше, самі тестові запитання і варіанти відповідей до них.

По-друге, необхідно визначити файл, у якому містяться ці тестові завдання.

По-третє, для того, щоб викладач міг управляти тестовими завданнями, тест має містити інформацію про кількість своїх завдань.

Це – мінімальна сукупність даних, яку має містити об'єкт класу «Тест». Питання полягає в тому, чи має об'єкт класу «Тестове запитання» містити в собі інформацію, який із варіантів відповіді є правильною відповіддю.

Необхідно враховувати, що автономне існування змісту тесту у вигляді окремого файлу, в якому в тому чи іншому вигляді міститься інформація про правильні відповіді, може призвести до спроб його «злому» й отримання цієї інформації до початку тестування.

Запобігти цьому можна, на наш погляд, двома шляхами:

а) кодуванням інформації. Не беремося судити, наскільки надійним буде такий захід – це справа фахівців із питань захисту інформації;

б) відокремлення змісту тестового завдання від інформації про правильні відповіді. В самому тесті (файлі) така інформація не міститься. Правильність відповідей може бути перевірена на основі окремої інформації (файлу), яка, наприклад, вводиться викладачем на початку тестування. Або взагалі оцінка результатів може здійснюватися після тестування. Доступ до файлу з інформацією про правильні відповіді мають тільки повноважні особи.

Надалі припустимо такі обмеження. Кількість запитань в тесті – не більш 10, варіантів відповідей у кожному завданні – завжди 4.

Таким чином, новий клас має мати такі властивості:

1) Масив тестових запитань та варіантів відповідей (Zavd).

Для визначення типу цієї властивості введемо такі допоміжні типи.

type

sAR4 = array [1..4] of string;

type

TZavd = record

Question:string;

Vars = sAR4;

end;

Тоді тип властивості Zavd можна визначити як array [1..10] of TZavd.

2) Count – кількість завдань у тесті. Тип даних – integer.

3) FileName – ім'я файлу, який містить тестові запитання і 4 варіанти відповіді для кожного з них.

Властивість Count має доступ тільки для читання. Властивість FileName має повний доступ. При наданні йому певного значення має бути виконано перевірку на існування файлу та можливість його відкриття. Властивість Zavd має доступ до своїх даних через відповідні методи. Значення цієї властивості визначаються тільки через дані у відповідному файлі.

Усі дані, які задані властивостями класу, мають бути відображені на сторінці (формі). При цьому тільки розробник прикладного додатку (тобто користувач класу) має визначати, де саме на сторінці і в якому вигляді буде розміщена ця інформація.

Таким чином, новий клас буде мати такі методи:

1) ShowQuestion – відображає тестове запитання на формі. Має два параметри:
– індекс, тобто номер запитання в масиві Zavd. Тип параметру – integer.
– компонент на формі (сторінці), де має бути відображена ця інформація. Тип параметру – TLabel;

2) ShowVar1 – відображає текст першого варіанту відповіді на сторінці. Має два параметри:

– індекс, тобто номер запитання в масиві Zavd. Тип параметру – integer.
– компонент на формі (сторінці), де має бути відображена ця інформація. Тип параметру – TLabel;

3) ShowVar2 – відображає текст другого варіанту відповіді на сторінці. Має два параметри:

– індекс, тобто номер запитання в масиві Zavd. Тип параметру – integer.
– компонент на формі (сторінці), де має бути відображена ця інформація. Тип параметру – TLabel;

4) ShowVar3 – відображає текст третього варіанту відповіді на сторінці. Має два параметри:

– індекс, тобто номер запитання в масиві Zavd. Тип параметру – integer.
– компонент на формі (сторінці), де має бути відображена ця інформація. Тип параметру – TLabel;

5) ShowVar4 – відображає текст четвертого варіанту відповіді на сторінці. Має два параметри:

– індекс, тобто номер запитання в масиві Zavd. Тип параметру – integer.
– компонент на формі (сторінці), де має бути відображена ця інформація. Тип параметру – TLabel;

Застосування разом усіх цих методів і властивостей не є обов'язковим і може відбуватися в будь-якій допустимій комбінації. Це зручно у випадку, коли варіанти відповіді необхідно відображати у вільному порядку.

Крім того, використання властивостей та методів нового класу ніяк не обмежує розробників прикладного додатку в тому, де саме на сторінці буде відображена ця інформація і в якому вигляді. Все це визначає сам розробник при проектуванні дизайну сторінки.

Такий підхід дає гнучкі можливості з ефективного використання цього класу.

У роботі розроблено програмний код класу TTest («Тест») мовою Delphi. Розробка була здійснена в середовищі Delphi.

Для тестування класу була розроблена окрема програма. До відповідного проекту в Delphi було додано розроблений модуль (unit) з класом TTest.

Вигляд форми для тестування класу надано на рис. 1.

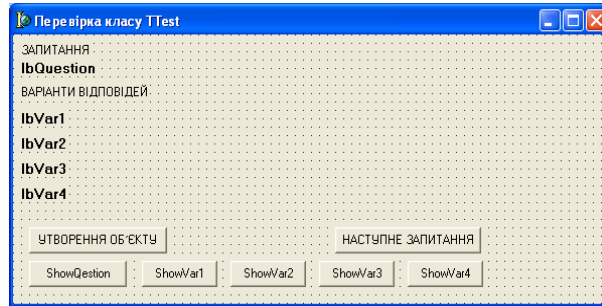


Рис. 1. Вигляд форми для тестування класу

Призначення компонентів:

lbQuestion – мітка для відображення тексту запитання;

lbVar1 – lbVar4 – мітки для відображення варіантів відповіді.

Командні кнопки:

«Утворення об'єкту» – для перевірки самого факту утворення об'єкта класу TTest та задання його властивості FileName;

«ShowQuestion» – для перевірки роботи методу ShowQuestion;

«ShowVar1» – «ShowVar4» – для перевірки роботи методів ShowVar1 – ShowVar4.

Початок програми для тестування має такий код:

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, TestClass, ExtCtrls, ComCtrls;
type
  TForm1 = class(TForm)
    lbQuestion: TLabel;
    lbVar1: TLabel;
    lbVar2: TLabel;
    lbVar3: TLabel;
    lbVar4: TLabel;
    Label1: TLabel;
    Label2: TLabel;
    Button2: TButton;
    Button1: TButton;
    Button3: TButton;
    Button4: TButton;
    Button5: TButton;
    Button6: TButton;
    Button7: TButton;
    procedure Button2Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure Button5Click(Sender: TObject);
    procedure Button6Click(Sender: TObject);
    procedure Button7Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

```

```

end;
var
  Form1: TForm1;
  T: TTest; // Об'єкт класу «Тест»
  NumZap: integer=1; // Номер запитання
implementation
  {$R *.dfm}

```

Процедура для тестування утворення об'єкта та призначення йому властивостей має такий код:

```

procedure TForm1.Button2Click(Sender: TObject);
begin
  {Перевірка утворення об'єкта класу TTest та читання його властивостей}
  T:= TTest.Create;
  T.FileName:='test.txt';
  ShowMessage('ОБ'ЄКТ УТВОРЕНО'+#13+'КІЛЬКІСТЬ ЗАВДАНЬ В ТЕСТІ = '
  +IntToStr(T.Count));
end;

```

Тестування відбувається в такий спосіб. На початку утворюється новий об'єкт P за допомогою оператора T:= TTest.Create.

Далі властивості FileName призначається конкретне значення 'test.txt'. Це ім'я файлу, який містить тестове запитання та варіанти відповідей на нього. Зміст цього файлу показано на рис. 2.

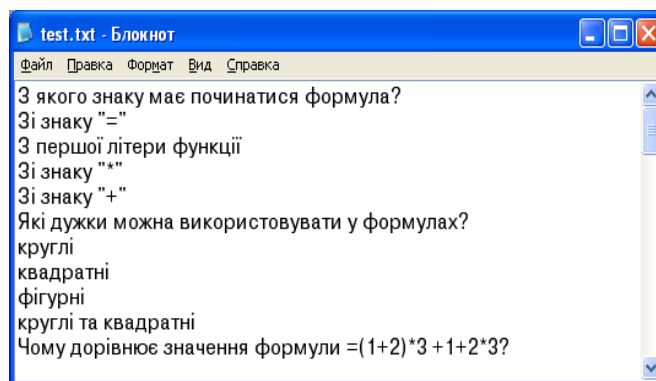


Рис. 2. Зміст файла з тестовим запитанням та варіантами відповіді

Після запуску цієї програми та натискання на кнопку «Утворення об'єкта» було отримано повідомлення про утворення об'єкта та кількість завдань у тесті (рис. 3).

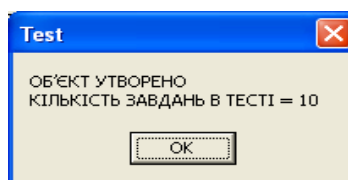


Рис. 3. Повідомлення про утворення об'єкта

Тобто відбулося звернення до властивості Count і отримано вірне значення.

Після цього в програмному коді оператор T.TextFileName:='test.txt' було замінено на оператор T.TextFileName:=' test1.txt', тобто зроблена спроба призначити властивості FileName помилкове значення (такий файл не існує).

Після запуску програми та другого натискання на кнопку «Утворення об'єкта» було отримано таке повідомлення (рис. 4).

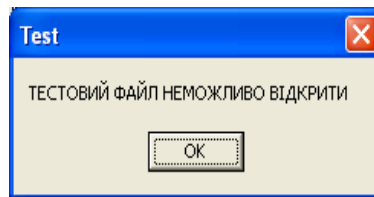


Рис. 4. Реакція класу на спробу призначити помилкове значення

Таким чином, процедура призначення властивості FileName правильно відреагувала на помилкове значення.

Процедури для тестування методів ShowQuestion, ShowVar1 – ShowVar4 мають такий код:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
    T.ShowQuestion(NumZap,lbQuestion);
end;
procedure TForm1.Button3Click(Sender: TObject);
begin
    T.ShowVar1(NumZap, lbVar1);
end; та інші.
```

Після запуску програми та натискання на кнопку «Утворення об'єкта» була натиснута кнопка «ShowQuestion». Результат представлено на рис. 5.

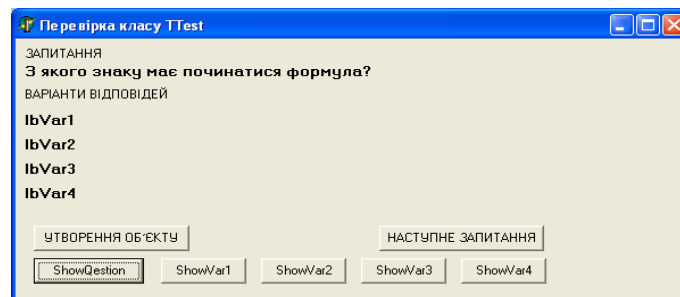


Рис. 5. Результат тестування методу ShowQuestion

Таким чином, метод ShowQuestion спрацював відповідно до вимог. Далі послідовно були натиснуті кнопки ShowVar1 – ShowVar4/ Результат представлено на рис. 6.

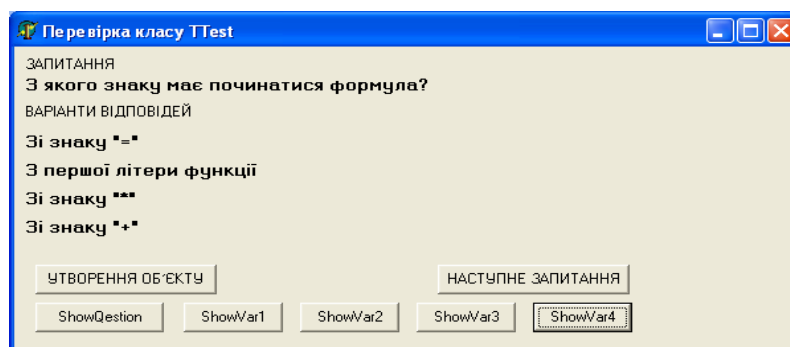


Рис. 6. Результат тестування методів ShowVar1 – ShowVar4

Таким чином, ці методи спрацювали відповідно до вимог.

Далі було перевірено звернення до будь-якого запитання в тесті. Для цього для командної кнопки «Наступне питання» був введений такий код:

```
procedure TForm1.Button7Click(Sender: TObject);
begin
  NumZap:=NumZap+1;
  if NumZap>T.Count then exit;
  label1.Caption:='ЗАПИТАННЯ № '+IntToStr(NumZap);
  T.ShowQuestion(NumZap,lbQuestion);
  T.ShowVar1(NumZap, lbVar1);
  T.ShowVar2(NumZap, lbVar2);
  T.ShowVar3(NumZap, lbVar3);
  T.ShowVar4(NumZap, lbVar4);
end;
```

Тут спочатку номер запитання збільшується на 1. Якщо це значення не перевищує кількості запитань у тесті, то до тесту застосовуються всі методи для відображення запитання і варіантів відповіді.

Після натискання на цю кнопку було отримано такий результат (рис. 7).

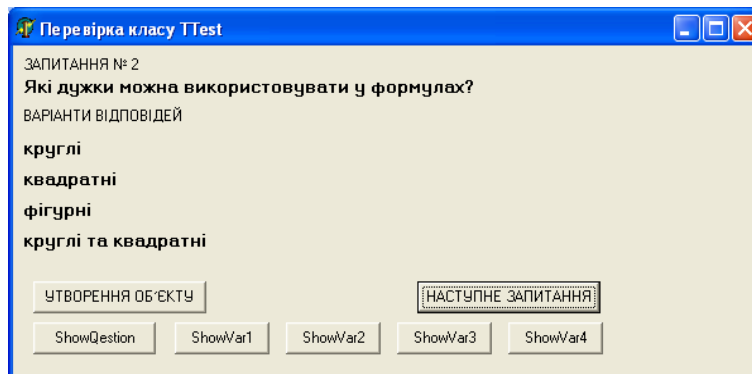


Рис. 7. Перехід на друге питання

Далі було перевірено відображення всіх запитань у тесті (рис. 8).

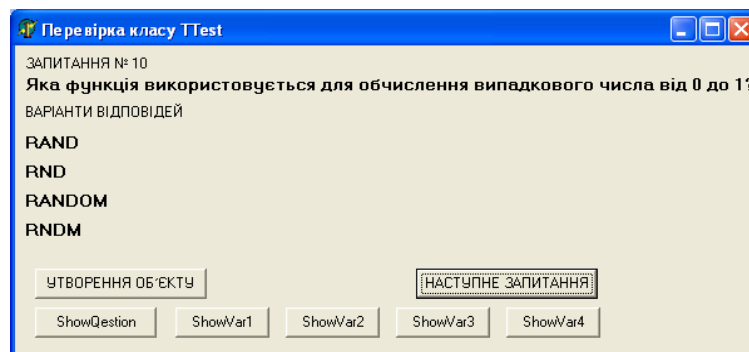


Рис. 8. Відображення останнього запитання

У цьому прикладі було розроблено прикладний додаток для відображення тесту з 10 запитань щодо роботи з формулами в редакторі електронних таблиць OpenOffice. Для відображення цих запитань було використано клас TTest. Було реалізовано послідовний перехід від одного запитання до наступного або попереднього. Також було реалізовано розташування варіантів відповіді у випадковому порядку.

Відповідна форма представлена на рис. 9.

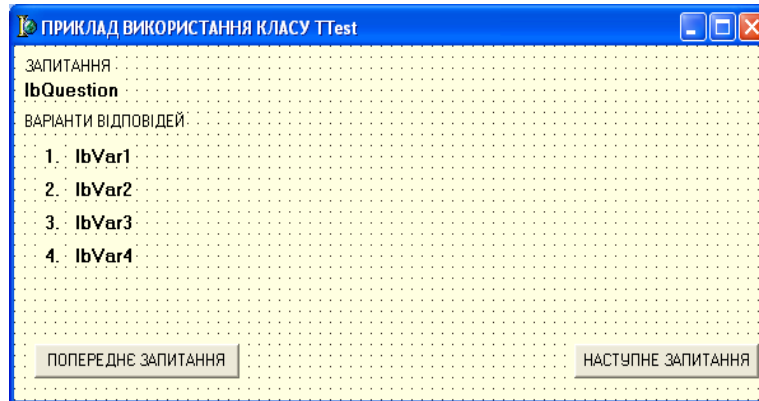


Рис. 9. Форма для прикладного додатку

Відповідний тестовий файл було попередньо скопійовано в ту ж саму папку, де знаходиться додаток. Файл має назву test.txt.

У змінній NumZap міститься номер поточного запитання (яке відображено на екрані). Перехід до попереднього (наступного) запитання здійснюється шляхом зменшення (збільшення) на 1 номеру змінної NumZap (з відповідною перевіркою) та відображення тестового запитання з цим номером за допомогою процедури ShowZap. Ця процедура реалізує відображення варіантів відповіді у випадковому порядку.

Після запуску цього додатку було отримано такий результат (рис. 10).

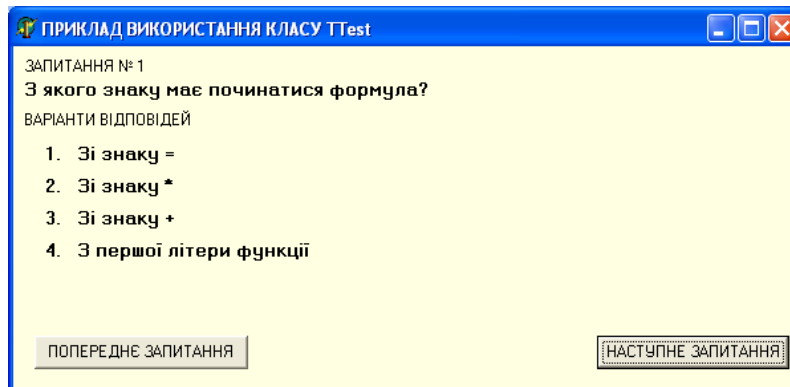


Рис. 10. Відображення першого запитання

Після натискання на кнопку «Попереднє питання» нічого не змінюється, оскільки попереднього питання не існує. Після натискання на кнопку «Наступне питання» на формі відображається друге тестове питання (рис. 11).

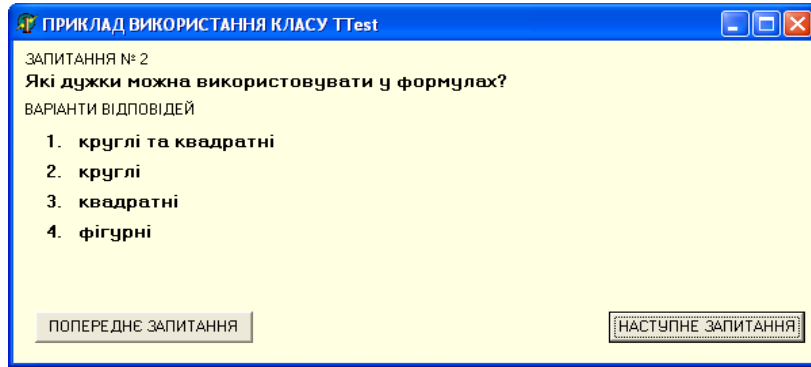


Рис. 11. Відображення другого тестового питання

Наступна перевірка показала, що натискання на кнопки переходу приводять до правильного переходу від одного питання до іншого.

Далі була натиснута кнопку «Попереднє питання». Отримано такий результат (рис. 12).

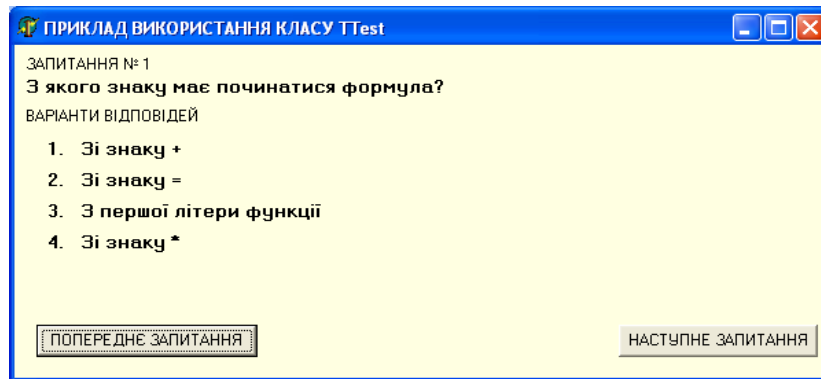


Рис. 12. Повернення до першого запитання

Аналогічним чином було перевірено відображення всіх завдань тесту.

Основні результати роботи:

1. Засобами середовища Delphi розроблена методика утворення нового класу TTest («Тест»).

2. Проведено перевірку роботи цього класу, яке показало правильність роботи властивостей та методів.

3. Розроблено прикладний додаток із використанням нового класу. Показано, що простота в програмуванні цього додатку досягнута саме за рахунок використання нового класу, тобто тих переваг, які дає об'єктно орієнтоване програмування.

Розроблено новий клас TTest («Тестове запитання»). Проведене тестування показало правильність роботи його властивостей та методів і можливість використання у прикладних додатках.

Утворений простий додаток наочно продемонстрував коректність роботи нового класу та переваги для розробників класу, які надає об'єктно орієнтоване програмування.

Перспективи подальших досліджень. Подальші дослідження будуть спрямовані на постійному вивченні проблем методики розробки нових класів TTest засобами системи Delphi.

Новий клас може бути корисним для розробників прикладних додатків, особливо для проектування комп'ютерного тестування в системах контролю якості знань.

На підставі аналізу вимог до об'єктів класу «Тестове запитання» визначено, які саме властивості і методи має мати цей клас. Це дає можливість почати безпосередню розробку нового класу на рівні програмного коду.

Підготовка бакалаврів спеціальності 6.010104.36 «Професійне навчання. Комп'ютерні технології в управлінні та навчанні» включає в себе розробку нового класу «Тест» для кожної дисципліни в середовищі об'єктно орієнтованого програмування Delphi. Цей клас може бути використаний при створенні автоматизованих систем контролю якості знань для кожної дисципліни спеціальностей Академії.

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 р. – Вип. 13. – С. 28–35.
2. Бондаренко М. А. Програмування у середовищі Delphi : навч. посіб. для вищ. навч. закл. / М. А. Бондаренко. – Х. : Канком, 2007. – 600 с.
3. Бондаренко М. А. Основні напрямки розвитку в комп'ютерній галузі : навч. посіб. для вищ. навч. закл. / М. А. Бондаренко. – Х.: ФОП Лібуркіна Л. М., 2007. – 696 с.
4. Бондаренко М. А. Основи інформаційних технологій та програмування : підруч. для вищ. навч. закл. / М. А. Бондаренко. – Х. :ФОП Павленко О. Г., 2010. – 536 с.

Бондаренко М. А.

Методика розробки класу «Тест» для підсистеми управління тестовими завданнями

Розглянуто методику розробки нового класу TTest засобами системи Delphi.

Методика процесу розробки класу «Тест» складається з таких етапів:

– Проектування. На цьому етапі визначаються властивості, методи і події нового класу.

– Програмування властивостей, методів і подій.

– Тестування і налагодження класу. Компілювання нового класу у файл з розширенням .dcu (Delphi compiled unit).

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

Бондаренко Н. А.

Методика разработки класса «Тест» для подсистемы управления тестовыми заданиями

Рассматривается методика разработки нового класса TTest средствами системы Delphi.

Методика процесса разработки нового класса состоит из таких этапов:

– Проектирование. На этом этапе определяются свойства, методы и события нового класса.

– Программирование свойств, методов и событий.

– Тестирование и отладка класса. Компилирование нового класса в файл с расширением .dcu (Delphi compiled unit).

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

N. Bondarenko

Methodology of Development of Class «Test» for the Subsystem of Management of Test Exercises

In this work the method of development of new class of TTest is examined by facilities of the system of Delphi.

The method of development of new class process consists of such stages:

– Planning. On this stage properties, methods and events of new class, are determined, coming from his functional setting.

– Programming of properties, methods and events.

– Testing and adjusting of class. Work of all methods, properties and events is checked up.

Key words: objective focused programming, object, a class, methods of a class, property of a class, the applied appendix, testing of a class.

Стаття надійшла до редакції 21.06.2012 р.