

Обробка інформації в складних організаційних системах

УДК 004.42

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

Українська інженерно-педагогічна академія, Харків

РОЗРОБКА ПРИКЛАДНОГО ДОДАТКУ ДЛЯ ТЕСТУВАННЯ КЛАСІВ В ПІДСИСТЕМІ ОБЛІКУ ПРОДАЖУ ТОРГІВЕЛЬНОГО ПІДПРИЄМСТВА

Розглядається розробка прикладного додатку, який дозволяє тестувати нові класи, що можуть бути використані для підсистеми обліку продаж торгівельного підприємства. Також наведено приклад тестування вище зазначених класів. Наведені основні результати тестування, які дали належні висновки щодо застосування в розробці прикладного додатку об'єктно-орієнтованого програмування. Розробка прикладу здійснена засобами системи ООП Delphi 2010.

Ключові слова: об'єктно-орієнтоване програмування, прикладний додаток, тестування класів.

Вступ

У [1] було розроблено нові класи, які можуть бути ефективно використані для підсистеми обліку продаж торгівельного підприємства. Проаналізовано ефективність розроблених класів і зроблено висновки, що розроблені класи можуть бути корисними для проектування автоматизованих систем бухгалтерського обліку для торгівельних підприємств.

Виходячи з цього, виникла задача перевірки на практиці роботи розроблених класів, перевірки достовірності вихідної інформації, правильності роботи підсистеми обліку продаж торгівельного підприємства в цілому.

Для цього пропонується розробка прикладного додатку, який дозволить в повній мірі протестувати розроблені класи підсистеми обліку.

Розробка прикладного додатку

Прикладний додаток, який було розроблено, моделює процес розробки справжньої підсистеми обліку продаж. Зрозуміло, що в повному обсязі така підсистема розробляється групою програмістів тривалий час.

Для зручності створення масиву об'єктів класу «Товар» (моделі бази даних) всі дані для визначення властивостей об'єктів цього класу було попередньо підготовлено в окремому текстовому файлі. Структура файлу така:

- 1-й рядок – код товару (вибрано 5 символів);
- 2-й рядок – найменування товару;
- 3-й рядок – одиниця виміру;
- 4-й рядок – ціна за одиницю;
- 5-й рядок – початкова кількість товару на початок продаж.

Наступні п'ять рядків містять дані про наступний товар і т.д.

Приклад такого файлу наведено на рис. 1. Всі дані – умовні.

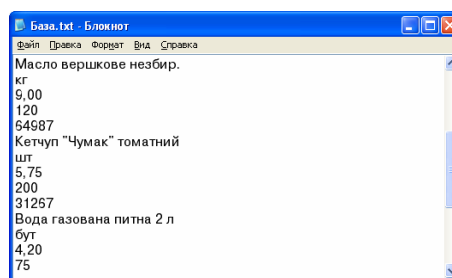


Рис. 1. Текстовий файл для тестування класу «Товар»

Тут базу даних товарів у продажу промодельовано у вигляді масиву об'єктів класу TТовар. Масив представлено у вигляді списку найменувань товарів. Для отримання повної інформації про кожний товар (в межах визначеної кількості властивостей) достатньо виділити його у цьому списку (натиснути на найменування товару за допомогою миші або перемістити курсор).

Відповідну форму для тестування наведено на рис. 2.

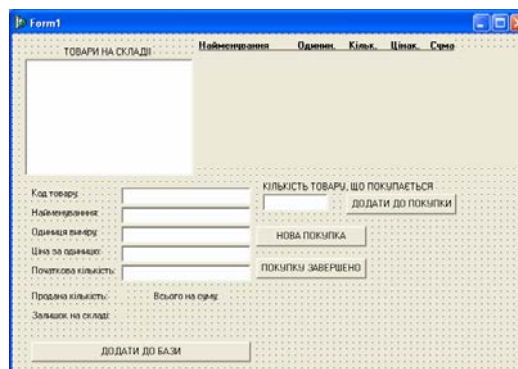


Рис. 2. Форма для тестування прикладного додатку

В операторі uses крім стандартних класів Delphi 6 додано розроблені класи ClassPokupka і ClassTovar.

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes,
  Graphics, Controls, Forms,
  Dialogs, StdCtrls, ClassPokupka, ClassTovar;
var
  Form1: TForm1;
  Tv: TTovar;
  Pok: TPokupka;
  Base: array[1..100] of TTovar; // Модель бази то-
варів
  NumTovars: integer;           // Кількість товарів
в бази
  T: TObject;
implementation
// Додавання товару до бази
procedure TForm1.Button1Click(Sender: TObject);
var
  tv: TTovar;
begin
  Tv:=TTovar.Create;
  tv.Kod:=edit1.Text;
  tv.Name:=edit2.Text;
  tv.Od:=edit3.Text;
  if edit4.Text<>"          then
tv.Price:=StrToFloat(edit4.Text) else tv.Price:=0;
  if edit5.Text<>"          then
tv.BeginKil:=StrToFloat(edit5.Text) else tv.BeginKil:=0;
  tv.ProdKil:=0;
  numtovars:=numtovars+1;
  base[numtovars]:=Tv;
  ListBox1.AddItem(Tv.Name,T);
end;
// Вибір товару зі списку
procedure TForm1.ListBox1Click(Sender: TObject);
var
  k: integer;
begin
  k:=ListBox1.ItemIndex+1;
  if k>NumTovars then exit;
  edit1.Text:=base[k].Kod;
  edit2.Text:=base[k].Name;
  edit3.Text:=base[k].Od;
  edit4.Text:=FloatToStr(base[k].Price);
  edit5.Text:=FloatToStr(base[k].BeginKil);
  Label9.Caption:=FloatToStr(base[k].ProdKil);
  Label11.Caption:=FloatToStr(base[k].ProdKil*base[
k].Price);
  Label13.Caption:=FloatToStr(base[k].BeginKil-
base[k].ProdKil);
end;
// Утворення масиву товарів з текстового файлу
procedure TForm1.FormCreate(Sender: TObject);
var
  s: string;
  f: TextFile;
  n: integer;
begin
  AssignFile(f,'База.txt');
  Reset(f);
  n:=1;
  while not eof(f) do
  begin
    Tv:=TTovar.Create;

```

```

  readln(f,s);
  Tv.Kod:=s;
  readln(f,s);
  Tv.Name:=s;
  readln(f,s);
  Tv.Od:=s;
  readln(f,s);
  Tv.Price:=StrToFloat(s);
  readln(f,s);
  Tv.BeginKil:=StrToFloat(s);
  Tv.ProdKil:=0;
  Base[n]:=Tv;
  n:=n+1;
  ListBox1.AddItem(Tv.Name,T);
end;
CloseFile(F);
Tv:=TTovar.Create;
TV.Kod:="";
Base[n]:=Tv;
NumTovars:=n-1;
end;
// Додавання товару до покупки
procedure TForm1.Button2Click(Sender: TObject);
var
  Kil: real;
  k: integer;
  Er: boolean;
  Kod, Name, Od: string;
  Price, ProdKil, BeginKil: real;
begin
  k:=ListBox1.ItemIndex+1;
  if k>NumTovars then exit;
  Kil:=StrToFloat(Edit6.Text);
  Kod:=Base[k].Kod;
  Name:=Base[k].Name;
  Od:=Base[k].Od;
  Price:=Base[k].Price;
  ProdKil:=Base[k].ProdKil;
  BeginKil:=Base[k].BeginKil;
  Pok.Add(
Kod,Name,Od,Price,BeginKil,ProdKil,Kil,Er);
  if Er then
  begin
    ShowMessage('В ПОКУПЦІ ЦЬОГО ТОВАРУ
ВІДМОВЛЕНО');
    edit6.text:="";
  end
  else ShowMessage('ТОВАР ДОДАНО ДО ПО-
КУПКИ');
end;
// Утворення нової покупки
procedure TForm1.Button3Click(Sender: TObject);
begin
  Pok:=TPokupka.Create;
  Pok.Open;
  L.Caption:="";
  edit6.Text:="";
  ShowMessage('ОБ'ЄКТ ПОКУПКА УТВОРЕНО');
end;
// Закриття покупки
procedure TForm1.Button4Click(Sender: TObject);
var
  i,k,Count: integer;
  s: string;
begin
  Pok.Close;
  Count:=Pok.Count;
  {Відображення покупки в компоненті користу-
вача}

```

```

L.Caption:="";
for i:=1 to Count do
begin
    TV:=Pok.GetTovar(i);
    L.Caption:=L.Caption+ TV.Name+#9;
    L.Caption:=L.Caption+ TV.Od+#9;
    L.Caption:=L.Caption+
FloatToStr(TV.ProdKil)+#9;
    L.Caption:=L.Caption+
FloatToStr(TV.Price)+#9;
    L.Caption:=L.Caption+
FloatToStr(TV.ProdKil*TV.Price)+#13;
end;
if count>0 then L.Caption:=L.Caption+
#13+'РАЗОМ НА СУМУ: '+FloatToStr(Pok.Sum);
for i:=1 to Count do
begin
    TV:=Pok.GetTovar(i);
    for k:=1 to 100 do
    begin
        s:=Base[k].Kod;
        if s="" then break;
        if s=TV.Kod then
            begin
                Base[k].ProdKil:=Base[k].ProdKil+TV.ProdKil;
                break;
            end; end; end; end; end;
end.
    
```

Тестування класу «Товар»

Після запуску цього додатку було відображено список товарів (рис. 3).

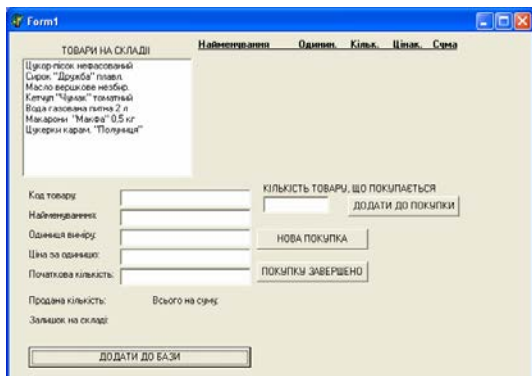


Рис. 3. Відображення списку товарів

Після натискання на будь-який рядок у цьому списку негайно отримуємо повну інформацію про товар (рис. 4).

Наступна перевірка показує, що довільне натискання на назві товару у списку або переміщення по списку за допомогою курсору також дає відображення повної інформації про товар.

На наступному етапі було перевірено зворотну операцію – дані вводилися вручну у відповідні поля і ця інформація додавалася до бази (кнопка «Додати до бази», працює процедура TForm1.Button1Click).

Таким чином було перевірено правильність доступу до властивостей об'єктів класу «Товар» для читання та запису (рис. 5).

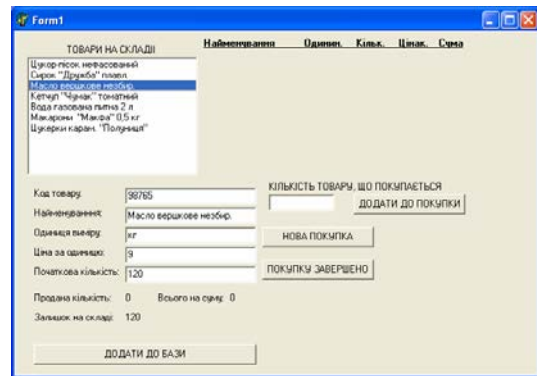


Рис. 4. Відображення інформації про товар, який виділено у списку

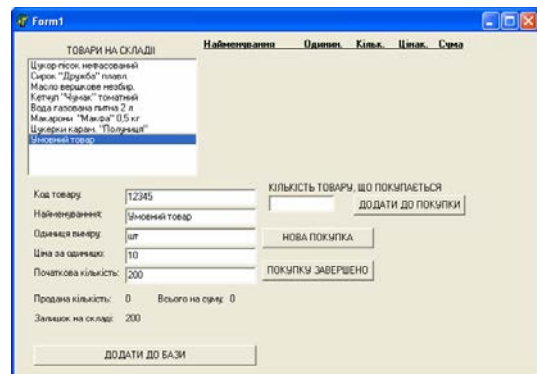


Рис. 5. Введення нового товару

Тестування класу «Покупка»

Перевірено властивості та методи класу «Покупка». При натисканні на кнопку «Нова покупка» (процедура TForm1.Button3Click) утворюється новий об'єкт класу «Покупка» і до нього застосовується метод Open (відкрити покупку). Метод відпрацював без помилок і отримано відповідне повідомлення (рис. 6).

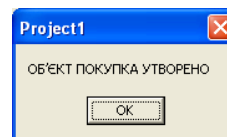


Рис. 6. Утворено нову покупку

Далі зі списку довільно вибиралися товари, вводилися їх кількість та натискалася кнопка «Додати до покупки». При цьому працює процедура TForm1.Button2Click, яка застосовує до об'єкту класу «Покупка» метод Add (Додати). При нормальному завершенні роботи цього методу видається відповідне повідомлення (рис. 7).

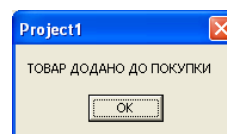


Рис. 7. Товар у визначеній кількості додано до покупки

Цей метод також фіксує помилкову ситуацію при додаванні товару до покупки, а саме коли кількість товару, що купується, більша за залишок цього товару на складі. Такі дані було введено та отримано відповідне повідомлення (рис. 8).

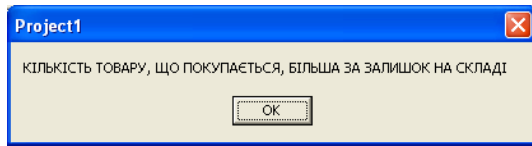


Рис. 8. Реакція методу Add на помилкову кількість товару

Наступне повідомлення вже видає прикладний додаток, який отримав від цього методу інформацію про помилку (рис. 9).

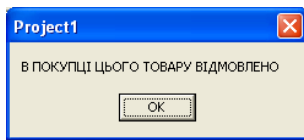


Рис. 9. Повідомлення з прикладного додатку

Тобто перевірено правильність роботи методу Add для класу «Покупка».

Після введення декількох товарів з їхньою кількістю (моделювання роботи каси) було натиснуто кнопку «Покупку завершено». При цьому працює процедура TForm1.Button4Click, яка застосовує метод Close. Далі ця процедура (аналог процедури підсистеми обліку) за допомогою властивості Count (кількість позицій у покупці) і методу GetTovar формує товарний чек, виводить його на екран і вносить зміни у кількість товарів в базі (масиву) (рис. 10).

Розроблено простий прикладний додаток, який тестує властивості і методи нових класів, а також демонструє їх можливості для розробки підсистем обліку продаж. Простота досягнута за рахунок переваг для програмістів, які дає об'єктно орієнтоване програмування.

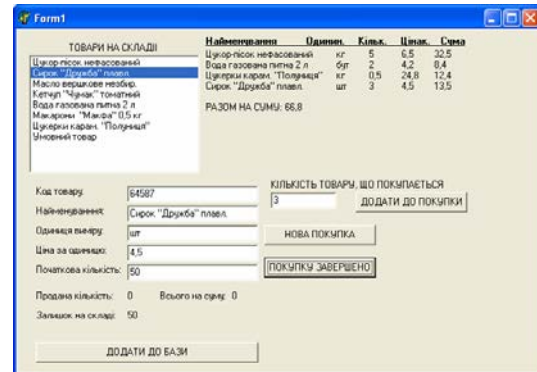


Рис. 10. Результат обробки завершення покупки

Основні результати роботи

1. Проведено тестування, яке показало правильність роботи властивостей та методів розроблених класів.
2. Тестування відбувалось шляхом розробки прикладного додатку з використанням нових класів. Показано, що простота в програмуванні цього додатку досягнута саме за рахунок використання нових класів, тобто тих переваг, які дає об'єктно орієнтоване програмування.
3. Нові класи можуть бути корисними для розробників відповідних прикладних додатків.

Список літератури

1. Бондаренко М.А. Розробка класів для підсистеми обліку продажу торговельного підприємства / М.А. Бондаренко, В.А. Жилин, Д.П. Панасенко, С.М. Трохимчук // Системи управління, навігації та зв'язку. – К.: ДП «ЦНДІ НіУ», 2011. – Вип. 2 (18). – С. 201-204.
2. Бондаренко М.А. Програмування на Object Pascal у середовищі Delphi 6 / М.А. Бондаренко. – Х.: Бізнес Інформ, 2002. – 344 с.
3. Бондаренко М.А. Основи інформаційних технологій та програмування / М.А. Бондаренко. – Х.: ФОП Павленко О.В., 2010. – 600 с.

Надійшла до редколегії 17.06.2011

Рецензент: д-р техн. наук, проф. В.Д. Сахацький, Українська інженерно-педагогічна академія, Харків.

РАЗРАБОТКА ПРИКЛАДНОГО ПРИЛОЖЕНИЯ ДЛЯ ТЕСТИРОВАНИЯ КЛАССОВ В ПОДСИСТЕМЕ УЧЕТА ПРОДАЖ ТОРГОВОГО ПРЕДПРИЯТИЯ

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

Рассматривается разработка прикладного приложения, которое позволяет тестировать новые классы, которые могут быть использованы для подсистемы учета продаж торгового предприятия. Также приведен пример тестирования вышеуказанных классов. Приведены основные результаты тестирования, которые дали надлежащие выводы относительно применения в разработке прикладного приложения объектно-ориентированного программирования. Разработка примера осуществлена средствами системы ООП Delphi 2010.

Ключевые слова: объектно-ориентированное программирование, прикладное приложение, тестирование классов.

DEVELOPMENT OF THE APPLIED APPLICATION FOR TESTING OF CLASSES IN THE SUBSYSTEM OF ACCOUNT OF SALES OF POINT-OF-SALE ENTERPRISE

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

We consider the development of application application that allows you to test new classes, which can be used for the subsystem excluding the sales trading business. Also an example of testing the above classes. The basic test results, which gave the appropriate conclusions regarding the application in the development of applied application of object-oriented programming. Development of an example implemented by means of the PLO Delphi 2010.

Keywords: object-oriented programming, applied application, testing of classes.