Міністерство освіти і науки, молоді та спорту України
Міжнародний економіко-гуманітарний університет
імені академіка Степана Дем'янчука
Факультет кібернетики
Кафедра математичного моделювання

Ціхоцька Катерина Валентинівна

ЗАСОБИ КОМП'ЮТЕРНОГО МОДЕЛЮВАННЯ НА МОВІ ПРОГРАМУВАННЯ С++ ПРИ ВИВЧЕННІ СКЛАДНИХ ЕКОНОМІЧНИХ ЯВИЩ

8.080201 – „Інформатика"

А В Т О Р Е Ф Е Р А Т
магістерської дисертації на здобуття академічного ступеня магістра з інформатики

Науковий керівник:
Р.М.Літнарович, доцент,
кандидат технічних наук

Рівне – 2012

УДК 004.42
Ціхоцька К.В. Засоби комп'ютерного моделювання на мові програмування С++ при вивченні складних економічних явищ. Автореферат магістерської дисертації на здобуття академічного ступеня магістра з інформатики. Науковий керівник Р.М.Літнарович. МЕГУ, Рівне, 2011.- 31 с.
. Робота виконана на кафедрі математичного моделювання Міжнародного економіко-гуманітарного університету імені академіка Степана Дем'янчука
Рецензенти:В.Г.Бурачек, доктор технічних наук, професор
В.О.Боровий, доктор технічних наук, професор
…….Є.С.Парняков, доктор технічних наук, професор
Відповідальний за випуск: Й.В.Джунь, доктор фіз.-мат. наук, професор

Об'єктом дослідження є сучасні методи та засоби комп'ютерного моделювання, а також мови програмування, що надають можливість розробляти власні програми для побудови математичних моделей.

Ключові слова: математична модель, мова програмування С++, економіка, програмний продукт.

Объектом исследования являются современные методы и средства компьютерного моделирования, а также языки программирования, которые предоставляют возможность разрабатывать собственные программы для построения математических моделей.

Ключевые слова: математическая модель, язык программирования С++, экономика, программный продукт.

A research object are modern methods and facilities of computer design, and also programming languages that give possibility to develop the own programs for the construction of mathematical models.

Keywords: mathematical model, programming of C++, economy, software product.

ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

Метою роботи *є написання програми, що дозволяє не лише побудувати модель економічного явища, а й обчислити такі важливі економічні характеристики, як прибуток, еластичність попиту, товарообіг, а також їх критичні значення.*

Робота є актуальною *тому, що більшість сучасних програм для статистичної обробки даних дають надто узагальнені висновки і не заглиблюються в результати побудованих моделей в контексті конкретної галузі.*

Наукова новизна *полягає в тому, що розроблена програма на ряду із математичними характеристиками змодельованого явища надає також основні економічні характеристики. Це дає змогу більш предметно оцінити результати проведених досліджень.*

Об'єктом дослідження *є сучасні методи та засоби комп'ютерного моделювання, а також мови програмування, що надають можливість розробляти власні програми для побудови математичних моделей.* В результаті даного дослідження мною було обрано середовище розробки програмного забезпечення Microsoft Visual Studio 2010 як одне з найсучасніших середовищ. Воно надає зручні інструменти для розробки сучасних програм, підвищення ефективності їх виконання та своєчасного виявлення помилок. Крім того засоби створення проекту, що вбудовані в середовище Visual C++ 2010, дозволяють автоматично створювати основу коду для широкого діапазону різноманітних прикладних програм.

Методологічною основою дисертації є Інтегроване середовище розробки (Integrated Development Environment – IDE), яке надається разом із середовищем Visual C++ 2010 – це повністю самодостатнє середовище, призначене для створення, компіляції, компонування і перевірки програм на мові C++. Це середовище включає в себе велику кількість повністю інтегрованих інструментів, призначених для полегшення написання програм.

Список фундаментальних складових середовища розробки Visual C++ 2010, що надаються IDE, включає в себе редактор, компілятор, компоновщик і бібліотеки. Це основні інструменти, що є необхідними для розробки, і відповідно були використані мною для написання програми.

Практична значимість *роботи полягає в створенні повноцінного програмного продукту, що може використовуватись на підприємствах що займаються торгівлею.*

Апробація роботи. Окремі розділи дисертації доповідались і отримали одобрення на наукових конференціях студентів і аспірантів у 2010 і 2011 роках, а також на науковому семінарі кафедри математичного моделювання.

Публікації. Основні положення дисертації опубліковані в монографії автора: Ціхоцька К. В. Засоби комп'ютерного моделювання на мові програмування С++ при вивченні складних економічних явищ. Монографія. Науковий керівник Р.М.Літнарович. МЕГУ, Рівне, 2012. - с.

Основні положення дисертації, що виносяться на захист:

- теоретичні основи побудови економіко-математичної моделі по способу найменших квадратів з повною оцінкою точності її елементів;
- аналіз середовища програмування Microsoft Visual Studio 2010;
- розробка допоміжних бібліотек;
- візуальне оформлення програми.

Структура і об'єм роботи. Магістерська дисертація складається із вступу, трьох розділів, розбитих на підрозділи, висновків і списку використаної літератури та додатків. Обсяг дисертації 86 сторінок, 15 рис. Список використаних джерел із 13 найменувань, в тому числі 4 складають Інтернет-джерела.

ОСНОВНИЙ ЗМІСТ РОБОТИ

У вступі обгрунтовується актуальність теми, даються основні положення дисертації, які виносяться на захист, та приводиться загальна характеристика магістерської дисертації.

В першому розділі описується поняття моделі та моделювання і розглядаються теоретичні основи побудови економіко-математичної моделі по способу найменших квадратів. Розробляються питання повної оцінки точності зрівноважених елементів економіко-математичної моделі.

Другий розділ описує середовище програмуванняMicrosoft Visual Studio 2010 і основні елементи інтерфейсу Microsoft Visual Studio 2010.

Третій розділ включає програмну реалізацію розробленого нами програмного продукту на мові програмування С++, опис та інтерфейс програми.

Сам програмний продукт приводиться в додатку до дисертації.

РОЗДІЛ 3. ЕТАПИ РОЗРОБКИ ПРОГРАМИ ТА ДЕМОНСТРАЦІЯ ЇЇ РОБОТИ

3.1. Розробка допоміжних бібліотек

Перед тим як приступити безпосередньо до розробки програми, підготовлено ряд бібліотек, що описують основні об’єкти та функції які будуть застосовуватись у програмі. А саме бібліотеку CMatrix.h, для роботи з матрицями і Modeling.h для побудови, власне, моделі та обчислення її основних математичних і економічних характеристик.

За допомогою бібліотеки CMatrix.h можна створювати матриці будь-якої розмірності і будь-якого типу даних. Також в бібліотеці передбачені наступні методи роботи з матрицями:

- int getRows(); – повертає кількість рядків матриці;
- int getColumns(); – повертає кількість стовпців матриці;

- int getNumOfElem(); – повертає кількість елементів матриці;
- T& operator () (unsigned int i, unsigned int j) – дозволязвертатись до елемента матриці $x_{ij;}$
- CMatrix<T>& CMatrix<T>::operator = (const CMatrix<T> &rhs) – перевантажений оператор =, дозволяє присвоювати значення одніє матриці іншій;
- CMatrix<T>& CMatrix<T>::operator += (const CMatrix<T> &rhs);, CMatrix<T>& CMatrix<T>::operator -= (const CMatrix<T> &rhs);, CMatrix<T>& CMatrix<T>::operator *= (const CMatrix<T> &rhs);, CMatrix<T>& CMatrix<T>::operator *= (const T &rhs);, CMatrix<T>& operator + (const CMatrix<T> &lhs, const CMatrix<T> &rhs);, CMatrix<T>& operator - (const CMatrix<T> &lhs, const CMatrix<T> &rhs);, CMatrix<T>& operator * (const CMatrix<T> &lhs, const CMatrix<T> &rhs);, CMatrix<T>& operator * (const CMatrix<T> &lhs, const T &rhs);, CMatrix<T>& operator * (const T &lhs, const CMatrix<T> &rhs); – перевантажені оператори, що дозволяють виконувати алгебраїчні дії над матрицями;
- T Determinant (const CMatrix<T> &M); – функція що повертає визначник матриці;
- CMatrix<T>& Inverse (const CMatrix<T> &M); – функція повертає матрицю обернену до тої, що передається в параметрі;
- CMatrix<T>& Minor (const CMatrix<T> &M, int fix_row, int fix_column); – повертає мінор матриці що передається у першому параметрі по рядку і стовпцю що передаються у другому і третьму параметрах відповідно;
- CMatrix<T>& Transpos (const CMatrix<T> &M) – повертає транспоновану матрицю що передається у параметрі.

Лістинг коду бібліотеки приведено в додатку А.

В бібліотеці Modeling.h описаний об’єкт, що містить усі основні характеристики моделі. Ці характеристики обчислюються

в 4 етапи під час виконання програми. Слід зазначити, що для побудови моделей застосовано метод найменших квадратів.

На першому етапі створюються динамічні масиви в яких зберігаються введені користувачем дані, а саме: кількість спостережень, ціна товару та відповідний цій ціні обсяг продажу за досліджуваний проміжок часу, а також витрати що пов'язані із кількістю виготовленої та реалізованої продукції і витрати що від неї не залежать.

На другому етапі обраховуються коефіцієнти a, b, c емпіричного рівняння:

$$y = ax^2 + bx + c,$$

що й буде уособленням математичних моделей у програмі. Для їх обрахунку обчислюється матриця обернена до матриці коефіцієнтів нормальних рівнянь. Після завершення виконання другого етапу вона не знищується, на відміну від інших змінних, що містили допоміжні обрахунки, а навпаки зберігається як один із членів класу. Це робиться тому, що дана матриця неодноразово буде використовуватись при обрахунку інших параметрів побудованих моделей.

На третьому етапі обраховуються основні економічні характеристики моделі, такі як товарообіг, собівартість проданої продукції, еластичність регресії попиту, прибуток, а також значення зрівноваженої функції. Усі знайдені значення зберігаються у вигляді динамічних масивів у змінних класу.

На завершальному етапі відбувається оцінка точності побудованої моделі. Обраховуються наступні величини:

- Абсолютні похибки зрівноваженої функції – це різниці значень отриманих практично та теоретично.
- Обернені ваги зрівноваженої функції за способом найменших квадратів:

$$\frac{1}{P_\varphi} = \varphi Q \varphi^T,$$

де φ – значення коефіцієнтів початкових рівнянь функції;

φ^T – транспонована матриця коефіцієнтів[2];

Q – обернена матриця коефіцієнтів нормальних рівнянь.

- Квадратичні похибки зрівноваженої функції, обчислюються як добуток кореня квадратного з обернених ваг на абсолютну похибку зрівноваженої функції.
- Обернені ваги коефіцієнтів рівняння. Ці значення розміщені у діагоналі оберненої матриці коефіцієнтів нормальних рівнянь Q наступним чином: $Q_{11}=1/P_c$, $Q_{22}=1/P_b$ і $Q_{33}=1/P_a$.[2]
- Середня квадратична похибка

$$\mu = \sqrt{\frac{[VV]}{n-m-1}},$$

де V – абсолютні похибки, n – кількість спостережень, m – порядок полінома[3].

- Середні квадратичні похибки коефіцієнтів емпіричного рівняння обчислюються як добуток середньої квадратичної похибки на корінь квадратний оберненої ваги коефіцієнта.
- Значимість коефіцієнтів емпіричного рівняння це відношення самого коефіцієнта до його середньої квадратичної похибки.
- Контроль зрівноваження обчислюється за контрольною формулою

$$[y^2]-a[yx^2]-b[yx]-c[y]=[\varepsilon\varepsilon]$$

Лістинг коду бібліотеки приводиться в додатку Б.

3.2. Візуальне оформлення програми

Усі вікна та форми програми було розроблено за допомогою Windows Forms. Допоміжні бібліотеки підключаються до головного вікна програми, адже саме у ньому будуть відображатись усі результати розрахунків.

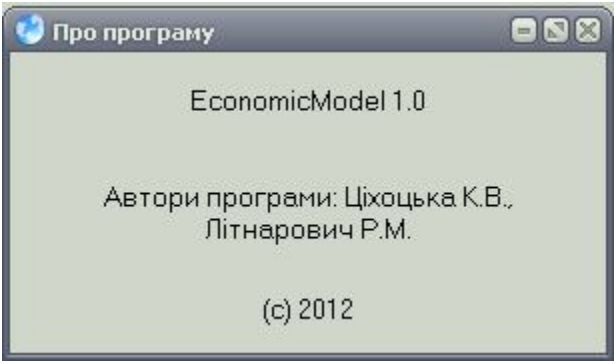
Після запуску програми у головному вікні відображається лише меню, що містить три пункти: «Нова модель», «Про програму», «Вихід».

При натисканні на пункті «Нова модель» відбувається виклик форми для внесення даних згідно яких проводитиметься

моделювання. Функція обробки даної події виглядає наступним чином:

```
void __fastcall TMainForm::N1Click(TObject *Sender)
{
    InputForm->ShowModal() ;
}
```

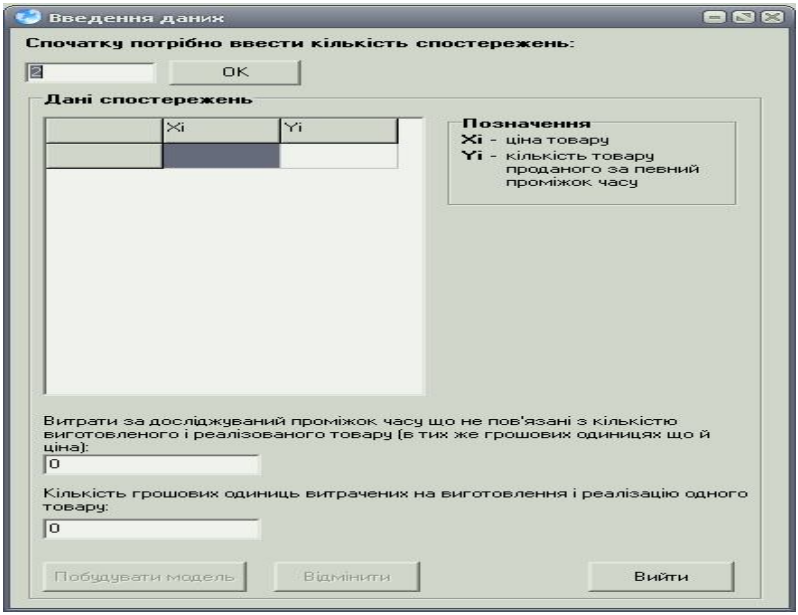
При обранні пункту «Про програму» викликається невеличке вікно із короткою інформацією про версію програми та її авторів (рис. 5).



(Рис. 5)

Пункт меню «Вихід» закриває програму, після чого відбувається знищення усіх введених даних, та вивільнення пам’яті виділеної під них.

Отже, повернемось до форми введення даних, що, як було зазначено раніше, викликається при натисканні на пункт меню «Нова модель». При відкритті ця форма має наступний вигляд:



(Рис. 6)

Користувачу спочатку пропонується ввести кількість спостережень (тобто відповідну кількість пар «ціна» – «кількість проданого товару»). Після чого в таблицю, яка знаходиться нижче додається зазначена кількість рядків. Далі користувач може або ввести необхідні дані у таблицю, або ж повернутись до попереднього пункту, якщо, наприклад, була вказана невірна кількість спостережень, натиснувши кнопку «Відмінити». При цьому всі дані з таблиці буде видалено, а кількість рядків стане такою як на початку. Також дану форму можна закрити натиснувши кнопку «Вихід».

Для зручності усі кнопки у формі введення даних забезпечені підказками, тобто якщо затримати курсор над будь-якою з них на декілька секунд, то з’явиться відповідна спливаюча підказка, що більш розширено описує призначення даної кнопки.

При натисканні на кнопку «Побудувати модель» відбувається перевірка введених даних. Якщо якісь із введених

користувачем значень не коректні, або ж не заповнені усі потрібні поля, з'являється повідомлення про помилку із роз'ясненням що саме було зроблено не так. Форма заповнена даними демонструється у додатку В.

Коли перевірка відбулась і усі введені дані визнані коректними відбувається, власне, побудова моделі, та обрахунок її математичних та економічних характеристик. Під час цього обрахунку відображається панель прогресу, де описується поточний етап розрахунків. Це дуже зручно при обробці великої кількості спостережень. Користувач може слідкувати за процесом обчислення і у разі неполадки йому буде відомо на якому саме етапі щось пішло не так.

Саме у тілі цієї форми відбуваються усі розрахунки, а обчислені дані передаються на вивід у головне вікно програми. Лістинг коду цих розрахунків:

```
void __fastcall TBuildModel::FormActivate(TObject *Sender)
{
    BuildModelComment->Caption = "Запис даних
спостережень..." ;
    /*Додання необхіної кількості рядків у таблицю*/
    int n = InputForm->TableData->RowCount - 1 ;
    double* x = new double[n] ;     // Масив цін
    double* y = new double[n] ;     // Масив кількостей
проданого товару
    /*Виведення даних у головне вікно програми*/
    for (int i = 1 ; i < InputForm->TableData->RowCount ; i++)
    {
        /*Перевірка чи не залишилось у таблиці порожніх
комірок*/
        if (InputForm->TableData->Cells[1][i].IsEmpty() ||
           InputForm->TableData->Cells[2][i].IsEmpty())
        {
            /*Виведення повідомлення про помилку*/
            ErrorForm2->ShowModal() ;
            BuildModelProgressBar->Position = 100 ;
            progressBarTimer->Enabled = true ;
```

```
                    return ;
                }
                *(x + i - 1) = InputForm->TableData-
>Cells[1][i].ToDouble() ;
                *(y + i - 1) = InputForm->TableData-
>Cells[2][i].ToDouble() ;
            }
            /*Перевірка чи не порожні поля із значеннями витрат*/
            if ((InputForm->InputDataIndependCost->Text.IsEmpty() ||
               (InputForm->InputDataIndependCost->Text.ToDouble()
< 0)) ||
               (InputForm->InputDataDependCost->Text.IsEmpty() ||
               (InputForm->InputDataDependCost->Text.ToDouble() <
0)))
            {
                /*Виведення повідомлення про помилку*/
                ErrorForm3->ShowModal() ;
                BuildModelProgressBar->Position = 100 ;
                progressBarTimer->Enabled = true ;
                return ;
            }
            /*Перевірка коректності введених значень витрат*/
            if ((InputForm->InputDataDependCost->Text.ToDouble()
== 0) &&
               (InputForm->InputDataIndependCost->Text.ToDouble()
== 0))
            {
                /*Виведення повідомлення про помилку*/
                ErrorForm4->ShowModal() ;
                BuildModelProgressBar->Position = 100 ;
                progressBarTimer->Enabled = true ;
                return ;
            }
            BuildModelProgressBar->Position += 25;
            BuildModelComment->Caption = "Обчислення
коефіцієнтів..." ;
```

```
            /*Створення об'єкту, що зберігатиме дані
можелювання*/
            MainForm->pAprObj = new CModeling(n, x, y,
           InputForm->InputDataIndependCost->Text.ToDouble(),
           InputForm->InputDataDependCost->Text.ToDouble()) ;
            /*Обчислення коефіцієнтів емпіричного рівняння*/
            MainForm->pAprObj->Aproximate() ;
            /*Виведення в головнуму вікні коефіцієнтів
емпіричного рівняння*/
            MainForm->outputA->Text = FormatFloat("0.000000",
            MainForm->pAprObj->coef_a) ;
            MainForm->outputB->Text = FormatFloat("0.000000",
            MainForm->pAprObj->coef_b) ;
            MainForm->outputC->Text = FormatFloat("0.000000",
            MainForm->pAprObj->coef_c) ;
            BuildModelProgressBar->Position += 25;
            BuildModelComment->Caption = "Виведення
економічних характеристик..." ;
            MainForm->MainFormObservOutput->Cells[1][0] = "Ціна"
;
            MainForm->MainFormObservOutput->Cells[2][0] =
"Обсяг продажу" ;
            MainForm->MainFormObservOutput->Cells[3][0] = "Теор.
обсяг продажу" ;
            MainForm->MainFormObservOutput->Cells[4][0] =
"Коеф. еластичності" ;
            MainForm->MainFormObservOutput->Cells[5][0] =
"Собівартість" ;
            MainForm->MainFormObservOutput->Cells[6][0] =
"Товарообіг" ;
            MainForm->MainFormObservOutput->Cells[7][0] =
"Прибуток" ;
            for (int i = 1 ; i < n+1 ; i++)
                  MainForm->MainFormObservOutput->Cells[0][i] = i ;
            MainForm->MainFormObservOutput->RowCount = n + 1 ;
            /*Обчислення економічних характеристик*/
```

```
            MainForm->pAprObj->getCalculations() ;
            /*Виведення таблиці з економічними характеристиками
моделей*/
            for (int i = 1 ; i < n+1 ; i++)
            {
                  /*Виведення цін*/
                  MainForm->MainFormObservOutput->Cells[1][i] =
                    MainForm->pAprObj->x[i-1] ;
                  /*Виведення попиту*/
                  MainForm->MainFormObservOutput->Cells[2][i] =
                    MainForm->pAprObj->y[i-1] ;
                  /*Виведення значенн зрівноваженої функції*/
                  MainForm->MainFormObservOutput->Cells[3][i] =
FormatFloat("0.000000", MainForm->pAprObj->calcY[i-1]) ;
                  /*Виведення коефіцієнту еластичності*/
                  MainForm->MainFormObservOutput->Cells[4][i] =
FormatFloat("0.000000", MainForm->pAprObj->coefEl[i-1]) ;
                  /*Вивеення собівартості*/
                  MainForm->MainFormObservOutput->Cells[5][i] =
FormatFloat("0.000000", MainForm->pAprObj->cost[i-1]) ;
                  /*Виведення товарообігу*/
                  MainForm->MainFormObservOutput->Cells[6][i] =
FormatFloat("0.000000", MainForm->pAprObj->trade[i-1]) ;
                  /*Виведення прибутку*/
                  MainForm->MainFormObservOutput->Cells[7][i] =
FormatFloat("0.000000", MainForm->pAprObj->profit[i-1]) ;
                  /*Відображення економічних характеристик у
вигляді графіків*/
                  MainForm->empericalData->AddXY(x[i-1], y[i-1]) ;
                  MainForm->balancedData->AddXY(x[i-1],
                   MainForm->pAprObj->calcY[i-1]) ;
                  MainForm->Cost->AddXY(x[i-1],
                   MainForm->pAprObj->cost[i-1]) ;
                  MainForm->Trade->AddXY(x[i-1],
                   MainForm->pAprObj->trade[i-1]) ;
                  MainForm->Profit->AddXY(x[i-1],
```

```
                MainForm->pAprObj->profit[i-1]) ;
                MainForm->Demand->AddXY(x[i-1],
                MainForm->pAprObj->calcY[i-1]) ;
                MainForm->StatisticDemand->AddXY(x[i-1], y[i-1]) ;
                MainForm->Trade_2->AddXY(x[i-1],
                MainForm->pAprObj->trade[i-1]) ;
                MainForm->CoefEl->AddXY(x[i-1],
                MainForm->pAprObj->coefEl[i-1]) ;

        }
        BuildModelProgressBar->Position += 25;
        BuildModelComment->Caption = "Обчислення критичних
значеннь економічних характеристик..." ;
        double p1, p2 ;
        int index = -1 ;
        double temp = sqrt(pow(MainForm->pAprObj->coef_b,2) -
3*MainForm->pAprObj->coef_a*MainForm->pAprObj->coef_c) ;
        /*Можливі критичні точки товарообігу*/
        p1 = (-MainForm->pAprObj->coef_b + temp)/(3*
              MainForm->pAprObj->coef_a) ;
        p2 = (-MainForm->pAprObj->coef_b - temp)/(3*
    MainForm->pAprObj->coef_a) ;
        /*Перевірка чи потрапляє знайдена критична точка у
          область досліджуваних значень*/
        if (p1 >= x[0] && p1 <= x[n-1])
        {
            for (int i = 0 ; i < n ; i++)
                if (x[i] >= p1)
                {
                    index = i ;
                    break ;
                }
            /*Перевірка чи знайдена критична точка є точкою
мінімуму*/
            if ((MainForm->pAprObj->trade[index] <=
                MainForm->pAprObj->trade[0]) &&
```

```
                (MainForm->pAprObj->trade[index] <=
                  MainForm->pAprObj->trade[n-1]))
                {
                    /*Виведення значень економічних показників в
цій точці*/
                    MainForm->NoneMinTrade->Visible = false ;
                    MainForm->minTradePriceText->Visible = true ;
                    MainForm->MinTradePrice->Visible = true ;
                    MainForm->MinTradePrice->Text =
FormatFloat("0.000000", p1) ;
                    MainForm->minTradeCoefElText->Visible = true
;
                    MainForm->minTradeCoefEl->Visible = true ;
                    MainForm->minTradeCoefEl->Text =
FormatFloat("0.000000", MainForm->pAprObj->getCoefEl(p1)) ;
                    MainForm->minTradeValueText->Visible = true ;
                    MainForm->minTradeVal->Visible = true ;
                    MainForm->minTradeVal->Text =
FormatFloat("0.000000", MainForm->pAprObj->getTrade(p1)) ;
                }
                /*Перевірка чи знайдена критична точка є точкою
максимуму*/
                if ((MainForm->pAprObj->trade[index] >=
                    MainForm->pAprObj->trade[0]) &&
                (MainForm->pAprObj->trade[index] >=
                  MainForm->pAprObj->trade[n-1]))
                {
                    /*Виведення значень економічних показників в
цій точці*/
                    MainForm->NoneMaxTrade->Visible = false ;
                    MainForm->maxTradePriceText->Visible = true ;
                    MainForm->maxTradePrice->Visible = true ;
                    MainForm->maxTradePrice->Text =
FormatFloat("0.000000", p1) ;
                    MainForm->maxTradeCoefElText->Visible = true
;
```

```
                    MainForm->maxTradeCoefEl->Visible = true ;
                    MainForm->maxTradeCoefEl->Text =
FormatFloat("0.000000", MainForm->pAprObj->getCoefEl(p1)) ;
                    MainForm->maxTradeValueText->Visible = true
;
                    MainForm->maxTradeVal->Visible = true ;
                    MainForm->maxTradeVal->Text =
FormatFloat("0.000000", MainForm->pAprObj->getTrade(p1)) ;
                }
            }
            /*Перевірка чи потрапляє знайдена критична точка у
              область досліджуваних значень*/
            if (p2 >= x[0] && p2 <= x[n-1])
            {
                for (int i = 0 ; i < n ; i++)
                    if (x[i] >= p2)
                    {
                        index = i ;
                        break ;
                    }
                /*Перевірка чи знайдена критична точка є точкою
мінімуму*/
                if ((MainForm->pAprObj->trade[index] <=
                    MainForm->pAprObj->trade[0]) &&
                   (MainForm->pAprObj->trade[index] <=
                    MainForm->pAprObj->trade[n-1]))
                {
                    /*Виведення значень економічних показників в
цій точці*/
                    MainForm->NoneMinTrade->Visible = false ;
                    MainForm->minTradePriceText->Visible = true ;
                    MainForm->MinTradePrice->Visible = true ;
                    MainForm->MinTradePrice->Text =
FormatFloat("0.000000", p2) ;
                    MainForm->minTradeCoefElText->Visible = true
;
```

```
                    MainForm->minTradeCoefEl->Visible = true ;
                    MainForm->minTradeCoefEl->Text =
FormatFloat("0.000000", MainForm->pAprObj->getCoefEl(p2)) ;
                    MainForm->minTradeValueText->Visible = true ;
                    MainForm->minTradeVal->Visible = true ;
                    MainForm->minTradeVal->Text =
FormatFloat("0.000000", MainForm->pAprObj->getTrade(p2)) ;
                }
                /*Перевірка чи знайдена критична точка є точкою
максимуму*/
                if ((MainForm->pAprObj->trade[index] >=
                     MainForm->pAprObj->trade[0]) &&
                    (MainForm->pAprObj->trade[index] >=
                     MainForm->pAprObj->trade[n-1]))
                {
                    /*Виведення значень економічних показників в
цій точці*/
                    MainForm->NoneMaxTrade->Visible = false ;
                    MainForm->maxTradePriceText->Visible = true ;
                    MainForm->maxTradePrice->Visible = true ;
                    MainForm->maxTradePrice->Text =
FormatFloat("0.000000", p2) ;
                    MainForm->maxTradeCoefElText->Visible = true
;
                    MainForm->maxTradeCoefEl->Visible = true ;
                    MainForm->maxTradeCoefEl->Text =
FormatFloat("0.000000", MainForm->pAprObj->getCoefEl(p2)) ;
                    MainForm->maxTradeValueText->Visible = true
;
                    MainForm->maxTradeVal->Visible = true ;
                    MainForm->maxTradeVal->Text =
FormatFloat("0.000000", MainForm->pAprObj->getTrade(p2)) ;
                }
            }
            double dv = 4 * (pow((MainForm->pAprObj->coef_b -
```

```
        MainForm->pAprObj->dependCost*MainForm->pAprObj-
    >coef_a), 2)  +  3 * MainForm->pAprObj->coef_a *
    (MainForm->pAprObj->
    coef_b*MainForm->pAprObj->dependCost -
    MainForm->pAprObj->coef_c)) ;
        /*Обчислення можливих критичних точок прибутку*/
        p1 = (MainForm->pAprObj->dependCost *
              MainForm->pAprObj->coef_a -
              MainForm->pAprObj->coef_b + 0.5*sqrt(dv))/
             (3*MainForm->pAprObj->coef_a) ;
        p2 = (MainForm->pAprObj->dependCost *
              MainForm->pAprObj->coef_a -
              MainForm->pAprObj->coef_b - 0.5*sqrt(dv))/
            (3*MainForm->pAprObj->coef_a) ;
        /*Перевірка чи потрапляє знайдена критична точка у
        область досліджуваних значень*/
        if (p1 >= x[0] && p1 <=x[n-1])
        {
            for (int i = 0 ; i < n ; i++)
                if (x[i] >= p1)
                {
                    index = i ;
                    break ;
                }
            /*Перевірка чи знайдена критична точка є точкою
мінімуму*/
            if ((MainForm->pAprObj->profit[index] <=
                 MainForm->pAprObj->profit[0]) &&
                (MainForm->pAprObj->profit[index] <=
                 MainForm->pAprObj->profit[n-1]))
            {
                /*Виведення значень економічних показників в
цій точці*/
                MainForm->NoneMinProfit->Visible = false ;
                MainForm->minProfitPriceText->Visible = true ;
                MainForm->minProfitPrice->Visible = true ;
```

```
                    MainForm->minProfitPrice->Text =
FormatFloat("0.000000", p1) ;
                    MainForm->minProfitTradeText->Visible = true ;
                    MainForm->minProfitTrade->Visible = true ;
                    MainForm->minProfitTrade->Text =
FormatFloat("0.000000", MainForm->pAprObj->getTrade(p1));
                    MainForm->minProfitValText->Visible = true ;
                    MainForm->minProfitVal->Visible = true ;
                    MainForm->minProfitVal->Text =
FormatFloat("0.000000", MainForm->pAprObj->getProfit(p1)) ;
                }
                /*Перевірка чи знайдена критична точка є точкою
максимуму*/
                if ((MainForm->pAprObj->profit[index] >=
                     MainForm->pAprObj->profit[0]) &&
                    (MainForm->pAprObj->profit[index] >=
                   MainForm->pAprObj->profit[n-1]))
                {
                    /*Виведення значень економічних показників в
цій точці*/
                    MainForm->NoneMaxProfit->Visible = false ;
                    MainForm->maxProfitPriceText->Visible = true ;
                    MainForm->maxProfitPrice->Visible = true ;
                    MainForm->maxProfitPrice->Text =
FormatFloat("0.000000", p1) ;
                    MainForm->maxProfitTradeText->Visible = true ;
                    MainForm->maxProfitTrade->Visible = true ;
                    MainForm->maxProfitTrade->Text =
FormatFloat("0.000000", MainForm->pAprObj->getTrade(p1));
                    MainForm->maxProfitValText->Visible = true ;
                    MainForm->maxProfitVal->Visible = true ;
                    MainForm->maxProfitVal->Text =
FormatFloat("0.000000", MainForm->pAprObj->getProfit(p1)) ;
                }
            }
            /*Перевірка чи потрапляє знайдена критична точка у
```

```
            область досліджуваних значень*/
            if (p2 >= x[0] && p2 <=x[n-1])
            {
                for (int i = 0 ; i < n ; i++)
                    if (x[i] >= p2)
                    {
                        index = i ;
                        break ;
                    }
                /*Перевірка чи знайдена критична точка є точкою
мінімуму*/
                if ((MainForm->pAprObj->profit[index] <=
                     MainForm->pAprObj->profit[0]) &&
                    (MainForm->pAprObj->profit[index] <=
                     MainForm->pAprObj->profit[n-1]))
                {
                    /*Виведення значень економічних показників в
цій точці*/
                    MainForm->NoneMinProfit->Visible = false ;
                    MainForm->minProfitPriceText->Visible = true ;
                    MainForm->minProfitPrice->Visible = true ;
                    MainForm->minProfitPrice->Text =
FormatFloat("0.000000", p2) ;
                    MainForm->minProfitTradeText->Visible = true ;
                    MainForm->minProfitTrade->Visible = true ;
                    MainForm->minProfitTrade->Text =
FormatFloat("0.000000", MainForm->pAprObj->getTrade(p2));
                    MainForm->minProfitValText->Visible = true ;
                    MainForm->minProfitVal->Visible = true ;
                    MainForm->minProfitVal->Text =
FormatFloat("0.000000", MainForm->pAprObj->getProfit(p2)) ;
                }
                /*Перевірка чи знайдена критична точка є точкою
максимуму*/
                if ((MainForm->pAprObj->profit[index] >=
                     MainForm->pAprObj->profit[0]) &&
```

```
                    (MainForm->pAprObj->profit[index] >=
                     MainForm->pAprObj->profit[n-1]))
                {
                    /*Виведення значень економічних показників в
цій точці*/
                    MainForm->NoneMaxProfit->Visible = false ;
                    MainForm->maxProfitPriceText->Visible = true ;
                    MainForm->maxProfitPrice->Visible = true ;
                    MainForm->maxProfitPrice->Text =
FormatFloat("0.000000", p2) ;
                    MainForm->maxProfitTradeText->Visible = true ;
                    MainForm->maxProfitTrade->Visible = true ;
                    MainForm->maxProfitTrade->Text =
FormatFloat("0.000000", MainForm->pAprObj->getTrade(p2));
                    MainForm->maxProfitValText->Visible = true ;
                    MainForm->maxProfitVal->Visible = true ;
                    MainForm->maxProfitVal->Text =
FormatFloat("0.000000", MainForm->pAprObj->getProfit(p2)) ;
                }
            }
            BuildModelProgressBar->Position += 25;
            BuildModelComment->Caption = "Обчислення
характеристик побудованої моделі..." ;
            MainForm->ModelCharactTable->RowCount = n + 1 ;
            MainForm->ModelCharactTable->Cells[1][0] = "Ціна" ;
            MainForm->ModelCharactTable->Cells[2][0] = "Теор.
обсяг продажу" ;
            MainForm->ModelCharactTable->Cells[3][0] = "Абс.
похибки" ;
            MainForm->ModelCharactTable->Cells[4][0] = "1/P" ;
            MainForm->ModelCharactTable->Cells[5][0] = "m(p)" ;
            MainForm->ModelCharactTable->Cells[6][0] =
"Конструювання" ;
            /*Обчислення математичних характеристик моделі*/
            MainForm->pAprObj->getModelCharacts() ;
            /*Виведення таблиці з характеристиками моделей*/
```

```
            for (int i = 1 ; i < n+1 ; i++)
            {
                MainForm->ModelCharactTable->Cells[0][i] = i ;
                /*Ціна*/
                MainForm->ModelCharactTable->Cells[1][i] = x[i-1] ;
                /*Значення зрівноваженої функції*/
                MainForm->ModelCharactTable->Cells[2][i] =
FormatFloat("0.000000", MainForm->pAprObj->calcY[i-1]) ;
                /*Абсолютні похибки*/
                MainForm->ModelCharactTable->Cells[3][i] =
FormatFloat("0.000000", MainForm->pAprObj->abs_error[i-1]) ;
                /*Обернені ваги*/
                MainForm->ModelCharactTable->Cells[4][i] =
FormatFloat("0.000000", MainForm->pAprObj->abs_error_weight[i-
1]) ;
                /*Квадратичні похибки*/
                MainForm->ModelCharactTable->Cells[5][i] =
FormatFloat("0.000000", MainForm->pAprObj->square_errors[i-1]) ;
                /*Значення функції з урахуванням похибок*/
                MainForm->ModelCharactTable->Cells[6][i] =
FormatFloat("0.000000", MainForm->pAprObj->refinedY[i-1]) ;
                /*Відображення характеристик моделі у вигляді
графіку*/
                MainForm->InversWeights->AddXY(x[i-1],
                MainForm->pAprObj->abs_error_weight[i-1]) ;
                MainForm->AbsoluteErrors->AddXY(x[i-1],
                MainForm->pAprObj->abs_error[i-1]) ;
                MainForm->MeanSquareErrors->AddXY(x[i-1],
                MainForm->pAprObj->square_errors[i-1]) ;
            }
            /*Таблиця коефіцієнтів емпіричної функції*/
            MainForm->CoefCharactTable->Cells[0][1] = "a" ;
            MainForm->CoefCharactTable->Cells[0][2] = "b" ;
            MainForm->CoefCharactTable->Cells[0][3] = "c" ;
            MainForm->CoefCharactTable->Cells[1][0] = "1/p" ;
            MainForm->CoefCharactTable->Cells[2][0] = "m(коеф.)" ;
```

```
            MainForm->CoefCharactTable->Cells[3][0] = "t(коеф.)" ;
            /*Обернені ваги коефіцієнтів*/
            MainForm->CoefCharactTable->Cells[1][1] =
FormatFloat("0.000000", MainForm->pAprObj->inv_weight_a) ;
            MainForm->CoefCharactTable->Cells[1][2] =
FormatFloat("0.000000", MainForm->pAprObj->inv_weight_b) ;
            MainForm->CoefCharactTable->Cells[1][3] =
FormatFloat("0.000000", MainForm->pAprObj->inv_weight_c) ;
            /*Середні квадратичні похибки коефіцієнтів*/
            MainForm->CoefCharactTable->Cells[2][1] =
FormatFloat("0.000000", MainForm->pAprObj->m_a) ;
            MainForm->CoefCharactTable->Cells[2][2] =
FormatFloat("0.000000", MainForm->pAprObj->m_b) ;
            MainForm->CoefCharactTable->Cells[2][3] =
FormatFloat("0.000000", MainForm->pAprObj->m_c) ;
            /*Значимість коефіцієнтів*/
            MainForm->CoefCharactTable->Cells[3][1] =
FormatFloat("0.000000", MainForm->pAprObj->t_a) ;
            MainForm->CoefCharactTable->Cells[3][2] =
FormatFloat("0.000000", MainForm->pAprObj->t_b) ;
            MainForm->CoefCharactTable->Cells[3][3] =
FormatFloat("0.000000", MainForm->pAprObj->t_c) ;
            /*Контроль зрівноваження*/
            MainForm->controlCompensationLeftSide->Text =
            MainForm->pAprObj->controlCompensLftSd ;
            MainForm->controlCompensationRightSide->Text =
            MainForm->pAprObj->controlCompensRghtSd ;
            MainForm->controlCompensationDelta->Text =
             MainForm->pAprObj->controlCompensDelta ;
            /*Закриття форми введення спостережень*/
            InputForm->Close();
            MainForm->MainFormPageControl->Visible = true ;
            /*Вивільнення динамічної пам'яті*/
            delete []x ;
            delete []y ;
            x = 0x000000 ;
```

```
        y = 0x000000 ;
        BuildModelProgressBar->Position = 100 ;
        progressBarTimer->Enabled = true ;
}
```

Якщо обробка пройшла успішно, то панель прогресу закривається разом із формою введення даних, а головне вікно знову стає активним.

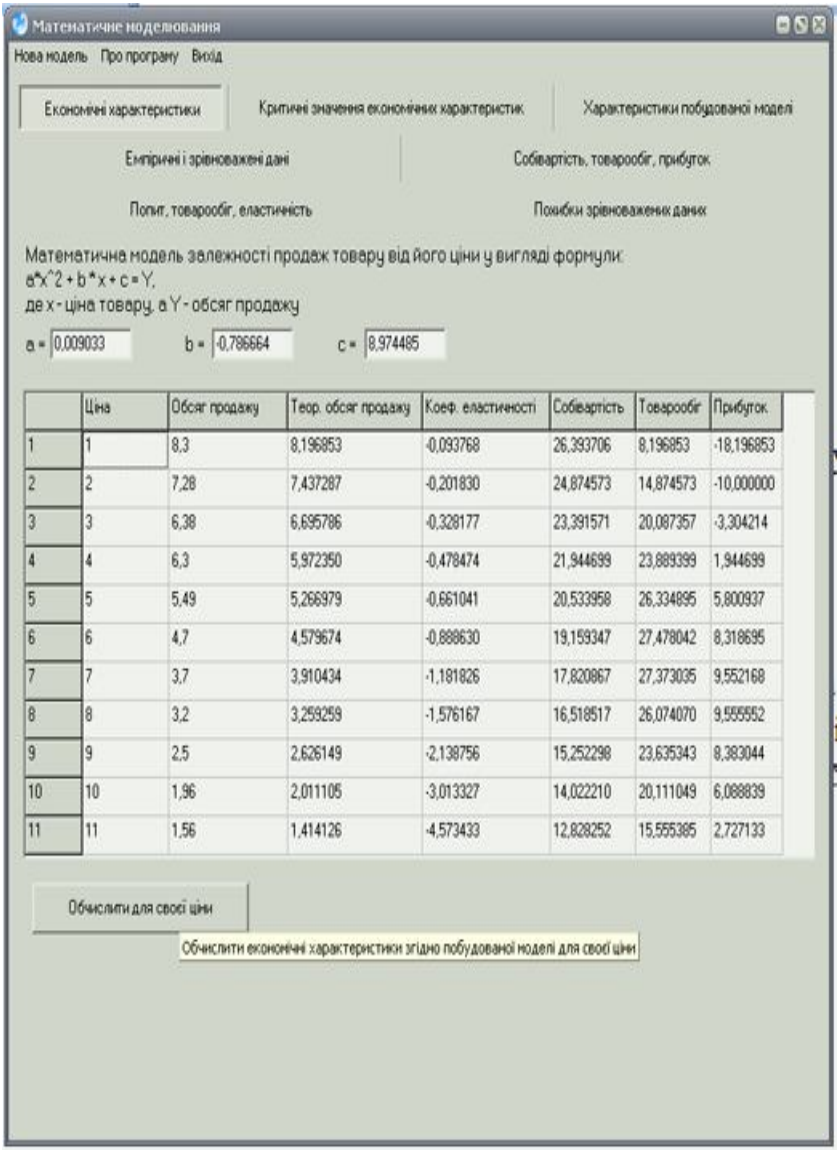
Тепер у головному вікні відображаються усі обчислені дані, для зручності їх розділено на 7 вкладок:

- Економічні характеристики
- Критичні значення економічних характеристик
- Характеристики побудованої моделі
- Емпіричні і зрівноважені дані (Додаток Д)
- Собівартість, товарообіг, прибуток (Додаток Е)
- Попит, товарообіг, еластичність (Додаток Ж)
- Похибки зрівноважених даних (Додаток З)

У перших трьох вкладках дані відображаються в табличному вигляді, а у інших чотирьох у вигляді графіків.

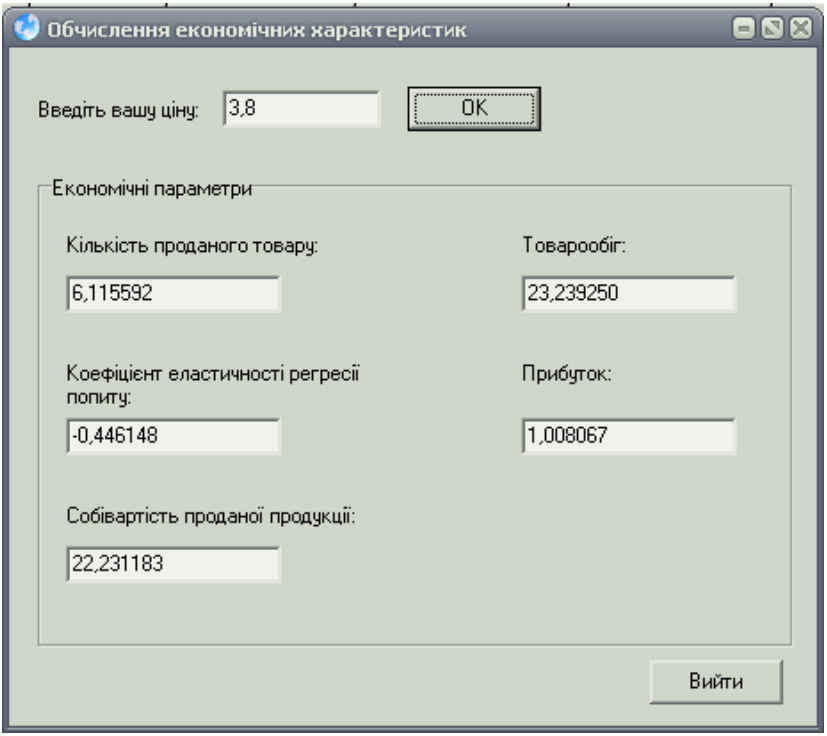
У вкладці «Економічні характеристики» розміщені такі дані: коефіцієнти емпіричної функції, ціна товару, обсяг продажу, теоретичний обсяг продажу, коефіцієнт еластичності попиту, собівартість реалізованої продукції, товарообіг, прибуток (рис. 7).

Також, передбачена можливість обрахувати усі вище зазначені характеристики (крім коефіцієнтів) для довільної ціни. Для цього під таблицею реалізована кнопка «Обчислити для своєї ціни», що викликає відповідну форму(рис. 8). Після вводу ціни і натискання кнопки «ОК» відбувається обчислення і виведення потрібних значень. У формі також реалізовано контроль, якщо користувач введе некоректне значення, наприклад, від'ємне, то форма видасть повідомлення про помилку із роз'ясненням яким повинен бути коректний ввід. Лістинг коду даної форми розміщений у додатку К.



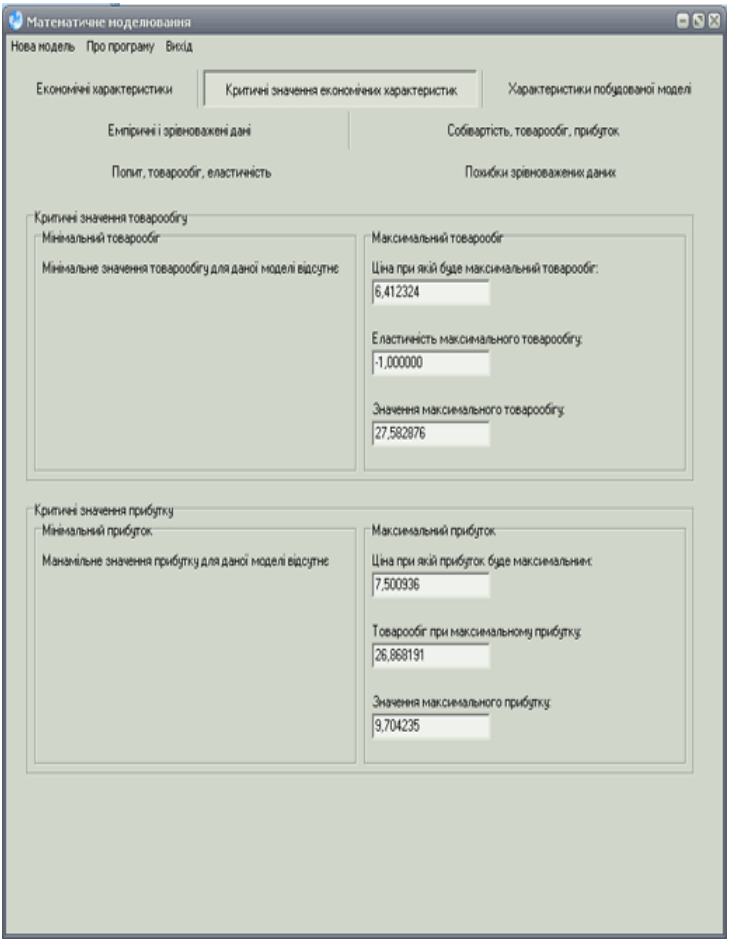
	Ціна	Обсяг продажу	Теор. обсяг продажу	Коеф. еластичності	Собівартість	Товарообіг	Прибуток
1	1	8,3	8,196853	-0,093768	26,393706	8,196853	-18,196853
2	2	7,28	7,437287	-0,201830	24,874573	14,874573	-10,000000
3	3	6,38	6,695786	-0,328177	23,391571	20,087357	-3,304214
4	4	6,3	5,972350	-0,478474	21,944699	23,889399	1,944699
5	5	5,49	5,266979	-0,661041	20,533958	26,334895	5,800937
6	6	4,7	4,579674	-0,888630	19,159347	27,478042	8,318695
7	7	3,7	3,910434	-1,181826	17,820867	27,373035	9,552168
8	8	3,2	3,259259	-1,576167	16,518517	26,074070	9,555552
9	9	2,5	2,626149	-2,138756	15,252298	23,635343	8,383044
10	10	1,96	2,011105	-3,013327	14,022210	20,111049	6,088839
11	11	1,56	1,414126	-4,573433	12,828252	15,555385	2,727133

(Рис. 7)



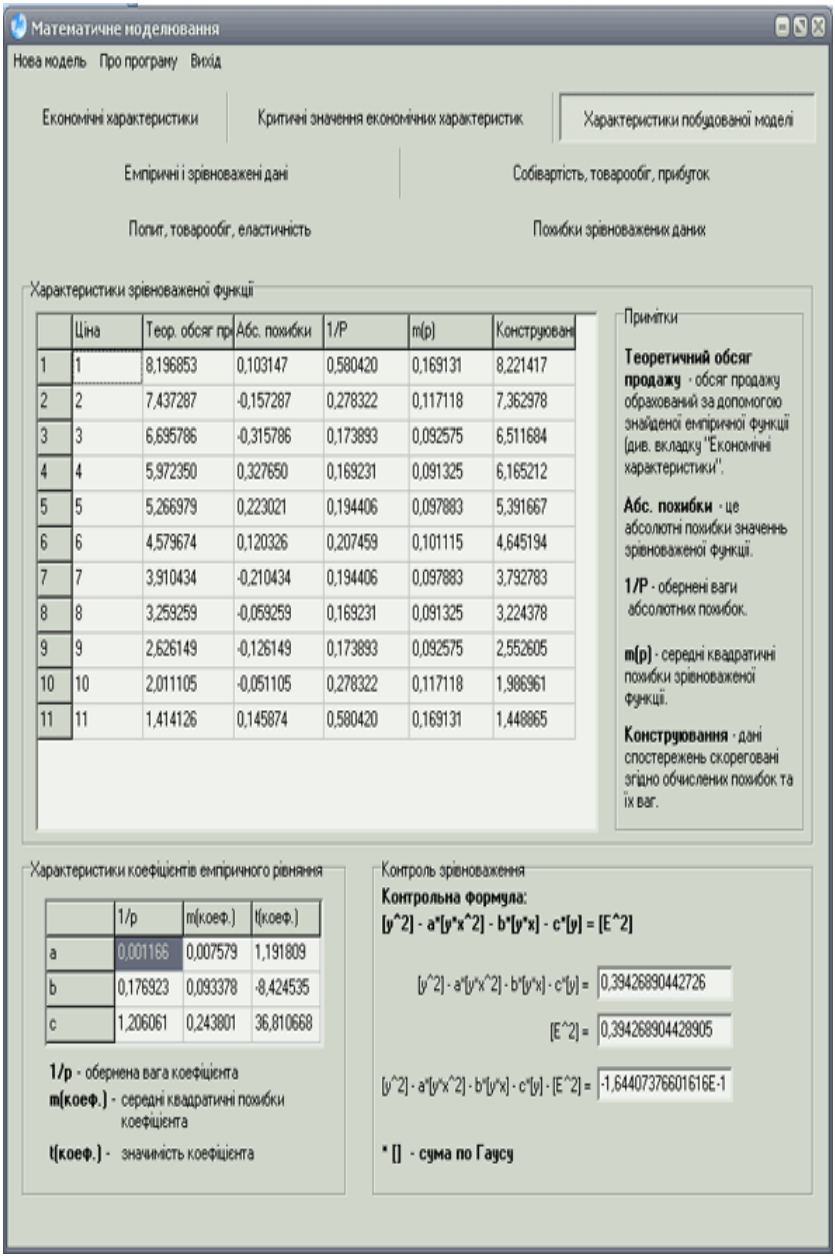
(Рис. 8)

У вкладці «Критичні значення економічних характеристик» виводяться мінімальні і максимальні значення прибутку і товарообігу, якщо такі існують. Якщо ж їх немає, то у формі виводиться відповідне повідомлення (рис. 9)



(Рис.9)

Вкладка «Характеристики побудованої моделі» розділена на три основних блока: «Характеристики зрівноваженої функції», «Характеристики коефіцієнтів емпіричного рівняння» і «Контроль зрівноваження». У зв'язку з тим, що в назвах колонок таблиці є багато скорочень, поряд із табличними даними знаходяться примітки, що коротко описують їх значення. (рис. 10)


Математичне моделювання
Нова модель
Про програму
Вихід
Економічні характеристики
Критичні значення економічних характеристик
Характеристики побудованої моделі
Емпіричні і зрівноважені дані
Собівартість, товарообіг, прибуток
Попит, товарообіг, еластичність
Похибки зрівноважених даних
Характеристики зрівноваженої функції
Ціна
Теор. обсяг пр
Абс. похибки
1/P
m(p)
Конструювані
1 1 8,196853 0,103147 0,580420 0,169131 8,221417
2 2 7,437287 -0,157287 0,278322 0,117118 7,362978
3 3 6,695786 -0,315786 0,173893 0,092575 6,511684
4 4 5,972350 0,327650 0,169231 0,091325 6,165212
5 5 5,266979 0,223021 0,194406 0,097883 5,391667
6 6 4,579674 0,120326 0,207459 0,101115 4,645194
7 7 3,910434 -0,210434 0,194406 0,097883 3,792783
8 8 3,259259 -0,059259 0,169231 0,091325 3,224378
9 9 2,626149 -0,126149 0,173893 0,092575 2,552605
10 10 2,011105 -0,051105 0,278322 0,117118 1,986961
11 11 1,414126 0,145874 0,580420 0,169131 1,448865
Примітки
Теоретичний обсяг продажу - обсяг продажу обрахований за допомогою знайденої емпіричної функції (див. вкладку "Економічні характеристики".
Абс. похибки - це абсолютні похибки значень зрівноваженої функції.
1/P - обернені ваги абсолютних похибок.
m(p) - середні квадратичні похибки зрівноваженої функції.
Конструювання - дані спостережень скореговані згідно обчислених похибок та їх ваг.
Характеристики коефіцієнтів емпіричного рівняння
1/p
m(коеф.)
t(коеф.)
a 0,001166 0,007579 1,191809
b 0,176923 0,093378 -8,424535
c 1,206061 0,243801 36,810668
1/p - обернена вага коефіцієнта
m(коеф.) - середні квадратичні похибки коефіцієнта
t(коеф.) - значимість коефіцієнта
Контроль зрівноваження
Контрольна формула:
[y^2] - a*[y*x^2] - b*[y*x] - c*[y] = [E^2]
[y^2] - a*[y*x^2] - b*[y*x] - c*[y] = 0,39426890442726
[E^2] = 0,394268904428905
[y^2] - a*[y*x^2] - b*[y*x] - c*[y] - [E^2] = -1,64407376601616E-1
* [] - сума по Гаусу

(Рис. 10)

ВИСНОВКИ

В ході виконання даної роботи було проаналізовано засоби мови С++, які можуть використовуватись у математичному моделюванні і на основі отриманої інформації розроблено повноцінний програмний продукт, що:

- дозволяє моделювати та досліджувати складні економічні явища на основі введених користувачем даних;
- встановлює залежність між ціною на товар та попитом на основі формули:

$$y = ax^2 + bx + c,$$

де y – попит на товар, а x – ціна одиниці товару;

- обчислює ряд економічних характеристик змодельованого явища, таких як еластичність регресії попиту, собівартість реалізованої продукції, товарообіг, прибуток від проданого товару;
- визначає критичні значення прибутку і товарообігу, якщо такі є;
- обчислює характеристики значень зрівноваженої функції: обернені ваги, абсолютні похибки та середні квадратичні похибки;
- обчислює характеристики коефіцієнтів функції залежності попиту від ціни: їх обернені ваги, середні квадратичні похибки та значимість;
- проводить контроль зрівноваження;
- наочно демонструє залежність економічних показників один від одного за допомогою графіків;
- здійснює контроль над коректністю введених користувачем даних.

Ціхоцька Катерина Валентинівна

спеціаліст системотехнік, магістрант інформаційних технологій

ЗАСОБИ КОМП'ЮТЕРНОГО МОДЕЛЮВАННЯ НА МОВІ ПРОГРАМУВАННЯ С++ ПРИ ВИВЧЕННІ СКЛАДНИХ ЕКОНОМІЧНИХ ЯВИЩ

8.080201 – „Інформатика"

А В Т О Р Е Ф Е Р А Т

магістерської дисертації на здобуття академічного ступеня магістра з інформатики

Комп'ютерний набір в редакторі Microsoft®Office® Word 2007 К.В.Ціхоцька

Редагування, верстка, макетування та дизайн Р.М.Літнарович.

Науковий керівник Р. М. Літнарович, доцент, кандидат технічних наук

Міжнародний Економіко-Гуманітарний Університет ім. акад. Степана Дем'янчука

Кафедра математичного моделювання

33027,м.Рівне,Україна

Вул.акад. С.Дем'янчука,4, корпус 1

Телефон:(+00380) 362 23-73-09

Факс:(+00380) 362 23-01-86

E-mail:mail@regi.rovno.ua

tkvpay@gmail.com