УДК 378.02:372.8

ПРАВОРСЬКА Н. І.
к. п. н., доцент
Хмельницький національний університет

ЛУЦКОВА Л. П.
к. е. н.
ПВНЗ "Хмельницький економічний університет"

ПІДХІД ДО ВДОСКОНАЛЕННЯ НАВИЧОК ПРОГРАМУВАННЯ СТУДЕНТАМИ ПІД ЧАС ВИКОНАННЯ КУРСОВИХ РОБОТ З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ "ПРОГРАМУВАННЯ" З МЕТОЮ РОЗВИТКУ СОЦІАЛЬНО-ЕКОНОМІЧНОЇ ОСВІТИ

У статті розкрито роль та завдання дисципліни "Програмування", принципи навчання програмуванню, доцільність використання творчих завдань, а саме курсових робіт, та змін, які торкнулися освітнього процесу. Визначено ефект від самостійного опанування навчального матеріалу. Наведено приклад розробки творчої роботи та етапи її створення.

Ключові слова: освітній процес, комп'ютери, програмування, творче завдання, курсова робота, програмний продукт, база даних, об'єктно-орієнтоване програмування.

ПРАВОРСКАЯ Н. И.
к. п. н., доцент
Хмельницкий национальный университет

ЛУЦКОВА Л. П.
к. э. н.
ПВУЗ "Хмельницкий экономический университет"

ПОДХОД К УСОВЕРШЕНСТВОВАНИЮ НАВЫКОВ ПРОГРАММИРОВАНИЯ СТУДЕНТАМИ ПРИ ВЫПОЛНЕНИИ КУРСОВЫХ РАБОТ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ "ПРОГРАММИРОВАНИЕ" С ЦЕЛЬЮ РАЗВИТИЯ СОЦИАЛЬНО-ЭКОНОМИЧЕСКОГО ОБРАЗОВАНИЯ

В статье раскрыты роль и задачи дисциплины "Программирование", принципы обучения программированию, целесообразность использования творческих заданий (курсовых работ) и изменений, которые коснулись образовательного процесса. Определен эффект от самостоятельного освоения учебного материала. Приведен пример разработки творческой работы и этапы ее создания.

Ключевые слова: образовательный процесс, компьютеры, программирование, творческое задание, курсовая работа, программный продукт, база данных, объектно-ориентированное программирование.

PRAVORSKA N. I.
candidate of pedagogical sciences, associate professor
Khmelnytsky National University

LUTSKOVA L. P.
candidate of economical sciences
Private Institution of Higher Education "Khmelnytsky Economical University"

AN APPROACH FOR IMPROVING THE PROGRAMMING SKILLS BY STUDENTS IN COURSEWORKS TO THE SUBJECT "PROGRAMMING" WITH PURPOSE OF DEVELOPING SOCIO-ECONOMIC EDUCATION

In the article, the role and tasks of the subject "Programming" are clarified, principles of teaching programming are stated, the appropriateness of use of creative tasks (coursework) and changes that have affected the educational process are substantiated. The effect of self-development over training material is determined. An example of creative work development and stages of its creation is given.

Keywords: educational process, computers, programming, creative assignment, coursework, software, database, object-oriented programming.

margana2007@list.ru

Актуальність і постановка задачі. Багато змін, що впливають на програмування, пов'язані з прогресом в технологіях. Більшість цих досягнень являють собою частину постійного еволюційного процесу, який продовжується вже багато років. Закон Мура (зроблений у 1965 році засновником "Intel" Гордоном Муром, в якому говориться про те, що щільність транзисторів на кристалі мікропроцесора буде подвоюватися кожні вісімнадцять місяців) до цього часу є істинним. В результаті, ми є свідками експоненціального росту обчислювальних можливостей, дякуючи яким стало можливим розв'язування задач, які здавалися нерозв'язними лише кілька років тому.

На комп'ютерну освіту також впливають зміни в культурному і соціальному контексті. Наприклад, всі з перерахованих нижче змін вплинули на природу освітнього процесу наступні:

• Зміни в педагогіці в результаті появи нових технологій. Технічні зміни, які призвели до розширення програмування, прямо впливають і на культуру навчання. Наприклад, комп'ютерні мережі зробили дистанційну освіту набагато більш доступною. Крім того, комп'ютерні мережі набагато полегшили спільне використання навчальних ресурсів географічно розподіленими інститутами. Технологія також впливає і на педагогіку. Демонстраційне програмне забезпечення, комп'ютерні проектори і персональні комп'ютери призвели до значних змін у навчанні програмування. Структура курсів з програмування повинна враховувати ці зміни в технології.

• Неочікувана швидкість розповсюдження комп'ютерів у всьому світі. Комп'ютери одержали надзвичайне розповсюдження протягом останніх десятиріч. Бурхливе розповсюдження комп'ютерних технологій призводить до великої кількості змін, що впливають на навчання, включаючи і загальне збільшення рівня освіченості студентів в галузі інформатики та програмування та їх прикладних задач. Однак в то же час збільшується розрив між рівнем знань тих, хто має доступ до сучасних комп'ютерних технологій, і тих, хто такого доступу не має.

• Зростаючий економічний вплив комп'ютерних технологій. Підвищений суспільний інтерес до індустрії високих технологій суттєво вплинув на освіту і ресурси, що для неї виділяються. Суттєвий попит на професіоналів в галузі інформатики та програмування і надія отримати прибуток в галузі програмування залучають все більшу кількість студентів до цієї сфери, в тому числі і тих, кому ці дисципліни взагалі не цікаві. В той же час, попит, що зростає, на спеціалістів з боку комерційних компаній ускладнив для більшості інститутів зацікавленість і утримання викладачів, тим самим значно обмежив можливості інститутів задовольняти потреби ринка у молодих спеціалістах.

• Визнання що постійно збільшується, інформатики та програмування як академічних дисциплін. В свої ранні роки інформатика та програмування були змушені відстоювати свою легітимність у багатьох навчальних закладах. Врешті це були нові дисципліни без глибоких історичних коренів, характерних для більшості академічних наук. В результаті впровадження комп'ютерних технологій в основні культурні і економічні аспекти нашого життя, боротьба за легітимність була виграна. У багатьох навчальних закладах інформатика, наприклад, стала однією з найактивніших дисциплін. Більше немає необхідності у відстоюванні викладання цих дисциплін у вищих навчальних закладах. Сьогодні основною проблемою є знаходження шляхів задоволення попиту на таке навчання.

• Розширення дисциплін. Інформатика та програмування не лише зросли і стали легітимними, а й значно розширили свої границі. Раніше інформатика та програмування, зазвичай, зводились до комп'ютерної науки. З роками, всі більше і більше галузей ставали часткою інформатики та програмування. Наприклад, безпека і криптографія, конкретні предметні галузі.

Спираючись на проведений аналіз навчальних планів і змін у дисципліні програмування, що викладено вище, нами сформульовані наступні принципи навчання:

1. Програмування – це широка галузь досліджень, яка не може бути зведеною до рамок комп'ютерної науки.

2. Програмування базується на цілому ряді дисциплін. Університетське навчання програмуванню вимагає від студентів використання концепцій з множини різних галузей. Всі студенти, що вивчають програмування, повинні вчитися об'єднувати теорію і практику, розуміти важливість спілкування і абстракції, а також цінити ефективні рішення.

3. Швидка еволюція комп'ютерної науки вимагає постійного перегляду навчальних планів. Враховуючи темп змін в програмуванні, оновлення навчальної програми раз в десять років вже не є нормальним. Повинен бути постійно організований процес перегляду типових навчальних планів, який дозволить оперативно поновлювати застарілі компоненти.

4. Під час розробки типових навчальних планів з програмування необхідно враховувати зміни в технологіях, нові розробки у галузі педагогіки, а також важливість навчання протягом всього життя. У такій галузі, як програмування, що розвивається стрімкими темпами, навчальні заклади повинні оперативно переймати передові стратегії, реагуючи на зміни, що відбуваються. Вищі навчальні заклади повинні не відставати від прогресу як в галузі технологій, так і у галузі педагогіки, не враховуючи існуючих обмежень у ресурсах. Крім того, навчання програмуванню в інституті повинно готувати студентів до подальшого навчання протягом всього життя, що дозволить їм просуватися в ногу з часом і бути спроможним розв'язувати складні проблеми майбутнього.

5. Необхідно створити технологічний стандарт, який не повинен обмежуватися описом розділів знань, необхідно також запропонувати набір знань та вмінь і систему задач.

6. Стандарт повинен визначити базисні навички і знання, якими повинні володіти всі студенти. Незважаючи на значне розширення комп'ютерної науки, існують концепції і навички, які є загальними для програмування в цілому. Необхідно зробити спробу визначити спільні теми дисципліни, і описати їх в рамках базової програми.

7. Набір обов'язкових для вивчення знань повинен бути зменшеним настільки, наскільки це можливо. По мірі розширення дисципліни програмування, кількість тем, що є обов'язковими для вивчення, суттєво збільшується. За останні десятиріччя програмування розрослося так, що тепер вже неможливо додавати нові теми без видалення старих.

Метою даної статті є відображення ефективності використання творчих завдань з новими для студентів знаннями. Тому, зважаючи на все вище згадане, необхідно намагатися навчити студентів використовувати самостійно набуті знання та навички, мати змогу знайти недостатню потрібну інформацію в світовій інформаційній мережі, підходити творчо до виконання запропонованих робот.

Студент отримує теоретичні знання протягом вивчення дисципліни «Програмування» на лекційних заняттях та може їх використати на лабораторних заняттях, але всі завдання надаються викладачем, який узгоджує правила та потрібний обсяг виконання цих робіт. Для того, щоб студент міг розвинути свої пізнавальні навички, уміння орієнтуватися в інформаційному просторі, критичне мислення, йому пропонують виконати курсову роботу. Виконуючі індивідуальне завдання, студент має вдосконалити всі набуті навички, оскільки не всі лабораторні роботи могли охопити наданий лекційний матеріал. В ході розв'язання конкретного практичного завдання студент вчиться самостійно застосовувати теоретичні знання, які він отримав. Під час написання програми для курсової роботи завжди планується рішення якоїсь проблеми, що передбачає, з одного боку, використання різноманітних методів, засобів навчання, а з іншого – інтегрування знань, умінь з різних галузей науки, техніки, технології, творчих сфер. Результати виконання цих індивідуальних завдань повинні бути «відчутними», тобто, якщо це теоретична проблема, то конкретне її рішення, якщо практична – конкретний результат, готовий до впровадження.

Під час написання програми для індивідуального завдання курсової роботи студент розв'язує не тільки відірвану від практики задачу, але й бачить конкретне застосування елементів уже розв'язаних на лабораторних роботах задач з програмування.

Приблизна тематика індивідуальних завдань до курсової роботи:

1. Розробка навчально-контролюючої програма з фізики на тему “Оптика”.
2. Розробка пакету процедур для знаходження екстремумів функцій.
3. Розрахунок тривалості виробничого циклу за різних видів переміщення предметів праці в процесі виробництва.
4. Розробка програми “Телефонний довідник”.
5. Розробка програми для розрахунку розходу деревини в столярній справі.
6. Розробка тест-програми “Визначення характеру людини”.

Тобто можна зрозуміти, що завдання можуть охоплювати різні сфери діяльності людини. Розглянемо приклад такого виконання курсової роботи на тему: “Опрацювання результатів сесії”. Метою даної курсової роботи є розробка комп’ютерної програми для опрацювання результатів сесії. Для виконання даного завдання необхідно розробити структуру бази даних, в якій надалі буде заповнюватись, редагуватись, видалятись, змінюватись інформація. Дану задачу будемо виконувати за допомогою Microsoft SQL Server. Коли структуру майбутньої бази даних розроблено, можна приступати до розробки програмного забезпечення, що буде складатись з рівнів доступу і роботи з базою даних. Дана програма буде складатись з двох панелей управління: панель адміністратора; панель викладача.

У кожній з них буде виконуватись різна робота з базою даних. Зокрема, адміністратор створює дисципліни, групи, додає викладачів та студентів і створює певні зв’язки між дисциплінами та викладачами, студентами та групами, та відповідні їм редагування. Також адміністратор може опрацьовувати результати сесії за різними критеріями. Панель викладача відображає групи та дисципліни, в яких він викладає, і відповідних студентів, для яких потрібно виставляти оцінки. Також в панелі викладача можлива функція формування відомості після виставлення усіх оцінок.

Для реалізації даного завдання обрано мову програмування C# та середовище розробки програмних продуктів Microsoft Visual Studio 2010. Для розробки структури бази даних та заповнення її інформацією обрано Microsoft SQL Server.

Програмний продукт складається з панелей управління:

1. Панель адміністратора (рис. 1):

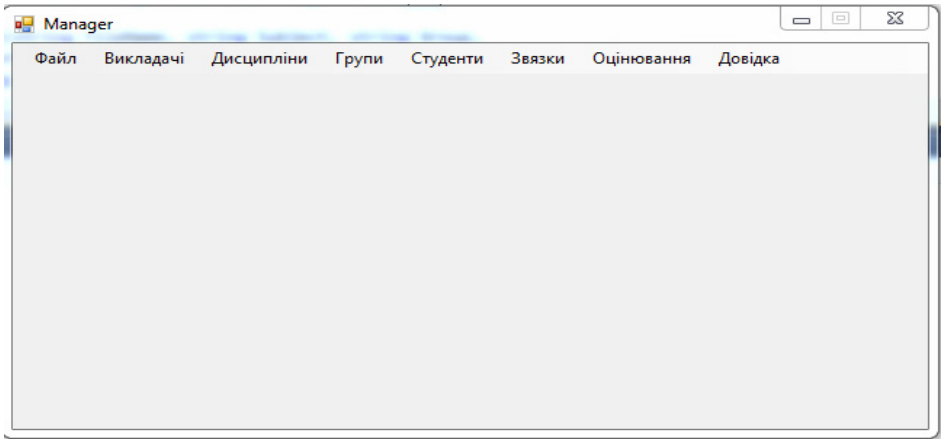

Рис. 1. Панель адміністратора

Панель адміністратора є головною, оскільки в ній виконуються основні дії з базою даних:

- додавання та редагування даних викладачів;
- формування та редагування дисциплін;
- створення та редагування груп;
- додавання та редагування даних студентів;
- встановлення зв’язків між групою, дисципліною та викладачем, в якій дисципліна

буде викладатися, та відповідне їх редагування; також можливість перегляду існуючих зв'язків;

• можливість опрацювання даних, формуванням відомостей із зазначенням їх критеріїв;
• можливість використання довідки.

2. Панель викладача (рис. 2):

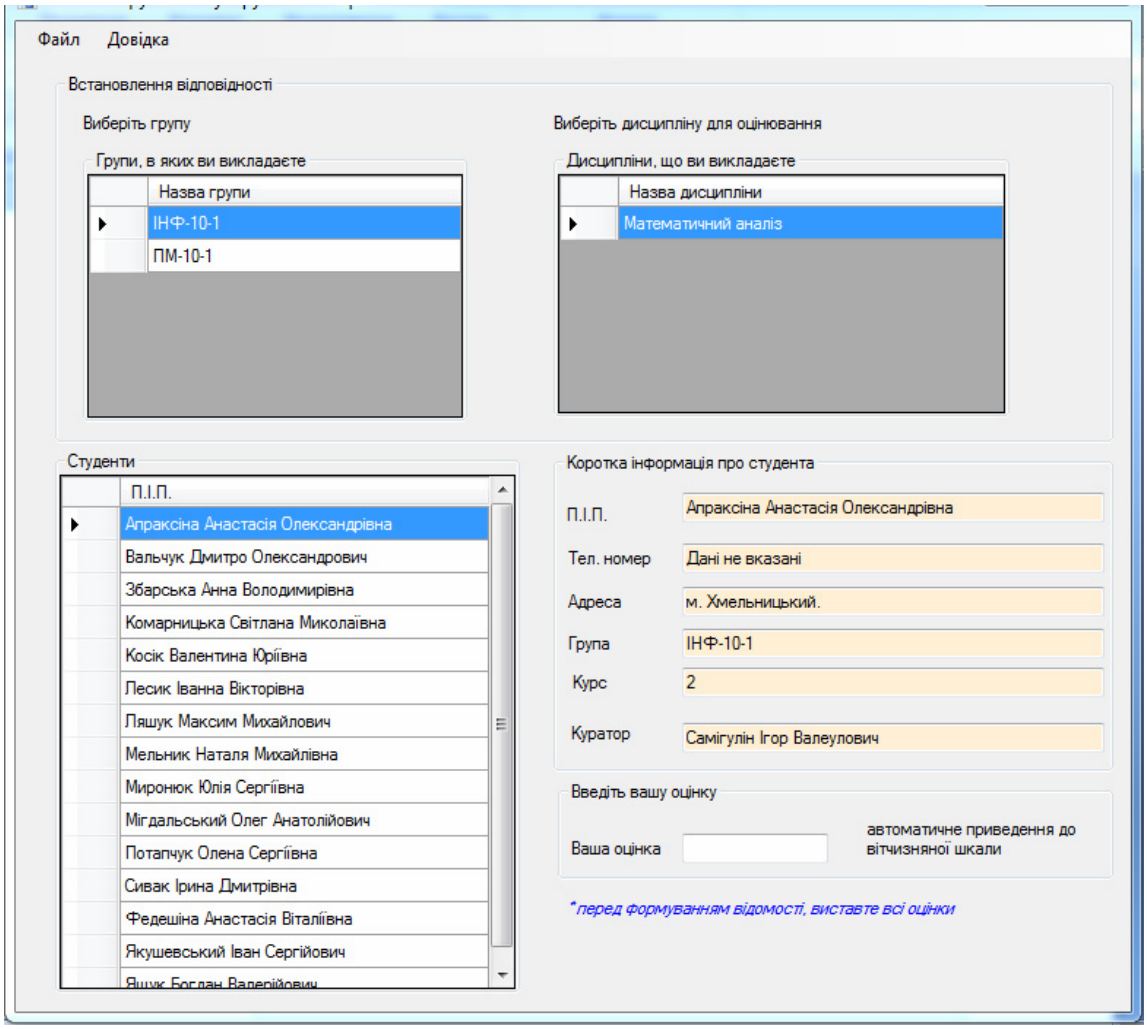

Рис. 2. Панель викладача

Панель викладача призначена для виставлення кінцевих результатів сесії та формування відповідного обліку успішності; також можливість використання довідки.

Залежно від авторизації (рис. 3), запускається відповідна панель управління.

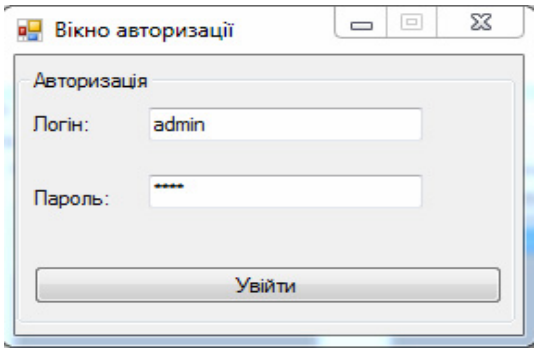

Рис. 3. Вікно авторизації

Тексти оригінальних функцій

Функція формування журналу успішності

```
public bool SaveDBToLocalFile(string FileName, string Subject, string Group,
            string Course, string TeacherID, string SubjectID,string GroupID)
        {
            string Teachers = "";
            ArrayList t = ShowTeachersForCurrectSubjectInCurrectGroup(GroupID,
SubjectID);
            for (int i = 0; i < t.Count; i++)
            {
                Teachers += t[i];
                if(i%2 == 0 && t.Count > 1)
                Teachers += ", ";
            }
            StreamWriter sw = new StreamWriter(new FileStream(FileName, FileMode.Create),
Encoding.GetEncoding(1251));

            sw.WriteLine(@""""";""Відомість оцінок"";""""");
            sw.WriteLine();
            sw.WriteLine(@"""з дисципліни: "";""{0}"";""Група: {1}""",Subject, Group);
            sw.WriteLine(@"""Викладачі: "";""{0}"";""Курс: {1}""", Teachers, Course);
            sw.WriteLine();
            sw.WriteLine(@"""№"";""П.І.П"";""Оцінка""");
            using (SqlConnection con = new SqlConnection(connectionString))
            {
                try
                {
                    con.Open();
                    string query = string.Format("Select
Students.FIO,Students.GroupID,Students.ID,Marks.StudentID,Marks.TeacherID From " +
                        "Students Join Marks ON Students.ID = Marks.StudentID And
Students.GroupID = '{0}' And TeacherID = '{1}' Order By FIO",
                        Guid.Parse(GroupID),Guid.Parse(TeacherID));
                    SqlCommand com = new SqlCommand(query, con);
                    SqlDataReader dr = com.ExecuteReader();

                    string[] data_StudentsID = new string[0];
                    string[] data_Marks = new string[0];
                    string[] data_Names = new string[0];
                    if (dr.HasRows)
                    {
                        foreach (DbDataRecord result in dr)
                        {
                            Array.Resize(ref data_StudentsID, data_StudentsID.Length +
1);
                            data_StudentsID[data_StudentsID.Length - 1] =
result["ID"].ToString();
                            Array.Resize(ref data_Names, data_Names.Length + 1);
                            data_Names[data_Names.Length - 1] = result["FIO"].ToString();
                        }
                    }
                    int c = 0;
                    dr.Dispose();
                    while (c < data_StudentsID.Length)
                    {
                        query = string.Format("Select Mark From Marks Where StudentID =
'{0}' "+
                        "And SubjectID =
'{1}'",Guid.Parse(data_StudentsID[c]),Guid.Parse(SubjectID));
                        com = new SqlCommand(query, con);
                        Array.Resize(ref data_Marks, c + 1);
                        data_Marks[c] = com.ExecuteScalar().ToString();
                        c++;
                    }
```

```
                    c = 0;
                    while (c < data_Names.Length)
                    {
                        sw.WriteLine(@"""{0}"";""{1}"";""{2}""", c + 1, data_Names[c],
data_Marks[c]);
                        c++;
                    }
                    c= 0;
                    sw.WriteLine();
                    query = string.Format("Select ID From Students Where GroupID =
'{0}'",Guid.Parse(GroupID));
                    com = new SqlCommand(query, con);
                    dr = com.ExecuteReader();
                    foreach(DbDataRecord result in dr)
                    {
                        c++;
                    }
                    sw.WriteLine(@"""Студентів в групі : "";""{0}""",c);
                    sw.WriteLine(@"""Кількість виставлених оцінок : "";""{0}""",
data_Marks.Length);

                    sw.WriteLine();
                    int five=0, four=0, three=0, two = 0;
                    for (int i = 0; i < data_Marks.Length; i++)
                    {
                        if (int.Parse(data_Marks[i]) == 5)
                            five++;
                        else if (int.Parse(data_Marks[i]) == 4)
                            four++;
                        else if (int.Parse(data_Marks[i]) == 3)
                            three++;
                        else
                            two++;
                    }

                    dr.Dispose();
                    sw.WriteLine(@"""Кількість п'ятірок"";""{0} ({1}%)""", five,
(double)(five * 100 / data_Marks.Length));
                    sw.WriteLine(@"""Кількість четвірок"";""{0} ({1}%)""", four,
(double)(four * 100 / data_Marks.Length));
                    sw.WriteLine(@"""Кількість трійок"";""{0} ({1}%)""", three,
(double)(three * 100 / data_Marks.Length));
                    sw.WriteLine(@"""Кількість двійок"";""{0} ({1}%)""", two,
(double)(two * 100 / data_Marks.Length));
                    sw.WriteLine();
                    sw.WriteLine(@"""Підпис викладача: """);
                    sw.Dispose();
                    return true;

                }
                catch
                {
                    sw.Dispose();
                    return false;
                }

            }
        }
```

Функція виставлення оцінок

```
public bool Teacher_InsertMark(string Mark, string StudentID, string SubjectID, string
TeacherID)
```

```
        {
            using (SqlConnection con = new SqlConnection(connectionString))
            {
                try
                {
                    con.Open();
                    float t = float.Parse(Mark);
                    CheckMark(ref t);
                    string query = string.Format("Insert Into
Marks(ID,Mark,StudentID,SubjectID,TeacherID,AddedDate) " +
                        "Values('{0}','{1}','{2}','{3}','{4}','{5}')",
Guid.NewGuid().ToString(), t, Guid.Parse(StudentID),
                        Guid.Parse(SubjectID), Guid.Parse(TeacherID),
DateTime.Now.ToShortTimeString());

                    SqlCommand com = new SqlCommand(query, con);
                    if (com.ExecuteNonQuery() == 1)
                        return true;
                }
                catch
                {
                    return false;
                }
            }
            return false;
        }
```

Функція редагування оцінки

```
public bool Teacher_UpdateMark(string Mark, string StudentID, string SubjectID, string
TeacherID)
        {
            using (SqlConnection con = new SqlConnection(connectionString))
            {
                try
                {
                    float t = float.Parse(Mark);
                    CheckMark(ref t);
                    con.Open();
                    string query = string.Format("Update Marks Set Mark =
Convert(float,{0}) Where " +
                        "StudentID = '{1}' And SubjectID = '{2}' And TeacherID =
'{3}'",t, Guid.Parse(StudentID),
                        Guid.Parse(SubjectID), Guid.Parse(TeacherID));

                    SqlCommand com = new SqlCommand(query, con);
                    if (com.ExecuteNonQuery() == 1)
                        return true;
                }
                catch
                {
                    return false;
                }
            }
            return false;
        }
```

Лістинг класу форми «Опрацювання результатів сесії»

```
using System;
using System.Collections.Generic;
```

```
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;

namespace Курсова
{
    public partial class ResultOfSession : Form
    {
        DAL dal = new DAL();
        bool first = true;
        string whatMark = "";
        public ResultOfSession()
        {
            InitializeComponent();
            cmbx_Teachers.DataSource = dal.AllTeachers();
            cmbx_Groups.DataSource = dal.AllGroups();
            cmbx_WhatMarks.SelectedIndex = 0;
        }

        private void btn_Enter_Click(object sender, EventArgs e)
        {
            TeacherForm teacherForm = new
TeacherForm(dal.GetTeacherID(cmbx_Teachers.SelectedValue.ToString()),true);
            teacherForm.ShowDialog();
            teacherForm.Dispose();
        }

        private void cmbx_Groups_SelectedIndexChanged(object sender, EventArgs e)
        {
            bool changes = false;
            cmbx_Subjects.DataSource =
dal.ShowSubjectsForCurrectGroup(dal.GetGroupID(cmbx_Groups.SelectedValue.ToString()),ref
changes);
            if (!changes)
            {
                cmbx_Subjects.ResetText();
                cmbx_CurrectTeachers.ResetText();
            }
            if (first)
            {
                cmbx_Subjects_SelectedIndexChanged(new object(), new EventArgs());
                first = false;
            }
            if (!check_Before_Create_Journal())
                this.btn_CreateMarks.Enabled = false;
            else
                this.btn_CreateMarks.Enabled = true;
        }
        private void cmbx_Subjects_SelectedIndexChanged(object sender, EventArgs e)
        {
            cmbx_CurrectTeachers.DataSource =
dal.ShowTeachersForCurrectSubjectInCurrectGroup(
                dal.GetGroupID(cmbx_Groups.SelectedValue.ToString()),
dal.GetSubjectID(cmbx_Subjects.SelectedValue.ToString()));
        }

        private void btn_CreateMarks_Click(object sender, EventArgs e)
        {
            saveFileDialog.Filter = "Exel File (*.csv)|*.csv";
            if (saveFileDialog.ShowDialog() == System.Windows.Forms.DialogResult.OK)
            {
```

```
                if (string.Equals(whatMark,"Усі"))
                {
                    if (dal.SaveDBToLocalFile(saveFileDialog.FileName,
cmbx_Subjects.SelectedValue.ToString(),
                        cmbx_Groups.SelectedValue.ToString(),
dal.GetCourse(dal.GetGroupID(cmbx_Groups.SelectedValue.ToString())),
                        dal.GetTeacherID(cmbx_CurrectTeachers.SelectedValue.ToString()),
dal.GetSubjectID(cmbx_Subjects.SelectedValue.ToString()),
                        dal.GetGroupID(cmbx_Groups.SelectedValue.ToString())))
                    {

                    }
                }
                else
                {
                    if (dal.SaveDBToLocalFile(saveFileDialog.FileName,
cmbx_Subjects.SelectedValue.ToString(),
                        cmbx_Groups.SelectedValue.ToString(),
dal.GetCourse(dal.GetGroupID(cmbx_Groups.SelectedValue.ToString())),
                        dal.GetTeacherID(cmbx_CurrectTeachers.SelectedValue.ToString()),
dal.GetSubjectID(cmbx_Subjects.SelectedValue.ToString()),
                        dal.GetGroupID(cmbx_Groups.SelectedValue.ToString()),
Convert.ToSingle(whatMark)))
                    {

                    }
                }
            }
        }

        private void cmbx_WhatMarks_SelectedIndexChanged(object sender, EventArgs e)
        {
            whatMark = cmbx_WhatMarks.Items[cmbx_WhatMarks.SelectedIndex].ToString();
        }
        private bool check_Before_Create_Journal()
        {
            if (cmbx_Subjects.SelectedIndex < 0)
                return false;
            else
                return true;
        }
    }
}
```

Під час роботи над курсовою було використано середовище створення програмних продуктів Microsoft Visual Studio 2010 та мову об'єктно-орієнтованого програмування C#.

Результатом розробки стала система управління «Деканат», яка має можливості опрацювання результатів сесії. В програмі була використана технологія ADO.NET, яка дозволяє працювати з базами даних, модель якої представлена на рис. 4 у досить зручному вигляді. Місцем зберігання бази даних став Microsoft SQL Server. Виконуючи цю роботу, необхідно пригадати як проводилася розробка таблиць баз даних та зв'язати їх з формою. Перевагами даного продукту є простий в розумінні інтерфейс, не надлишковість функцій, що робить програму зручною, а головне практичною у використанні.

Програма має бути розміщена на комп'ютері, який в подальшому буде сервером для бази даних, на комп'ютері має бути встановлений Microsoft SQL Server для коректної роботи програми.

Тому, виконавши запропоновані завдання, студенти не тільки закріплять набуті навички та згадають пройдений матеріал з інших інформаційних дисциплін, які передували програмуванню, а також зможуть набути нових творчих навичок при розробці власного програмного продукту.

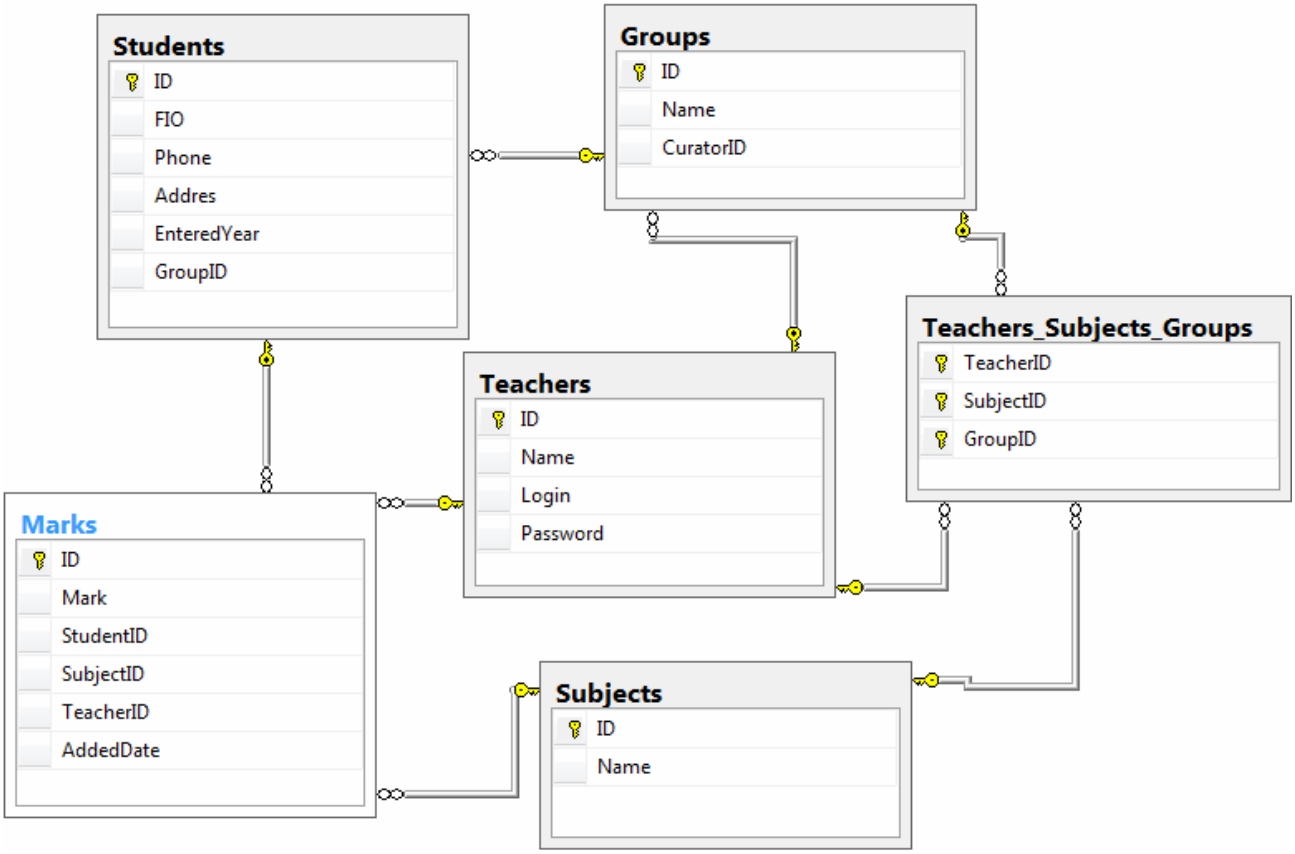

Рис. 4. Інфологічна модель бази даних

«Процес складання програм для цифрової обчислювальної машини особливо привабливий тим, що він не тільки дає економічні та наукові результати, але й надає естетичні переживання, в чомусь близькі переживанням, які відчуваються під час написання віршів та музики» [5, с.8].

Тому треба визнати, що процес самостійного опанування матеріалом і творчого підходу до виконання завдань дає змогу студентам, визнати що «заняття програмуванням відповідають глибинній внутрішній потребі в творчості та задовольняють чуттєві потреби, які є у всіх нас, надаючи чоти види радості:

• радість, яку отримують створюючи що-небудь своїми руками;

• задоволення створювати речі, які можуть бути корисними іншим людям;

• чарівність створення складних головоломних об'єктів, які складаються з частин, які рухаються взаємодіючи;

• радість, яку отримують від незмінного пізнання нового, неповторності задачі» [2, с.212].

Висновки. Використання індивідуальних завдань, а саме виконання курсових робіт з програмування, дозволяє суттєво поглибити розуміння студентами навчального матеріалу, посилити мотивацію до навчання, активізувати навчальну діяльність, надати навчанню творчого, дослідницького спрямування, розширити теоретичну базу знань, а також посилити прикладну спрямованість результатів навчання програмуванню у ВНЗ за рахунок:

1) включення до індивідуальних завдань задач практичного змісту, особливе значення серед яких надається задачам на пошук оптимальних рішень;

2) використання індивідуальних завдань, розв'язання яких стимулює розвиток як абстрактного, так і образного мислення, завдяки використанню комп'ютерної графіки та знаково-символьного подання об'єктів;

3) суттєвого посилення міжпредметних зв'язків на основі інтегруючого характеру завдань, що потребують використання інформаційного моделювання;

4) зміщення акценту щодо навчання програмування у ВНЗ до розширення завдань на формування професійно-орієнтованих умінь (пошук, систематизація необхідної інформації та її опрацювання, формулювання узагальнюючої інформації за результатами використання інформаційних технологій, використання математичного апарату, встановлення зв'язків між відповідними величинами, складання плану розв'язання поставленої задачі тощо), створення інформаційних моделей різних процесів і явищ та дослідження їх за допомогою засобів сучасних інформаційних технологій, що сприяє вирішенню проблем базових рівнів знань в конкретних предметних галузях.

Список використаних джерел

1. Байдачный С.К. "C# для новичков" / Байдачный С.К. – КУДИЦ-ПРЕСС, 2011. – 856 с.

2. Брукс Ф. Мифический человек-месяц или как создаются программные системы / Брукс Ф. ; [пер. с англ.]. – СПб :Символ Плюс, 2001. – 304 с.

3. Ватсон Карли «C# для профессионалов» / Ватсон Карли. – М. : «Лори», 2005. – 478 с.

4. Вирт Н. Алгоритмы+Структуры данных=Программы / Вирт Н. ; [пер. с англ.]. – М. : Мир, 1985. – 406 с.

5. Кнут Д. Искусство программирования для ЭВМ : т. 1. Основные алгоритмы / Кнут Д. – М. : Мир, 1976. – 736 с.

6. Троелсен Э. Язык программирования C# 2010 и платформа .NET 4.0 / Троелсен, Эндрю. – 5-е изд. – М. : Вильямс. – 1392 с.