

Упражнение №4 по ПС

Извеждане на студентска информация. Стандартни диалози.

Целта на упражнението е студентите да затвърдят знанията си за базите данни и ползването им в информационни системи. Ще бъдат разгледани стандартните диалогови прозорци – OpenFileDialog, FontDialog, ColorDialog и SaveFileDialog, както и техните основни свойства. Ще се научат да съхраняват генериран от програмата им форматиран текстов файл.

Извеждане на студентската информация.

Ако сте завършили домашното от упражнение 3, трябва да разполагате във вашата база данни с таблица Students. Тя ще съдържа студентската информация за всеки от студентите, която трябва да изведем в полетата на MainForm след успешно вписване.

1. Създайте клас Student описващ таблица Students (както създадохте класа User в предното упражнение).
2. Модифицирайте си метода за извеждане на студентската информация от класа MainForm, така че да приема параметър от тип Student и да извежда информацията записана в него. (В момента този метод извежда статични имена).
3. В Data слоя създайте статичен метод, който да връща данните за студент с даден факултетен номер. Ако такъв студент не е намерен да се връща *null*.
4. В слоя Logic направете метод (най-добре в отделен клас), който да вика горния метод. Например:

```
using StudentInfoSystem.Data;  
  
namespace StudentInfoSystem.Logic  
{  
    class StudentValidation  
    {  
        public static Student IsThereStudent(User usr)  
        {  
            return UserData.GetStudentDataByUser(usr.FacNumber);  
        }  
    }  
}
```

5. В класа LoginForm добавете свойство user (дадено отдолу). В конструктора на класа му задайте начална стойност *null*.

```
public User user { get; private set; }
```

6. Помислете къде в класа MainForm трябва да се извърши вземането на данните за студента. Вземете под внимание, че метода който трябва да извикате IsThereStudent(User usr) има нужда от данни за потребителя, а те се намират в полето user на Login-формата.

Направеното до тук не е достатъчно. Тъй като все още не сме написали определянето на стойността на полето `user` в класа `LoginForm`. За да видим как може да бъде променяна стойност на параметър ще разгледаме използването на `out` параметри.

7. Променете методите `_IsUserPassCorrect` и `IsUserPassCorrect`, така че да връщат като резултат цялата информация за потребителя (данни от тип `User`). Ако такъв потребител не е открит, нека метода връща стойност `null`.
8. Променете типа на `queryResult` в метода `ValidateUserInput`.
9. Като параметър на метода `ValidateUserInput` добавете параметър **out** `User user`. След горните точки кода на метода ви трябва да наподобява:

```
public bool ValidateUserInput(out User user)
{
    .....(вашия код си стои, променяте само жълтото).....

    //database proverka
    User queryResult = UserData.IsUserPassCorrect(_username, _password);
    // returns the user
    if (queryResult == null)
    {
        errText += "Въвели сте грешно потребителско име или парола!";
        return false;
    }
    user = queryResult;
    return true;
}
```

10. Променете викането на валидацията в `LoginForm`, така че `out` параметърът да се запази в полето `user`. Примерен код:

```
LoginValidation lv =
    new LoginValidation(txtUserName.Text, txtPassword.Text);
User usr;
if (lv.ValidateUserInput(out usr))
{
    //MessageBox.Show(lv.errText);
    this.user = usr;
    this.DialogResult = DialogResult.OK;
    this.Close();
}
else
{
    MessageBox.Show(lv.errText);
}
```

Стандартни диалогови прозорци.

OpenFileDialog

11. Добавете към вашата `MainForm` един `OpenFileDialog`. Преименувайте го на `openPictureDialog`.
12. Едно от свойствата позволява да направите филтър на файловете. За да можете

да избирате само изображения задайте на свойството Filter стойност Image Files (*.BMP;*.JPG;*.GIF)|*.BMP;*.JPG;*.GIF|All files (*.*)|*.*

13. В менюто на MainForm добавете елемент, който да служи за извикването на горната форма (Open Picture). Направете така, че той да е активен само когато има вписан потребител.
14. Добавете Click метод на елемента от точка 13.
15. Създайте следния метод и направете обръщение към него.

```
private void LoadStudentPicture()
{
    if (openPictureDialog.ShowDialog() == DialogResult.OK)
    {
        pbStudentPhoto.ImageLocation = openPictureDialog.FileName;
    }
}
```

16. Тествайте дали работи правилно.

FontDialog и ColorDialog

17. Добавете нова визуална форма към вашия проект с име InfoForm. В нея поставете 2 бутона – btnFont и btnColor, както и един richTextBox контрол. (Подчертавам има разлика между класовете RichTextBox и TextBox).
18. Направете втори конструктор на класа, който приема параметър string текст, за да инициализира съдържанието на richTextBox контролата.
19. Създайте метода за Click на бутона за промяна на фонта:

```
private void btnFont_Click(object sender, EventArgs e)
{
    if (fontDialog.ShowDialog() == DialogResult.OK)
    {
        if (richTextBoxText.SelectedText == "")
        {
            richTextBoxText.Font = fontDialog.Font;
        }
        else
        {
            richTextBoxText.SelectionFont = fontDialog.Font;
        }
    }
}
```

20. Напишете сами метода за промяна на цвета на маркирания/целия текст. Той е подобен на горния, само съобразете че свойството за цвета на текста в дадена контрола се нарича ForeColor.
21. В основната форма (MainForm), добавете извикване на инстанция на InfoForm, чрез натискане на бутон от менюто. Той също трябва да е активен само при вписан потребител.

```
private void miUverenie_Click(object sender, EventArgs e)
{
    string text = GenerateInfoText();
    InfoForm infoForm = new InfoForm(text);
    infoForm.Show();
}
```

22. Напишете метода `GenerateInfoText()`. Той трябва да сформира текста на уверението чрез данните за студента. Може да ползвате по-долния код и да го довършите за домашно:

```
private string GenerateInfoText()
{
    string text = "Уверение\n";
    text += txtFirstName.Text + " " + txtSecondName.Text + " " +
        txtLastName.Text;
    text += " е студент/ка във Факултет " + txtFaculty.Text +
        ", специалност " + txtSpecialty.Text;
    text += "\n";
    return text;
}
```

23. Пробвайте дали формата ви се отваря и дали бутоните за смяна на шрифт и цвят работят.

SaveFileDialog

24. Към формата `InfoForm` добавете диалог за запаметяване на файл - `saveFileDialog`.
25. Към формата `InfoForm` добавете бутон – `btnSave`. Създайте `Click` метода му.

```
private void btnSave_Click(object sender, EventArgs e)
{
    saveFileDialog.DefaultExt = "*.rtf";
    saveFileDialog.Filter = "RTF Files|*.rtf";
    if (saveFileDialog.ShowDialog() == System.Windows.Forms.DialogResult.OK
        && (saveFileDialog.FileName.Length > 0))
    {
        try
        {
            // Save the contents of the RichTextBox into the file.
            richTextBoxText.SaveFile(saveFileDialog.FileName);
        }
        catch (Exception)
        {
            MessageBox.Show("Файлът не беше съхранен, поради грешка в записа!",
                "Грешка", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
```

26. Тествайте.

Допълнително информация за тези, които се чудеха за Enter.

За да настроите някой бутон да се избира с `enter` на вашата форма трябва да направите следното:

От свойствата на диалоговата ви форма (най-вероятно `LoginForm`) намерете

AcceptButton и му задайте стойност името на желания бутон.

Ако искате някой бутон да се натиска при натискане на Esc на клавиатурата аналогично на горното свойство отговарящо за това е CancelButton.

За домашно:

Направете бутоните за форматиране и запис да съдържат икона и текст.

Опитайте се да добавите форматиране на положението на текста – подравняване в ляво или дясно, центриране.

Важно: Запишете си направеното в часа на USB флаш памет и на следващото упражнение си я носете, за да продължите проекта.