**Вступ**

При розробці системи для музичної школи була використана уніфікована мова моделювання UML.

Мова UML є інструментом візуального моделювання систем, тому основним за­собом її використання вважаються діаграми. Кожна з діаграм віддзеркалює певний аспект системи, а вся множина діаграм - становить структурну основу проекту.

Обєктно-орієнтована методологія опирається на систему діаграм – одиничних описів фрагментів системи. Різні типи діаграм відображають різні аспекти системи. У кожній діаграми є своя мета і своя аудиторія.

В UML інтегровані різноманітні відомі засоби візуального моделювання, які добре зарекомендували себе на практиці, зокрема, забезпечується можливість опису двох визначальних видів об'єктних моделей:

• структурних (або статичних) моделей – описується структура сутностей системи, включаючи класи, інтерфейси, відношення, атрибути;

• моделей поведінки (або динамічних моделей) – описується поведінка (функціонування) об'єктів системи, включаючи методи, взаємодію, процес зміни станів окремих компонент чи всієї системи.

• Надати користувачу засоби візуального моделювання систем різного призначення з акцентуацією на можливості їх розробки та отримання документації. (UML містить як абстрактні конструкції для представлення моделей, так і цілком конкретні, які дозволяють описувати деталі реалізації програмних систем).

• Забезпечити користувачів засобами розширення та специфікації з метою більш точного опису конкретних предметних областей. (Хоча у більшості випадків для побудови моделей цілком достатньо базових конструкцій UML, все ж в UML уведено механізм розширення базових понять. Крім того, можлива спеціалізація базових понять, шляхом доповнення останніх новими атрибутами чи властивостями).

• Підтримувати таку специфікацію моделей, яка, з одного боку, була б незалежною від конкретних мов програмування і, з іншого боку, забезпечувала б потенційні можливості реалізації у таких мовах.

Діаграма UML – це зображення у вигляді графа з вершинами (сутностями) і ребрами (відношеннями).

Основна мета діаграм – візуалізація архітектури розроблюваної системи з різних точок зору.

Діаграма – деякий зріз системи. Звичайно діаграми дають згорнуте представлення елементів, із яких складається розроблювана система. При цьому один і той самий елемент може бути присутнім у декількох (а іноді й в усіх) діаграмах.

При візуальному моделюванні з UML використовуються вісім видів діаграм, кожна з яких може містити елементи певного типу. Типи можливих елементів і Діаграми прецедентів або діаграми використання (use case diagrams). Такі діаграми описують функціональність, яка буде надаватись користувачам системи, котра проектується. Представляються шляхом використання прецедентів та акторів, а також відношень між ними. Набір усіх прецедентів діаграми фактично визначає функціональні вимоги, за допомогою яких може бути сформульоване технічне завдання.

Діаграми класів (class diagrams) описують статичну структуру класів. Дозволяють (на концептуальному рівні) формувати "словник предметної області" та (на рівні специфікацій і рівні реалізацій) визначати структуру класів у програмній реалізації системи. Можуть використовуватись для генерації каркасного програмного коду (в реальній мові програмування).

Для опису динаміки використовуються діаграми поведінки (behavior diagrams), що підрозділяються на

* + діаграми станів (statechart diagrams);
	+ діаграми діяльності (активності) (activity diagrams);
	+ діаграми взаємодії (interaction diagrams), що у свою чергу підрозділяються на
		- діаграм послідовності (sequence diagrams);
		- діаграм кооперації (співробітництва) (collaboration diagrams).

 І, нарешті, діаграми реалізації (implementation diagrams) поділяються на

* + компонентні діаграми (діаграми компонентів) (component diagrams);
	+ діаграми розгортання (deployment diagrams).

Спочатку для програмної системи серією діаграм прецедентів визначається її зовнішня функціональність (виділяються всі прецеденти та актори, а також відношення між ними).

Уся подальша робота над проектом має здійснюватись на основі прецедентів: для кожного прецеденту формується опис його динаміки у вигляді серії діаграм взаємодії та діаграм діяльності.

З отриманих описів шляхом виявлення об'єктів, що задіяні в реалізації прецедентів, будуються діаграми класів.

Для визначення поведінки класів із складною динамікою реагування на події можуть формуватись діаграми станів.

Розміщення об'єктів по програмних модулях описується в компонентних діаграмах, а розміщення програмних модулів по вузлах комп'ютерам мережі – у діаграмах розгортання.

**1.Вибір життєвого циклу**

При розробці системи для музичної школи для досягнення оптимального результату вибраною є модель життєвого циклу при якій була б можливість повернутися на певний попередній етап розробки при виникненні певних помилок, що дало б можливість швидкого їх виправлення (ітераційна модель). В результаті це б дало можливість як найкраще, і як найшвидше виконати поставлене завдання.

Дану модель можна представити в графічному вигляді таким чином:

Анал

і

з вимог

Проектування

Кодування

Тестування

Рис.1- Графічне зображення моделі життєвого циклу.

Рис.1- Графічне зображення моделі життєвого циклу.

Дана модель не є повністю досконалою, але вона повністю підходить для вирішення поставленої задачі.

**2. Аналіз аналогів**

Перед початком розробки системи я пробував найти існуючі аналоги, які використовуються в сучасних музичних школах для оптимізації навчального процесу.

Проаналізувавши існуючі системні рішення я дійшов до висновку, що на сьогоднішній день в Україні не існує такої системи, яка б на основі однієї програмної розробки виконувала б всі поставленні завдання. Так в музичних школах для задоволення потреб всіх користувачів використовується ціла низка програмного забезпечення,але інколи буває що воно відсутнє.

Для ведення обліку учнів і гуртків в більшості випадків використовується програма Microsoft Office Access, з пакету Microsoft Office, або аналоги, що дозволяють створювати бази даних.

Складання розкладу занять виконуються спочатку без програмного забезпечення, лише після його повної розробки, він переноситься в електронний вигляд для розповсюдження. Для цього також використовується найрізноманітніше програмне забезпечення.

Для обміну даними з міжнародними організаціями, іншими музичними школами, та для розсилання запрошень, також використовуються різноманітні програмні засоби для обробки електронної пошти.

Система, яка є розроблена під час виконання даного курсового проекту частково вирішує існуючі програмні рішення і є оптимальною для організації роботи музичної школи.

**3.Моделювання системної архітектури засобами UML**

**3.1. Аналіз предметної галузі**

 Під час розробки системи для роботи музичної школи спочатку було визначено всіх потенційних користувачів даної системи і визначено всі їх вимоги до даної системи. Таким чином серед прецедентів, які мають відношення до музичної школи можна виділити:

* Директор
* Викладач
* Керівник музичного гурту
* Комісія
* Учні

 Далі було проаналізовано всі їх вимоги від системи, які можна структуризувати наступним чином:

Учень

Всі точки зору

Колектив школи

Директор

Керівник муз.гурта

Комісія

Викладач

Сервіс

Перегляд розкладу, Редагування баз даних гуртків,оціню вання учнів, редагування баз даних музичних творів,участь в музичному гурті

Сервіс

Перегляд розкладу, прийняття та виключення учнів, організація концертів, розсилання запрошень,рекламна компанія

Сервіс

Контроль успішності, оновлення інвентаря,прийняття та звільнення викладачів, організація концертів редагування груп.

Сервіс

Ознайомлення з рівнем підготовки викладачів, перевірка стану інструментів, ознайомлення з успішністю учнів

Сервіс

Ознайомлення з списком гуртків, Перегляд розкладу, запис в гурток, прийняття участі в концерті,перегляд баз даних музичних інструментів

 Рис.2. Ієрархія точок зору усіх користувачів системи.

Для графічного відображення вимог кожного користувача системи було використано уніфіковану мову моделювання UML. За допомогою якої було створено діаграми прецедентів, що і відображає їх вимоги (діаграма прецедентів — в UML, діаграма, на якій зображено відношення між акторами та прецедентами в системі. Також, перекладається як діаграма варіантів використання).



 





Рис.3 - Діаграми прецедентів: учня,викладача,директора,комісії,керівника муз.гурту

На першому періоді розробки системи було визначено місце системи в середовищі:

 Рис.4. - Структурна схема моделі системного середовища.

При реалізації системи було змоделювано ряд стандартних процесів, які відбуваються в музичній школі, серед яких можна виділити процес прийняття учня в музичну школу, процес організації концертів, процес формування нових гуртків.

Кінець

Початок

БД

викладачів

Проведення концерту

Вибір відповідального за концерт

БД

гуртків

Запрошення викладацького складу

Реклама концерту

Репетиція

Налаштування інструментів

Отримання потрібного інвентаря

Підбір музичних творів

Формування сценарію концерту

Час і дата концерту

Місце проведення концерту

Підбір учнів для участі

БД

інвентаря

БД

концертів

БД

муз.творів

 Рис.5 – Модель процесу організації концерту в музичній школі.

БД

інвентаря

БД

успішності

БД

гуртків

Кінець

Формування звіту про перевірку

Святковий концерт

Перевірка інвентаря

Прибуття комісії

Відвідування гуртка

Ознайомлення з загальною успішністю учнів

Прослуховування учнів

Перевірка рівня підготовки викладачів

Прослуховування викладачів

Початок

 Рис.6 – Модель процесу перевірки музичної школи комісією.

Кінець

Отримання потрібного музичного інвентаря

БД

інвентаря

БД

гуртків

Реєстрація

Знайомство з викладачем

Написання заяви

Пробне заняття

Перевірка на вільне місце в гуртку

Перевірка вмінь учнів

Ознайомлення з списком гуртків

Початок

 Рис.7 – Модель процесу прийняття учня в музичну школу.

При реалізації даної системи використовується ряд баз даних:

* Гуртків; - Успішності учнів; - Інвентаря;
* Музичних творів; - Концертів; - Викладацького складу;
* Учнів.

Кожна база даних зберігає відповідну інформацію, і містить поля за допомогою яких вона зв`язується з іншими базами даних, що забезпечує можливість реалізації системи.

 Для відображення сутності та зв`язків між базами даних можна використати наступну діаграму:

БД Викладацького складу

Id викладача

Прізвище викладача

Імя

По батькові

Освіта

Дата народженя

Професійні досягнення

Характеристика

БД музичних творів

Назва

Автор

Текст

Ноти

БД інвентаря

Інвентарний номер

Назва

Дата отримання

Дата випуску

Інвентарна вартість

БД концертів

Дата проведення

Місце проведення

Організатор

Учасники

Інвентар

БД успішності учнів

Id учня

Прізвище учня

Імя

По батькові

Назва предмету

Оцінка за відповідний предмет

Рейтинг

База даних Учнів

Id учня

Прізвище учня

Імя

По батькові

Освіта

Дата народження

База даних гуртків

Id гуртка

Назва гуртка

Інвентар

Дні проведення занять

Аудиторія

Інформація про учня

 Рис.8. – Модель сутності баз даних, та зв`язків між ними.

**3.2. Розробка архітектури програмного забезпечення**

Наступним етапом в розробці системи є розробка архітектури програмного забезпечення. На цьому етапі визначається структуру системи, та структуру керування та взаємодії складових системи.

Для реалізації структури системи було використано одну із стандартних моделей – модель репозиторій. Яка забезпечує доступ до потрібної інформації кожному користувачу. Всі дані зберігаються на обному більш-менш потужному компютері, що дозволяє централізувати роботу, і економити при виборі параметрів машин клієнтів, що здешевлює реалізацію системи.



 Рис.9. Структура системи для роботи музичної школи (модель репозиторій) . Для реалізації системи роботи музичної школи було обрано модель , що керуються подіями. А саме модель передачі повідомлень.



Рис.10- Модель керування.

**4. Тестування програмного забезпечення**

**4.1. Теоретична частина**

Тестування програмного забезпечення - це процес аналізу або експлуатації програмного забезпечення з метою виявлення дефектів.

Тестування передбачає "аналіз" або "експлуатацію" програмного продукту. Тестова діяльність, що пов'язана з аналізом результатів розробки програмного забезпечення, називається статичним тестуванням. Воно передбачає перевірку програмних кодів, контроль та перевірку програми без запуску на комп'ютері. Тестова діяльність, що передбачає експлуатацію програмного продукту, називається динамічним тестуванням. Динамічне та статичне тестування доповнюють одне одного.

Мета розробника полягає в тому, щоб зробити програмний код без дефектів, який відповідає призначенню програмного продукту та відповідає вимогам замовника. Мета тестувальника пов'язана з аналізом коду та експлуатації програми, що у результаті повинно призвести до виявлення дефектів, які проявляються під час його інтегрування, конфігурування та виконання в різних середовищах.

Тестування програмного забезпечення виконує дві базові функції: верифікацію та атестацію. Верифікація забезпечує відповідність результатів конкретної фази процесу розробки вимог даної та попередньої стадії. Атестація є гарантією того, що програмний продукт задовольняє системним вимогам.

Програмний продукт є якісним, коли:

* під час роботи користувача з програмним продуктом виникає невелика кількість відмов;
* програмний продукт надійний, а це означає, що його використання рідко викликало аварійні відмови;
* програмний продукт задовольняє вимогам більшості користувачів.

Процес тестування є дуже важливим при розробці програмного забезпечення. Адже на цьому етапі можна ще до введення в експлуатацію виявити помилки в його роботі. На сьогоднішній день цьому процесу приділяють надзвичайно велику увагу всі розробники, адже він допомагає економити і підтримувати імідж фірми.

Тому розроблено велику кількість методів тестування, серед яких :

* Тестування на відмову;
* Структурне тестування ("білого ящика");
* Тестування "чорного ящика";
* Перевірка на модель;
* Та інші.

Для тестування розробленого програмного забезпечення в даному курсовому проекті будуть використані саме такі методи тестування.

Будь яке тестування повинно складатися з наступних елементів і процесів, тобто виконуватись за наступною схемою:



Рис. 12 – Схема процесу тестування.

**4.2.Тестування "чорної скриньки"**

При тестуванні "чорної скриньки" (англ. black-box testing), тестер має доступ до ПЗ тільки через ті ж інтерфейси, що й замовник або користувач, або через зовнішні інтерфейси, що дозволяють іншому комп'ютеру або іншому процесу підключитися до системи для тестування.

Для тестування було обрано завдання на курсову роботу.

Вимогами до даної програми є зрозумілість в користуванні, зрозуміле діалогове вікно при введенні даних і виводі результату.

Для проведення тестування було запущено файл Musical\_school.exe.



Рис.13 – Результат виконання програми.

Аналіз результату. Діалогове вікно для роботи має ряд переваг: при запуску програми вікно є зрозумілим, діалог з користувачем ведеться на українській мові і що є зручним для користувача. При некоректному вводі даних програма продовжує свою роботу видаючи повідомлення про помилку.

В процесі тестування було визначено що програма працює правильно.

**4.3.Тестування "білої скриньки"**

 При тестуванні "білої скриньки" (англ. white-box testing, також говорять - прозорого ящика), тести створюються на основі знань про структуру системи і про те, як вона працює і критерії повноти тестування на відсотках елементів коду, які відпрацювали в ході виконання тестів.

Для проведення даного тестування було обрано завдання на курсову роботу

Критерії тестування:

Наявність змістовних ”шапок” до файлів

1. Наявність змістовних ”шапок” до функцій
2. Відповідність імен функції Венгерській нотації
3. Відповідність імен змінних Венгерській нотації
4. Наявність пояснюючих коментарів до коду
5. Дотримання форматування тексту
6. Мобільність
7. Оптимальне функціонування окремих блоків програми.

В процесі виконання тестів було проаналізовано текст файлу Violin\_Form.cs

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

FILE..........: Violin\_Form.cs

AUTHOR........: Sergiy Popil

DESCRIPTION...: The form where you may make changes in group of violin

METHOD........: Guitar\_Form(),button1\_Click(), button2\_Click(),Guitar\_Form\_Load()

SWITCHES......: .NET - if defined, 32-bit version is

compiled, otherwise 16-bit edition is compiled.

COPYRIGHT.....: Copyright (c) 2010, KI-21.

HISTORY.......: DATE COMMENT

----------------------------------------------------------------

 \* 05-19-10 Created - Sergiy

 \* 05-20-10 Modified - Sergiy

 \* 05-22-10 Modified - Sergiy

\\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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;

namespace musical\_school

{

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 CLASS…......: Violin\_Form.

 DESCRIPTION…: Violin\_Form, show information about group and make some changes.

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 public partial class Violin\_Form : Form

 {

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: Violin\_Form

 DESCRIPTION...: initicialization object

 ATTRIBUTES....: Public

 ARGUMENTS.....: none

 RETURNS.......: none

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 public Violin\_Form()

 {

 InitializeComponent();

 }

 CPupil[] clas = new CPupil[35];

 CPupil pupil;

 int i = 0;

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: Violin\_Form\_Load

 DESCRIPTION...: Inicialization form

 ATTRIBUTES....: private

 ARGUMENTS.....:

 \* object sender

 \* EventArgs e

 RETURNS.......: void

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 private void Violin\_Form\_Load(object sender, EventArgs e)

 {

 }

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: button1\_Click

 DESCRIPTION...: Cheked if the data is entry and add pupils

 ATTRIBUTES....: private

 ARGUMENTS.....:

 \* object sender

 \* EventArgs e

 RETURNS.......: void

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 private void button1\_Click(object sender, EventArgs e)

 {

 pupil = new CPupil();

 pupil.szName = textBox1.Text;

 clas[i] = pupil;

 i++;

 if (textBox1.Text == "")

 {

 MessageBox.Show("Error!No data");

 }

 else

 {

 listBox1.Items.Add(" - " + pupil.szName);

 textBox1.Text = "";

 }

 }

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: button2\_Click

 DESCRIPTION...: Create button to delete pupil

 ATTRIBUTES....: private

 ARGUMENTS.....:

 \* object sender

 \* EventArgs e

 RETURNS.......: void

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 private void button2\_Click(object sender, EventArgs e)

 {

 pupil = new CPupil();

 pupil.szName = textBox1.Text;

 clas[i] = pupil;

 i++;

 listBox1.Items.Remove(listBox1.SelectedItem);

 }

 }

}

/\*\* (END OF FILE : Violin\_Form.cs)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Аналіз результату.

 1. Змістовні ”шапки” до файлів присутні, виконанні відповідно до вимог, в них містить інформація про назву файла, його призначення, автора і статистика введення змін.

2. Змістовні ”шапки” до функцій присутні і виконанні відповідно до вимог в них описується назва функції, її призначення.

3. Всі імена функцій і змінних програми оформлені відповідно до Венгерської нотації,

4. Коментарі в програмі присутні, вони використовуються для пояснення використання кожної змінної і для кращого розуміння роботи програми. Написані вони відповідно до вимог, на міжнародній мові, тобто англійській.

5. Текст програми відформатовано відповідно до вимог.

6. Програма є мобільною і функціонування окремих блоків є оптимальним.

**4.4.Перевірка на модель**

Метою даного виду тестування є перевірка на правильність виконання поставленої задачі за допомогою додаткової моделі, яка вже відлагоджена і працює правильно. Дане тестування має наступний структурний вигляд:



Рис. 14 – Схема процесу перевірки на модель.

Для проведення тестування в якості системи було використано задачу №1. Завданням якої є піднести число до квадрату.

В якості моделі для перевірки було використано програму Calculator, яка дозволяє обчислювати математичні вирази.

В процесі тестування було здійснено перевірку для 15 різних n, і в 15 з 15 випадків результати виконання і системи і моделі були однаковими, тобто правильність виконання програми становить 100 %.

Нище наведено приклад виконання перевірки для n=4:



Рис. 15 – Результат виконання програми - системи.



Рис. 16 – Результат виконання програми - моделі.

**4.5. Тестування на відмову**

 Для проведення тестування на відмову ми обрали нашу програму, яка розроблялася як модель функціонування музичної школи.

 Основними критеріями тестування було дослідження реакції системи на виконання поставлених користувачем задач (здебільшого це було введення різноманітних вхідних даних).

 Аналізуючи отриманні результати можна зробити наступний висновок, що хоча наш модуль в цілому є середньоспеціалізованим, проте працює коректно і безвідмовно.

**Висновок**

 Під час виконання даного курсового проекту було досягнуто основну його мету, тобто було розроблено модель функціонування системи музичної школи.

 При проектуванні архітектури системи було використано мову UML. Перевага її в тому що за допомогою діаграм можна представити практично все що потрібно – при цьому вони є досить зрозумілими для рядового програміста або менеджера проекту. Хоча також є певні недоліки – нема жорсткої стандартизації при розробці певної системи. Це може призвести до того що різними особами окремі моменти на схемі можуть інтерпретуватися по різному. Також відсутність чіткої стандартизації ускладнює вивчення. Але попри все це використання даної мови все ж полегшує процес розробки.

 При розробці системи головний акцент робився на розклад навчання, редагування груп та отримання інформації про викладача.

 В процесі написання другої частини курсового – проведення тестування програмного забезпечення було досліджено модуль на відповідність поставленим вимогах. Для цього було розроблено ряд тестів різноманітного характеру. І було визначено, що дана система, яка була розроблена під час виконання курсової роботи не є досконалою в використанні, але є дуже спеціалізована і прив’язана до вирішення конкретно поставленого завдання. Додатково вона повністю задовольняє вимоги навчального процесу, адже вона написана відповідно до вимог Венгерській нотації і виконує поставленні завдання.

**Список літератури**

1. Лаврищева Е.М. , Петрухин В.А. “Методи и средства инженерии

 программного обеспечения”

1. [http://www.uk.wikipedia.org](http://www.uk.wikipedia.org/)
2. Мартін Фаулер, Кендалл Скот. “UML-основи. II видання”.
3. Дудзяний І.М., «Обєктно-орієнтоване моделювання програмних систем»- Львів, Видавничий центр ЛНУ ім. Івана Франка, 2007.
4. Андон Ф.И. , Коваль Г.И. ,Коротун Т.М, Сус­лов В.Ю. «*Основы* инженерии качества программных систем» . – Киев: Академпериодика, 2002. – 504 с.

**Додаток А**

**CPupil.cs**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

FILE..........: CPupil.cs

AUTHOR........: Sergiy Popil

DESCRIPTION...: class Pupil

SWITCHES......: .NET - if defined, 32-bit version is

compiled, otherwise 16-bit edition is compiled.

COPYRIGHT.....: Copyright (c) 2010, KI-21.

HISTORY.......: DATE COMMENT

----------------------------------------------------------------

 \* 05-20-10 Created - Sergiy

 \* 05-22-10 Modified - Sergiy

\\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace musical\_school

{

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 CLASS…......: CPupil

 DESCRIPTION…: Some information about pupils.

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 public class CPupil

 {

 public string szName;

 public string szSerName;

 public int iYear\_born;

 }

}

/\*\* (END OF FILE : CPupil.cs)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**CSchool.cs**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

FILE..........: CMusical\_School.cs

AUTHOR........: Sergiy Popil

DESCRIPTION...: class Musical\_School

SWITCHES......: .NET - if defined, 32-bit version is

compiled, otherwise 16-bit edition is compiled.

COPYRIGHT.....: Copyright (c) 2010, KI-21.

HISTORY.......: DATE COMMENT

----------------------------------------------------------------

 \* 05-20-10 Created - Sergiy

 \* 05-22-10 Modified - Sergiy

\\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace musical\_school

{

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 CLASS…......: CMusical\_School

 DESCRIPTION…: Give some information about school.

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 public class CMusical\_School

 {

 public string szName;

 public string szAddress;

 public int iYear\_create;

 public int iAcredetion;

 public string []szSpecialization;

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: CMusical\_School

 DESCRIPTION...: Inicialization data

 ATTRIBUTES....: public

 ARGUMENTS.....:

 RETURNS.......: void

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 public CMusical\_School()

 {

 szName = "Музична школа №1 ім.Чайковського";

 szAddress = "Львів,Сахарова 15";

 iYear\_create = 1995;

 iAcredetion = 3;

 szSpecialization[0] = "Фортепіано";

 szSpecialization[1] = "Гітара";

 szSpecialization[2] = "Скрипка";

 }

 }

}

/\*\* (END OF FILE : CSchool.cs)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**CTeacher.cs**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

FILE..........: CTeacher.cs

AUTHOR........: Sergiy Popil

DESCRIPTION...: class Teacher

SWITCHES......: .NET - if defined, 32-bit version is

compiled, otherwise 16-bit edition is compiled.

COPYRIGHT.....: Copyright (c) 2010, KI-21.

HISTORY.......: DATE COMMENT

----------------------------------------------------------------

 \* 05-20-10 Created - Sergiy

 \* 05-22-10 Modified - Sergiy

\\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace musical\_school

{

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 CLASS…......: CTeacher

 DESCRIPTION…: Some information about teacher.

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 public class CTeacher

 {

 public string szName;

 public int iYear\_born;

 public string szSpecialization;

 public string szContact\_Inform;

 }

}

/\*\* (END OF FILE : CTeacher.cs)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**Teacher.cs**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

FILE..........: CTeachers\_group.cs

AUTHOR........: Sergiy Popil

DESCRIPTION...: class Teachers\_group

SWITCHES......: .NET - if defined, 32-bit version is

compiled, otherwise 16-bit edition is compiled.

COPYRIGHT.....: Copyright (c) 2010, KI-21.

HISTORY.......: DATE COMMENT

----------------------------------------------------------------

 \* 05-20-10 Created - Sergiy

 \* 05-22-10 Modified - Sergiy

\\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace musical\_school

{

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 CLASS…......: CTeachers\_group

 DESCRIPTION…: Set some information about teachers.

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 public class CTeachers\_group

 {

 public CTeacher[] teach = new CTeacher[10];

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: CTeachers\_group

 DESCRIPTION...: Inicialization data

 ATTRIBUTES....: public

 ARGUMENTS.....:

 RETURNS.......: void

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 public CTeachers\_group()

 {

 teach[0] = new CTeacher();

 teach[0].szName = "Гамків П.А.";

 teach[0].iYear\_born = 1970;

 teach[0].szSpecialization = "Піаніно";

 teach[0].szContact\_Inform = "0678040010";

 teach[1] = new CTeacher();

 teach[1].szName = "Дружинець О.І.";

 teach[1].iYear\_born = 1973;

 teach[1].szSpecialization = "Гітара";

 teach[1].szContact\_Inform = "0936740219";

 teach[2] = new CTeacher();

 teach[2].szName = "Коберник Т.Б.";

 teach[2].iYear\_born = 1972;

 teach[2].szSpecialization = "Скрипка";

 teach[2].szContact\_Inform = "0987540210";

 teach[3] = new CTeacher();

 teach[3].szName = "Пилат О.Б.";

 teach[3].iYear\_born = 1968;

 teach[3].szSpecialization = "Піаніно";

 teach[3].szContact\_Inform = "096836767";

 teach[4] = new CTeacher();

 teach[4].szName = "Сікан Б.В.";

 teach[4].iYear\_born = 1975;

 teach[4].szSpecialization = "Гітара";

 teach[4].szContact\_Inform = "0685934210";

 teach[5] = new CTeacher();

 teach[5].szName = "Ціник І.Г.";

 teach[5].iYear\_born = 1971;

 teach[5].szSpecialization = "Скрипка";

 teach[5].szContact\_Inform = "0990014512";

 }

 }

}

/\*\* (END OF FILE : Teacher.cs)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**Main\_Form.cs**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

FILE..........: Main\_Form.cs

AUTHOR........: Sergiy Popil

DESCRIPTION...: Programm system from work with sructs of musical school

METHOD........: Main\_Form(),button1\_Click(),comboBox1\_SelectedIndexChanged(),Form1\_Load()

 \* comboBox1\_SelectedValueChanged(), button2\_Click()

SWITCHES......: .NET - if defined, 32-bit version is

compiled, otherwise 16-bit edition is compiled.

COPYRIGHT.....: Copyright (c) 2010, KI-21.

HISTORY.......: DATE COMMENT

----------------------------------------------------------------

 \* 05-19-10 Created - Sergiy

 \* 05-20-10 Modified - Sergiy

 \* 05-22-10 Modified - Sergiy

\\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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;

namespace musical\_school

{

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 CLASS…......: Main\_Form.

 DESCRIPTION…: Main\_Form, show some information about shcool.

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 public partial class Main\_Form : Form

 {

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: Main\_Form

 DESCRIPTION...: initicialization object

 ATTRIBUTES....: Public

 ARGUMENTS.....: none

 RETURNS.......: none

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 public Main\_Form()

 {

 InitializeComponent();

 }

 CPupil[] clas = new CPupil[35];

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: button1\_Click

 DESCRIPTION...: Create shedule

 ATTRIBUTES....: private

 ARGUMENTS.....:

 \* object sender

 \* EventArgs e

 RETURNS.......: void

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 private void button1\_Click(object sender, EventArgs e)

 {

 MessageBox.Show(" Monday " + " Tuesday " + " Wednesday " + " Thursday " + " Friday \n" +

 "--------------+--------------+--------------+---------------+--------------\n" +

 "Piano 13.00 |" + " Violin 14.30 |" + " Guitar 15.00 |" + " Piano 13.45 |" + " Guitar 14.45 \n" +

 " aud.12 |" + " aud.15 |" + " aud.14 |" + " aud.12 |" + " aud.14 \n" +

 "--------------+--------------+---------------+---------------+-------------\n" +

 "Guitar 14.30 |" + " Piano 16.00 |" + " Piano 16.30 |" + " Guitar 15.00 |" + " Guitar 14.45 \n" +

 " aud.14 |" + " aud.12 |" + " aud.14 |" + " aud.12 |" + " aud.14 \n" +

 "--------------+-------------+----------------+---------------+-------------\n" +

 "Violine 16.00 |" + " Guitar 17.30 |" + " Violine 18.00 |" + " Violine 16.30 |" + " Guitar 14.45 \n" +

 " aud.15 |" + " aud.14 |" + " aud.14 |" + " aud.12 |" + " aud.14 \n" +

 "-------------+--------------+----------------+--------------+--------------\n");

 }

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: comboBox1\_SelectedIndexChanged

 DESCRIPTION...: Create a list of groups

 ATTRIBUTES....: private

 ARGUMENTS.....:

 \* object sender

 \* EventArgs e

 RETURNS.......: void

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 private void comboBox1\_SelectIndexChanged(object sender,EventArgs e)

 {

 comboBox1.DropDownStyle = ComboBoxStyle.DropDownList;

 }

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: Form1\_Load

 DESCRIPTION...: Make some interface of combo box

 ATTRIBUTES....: private

 ARGUMENTS.....:

 \* object sender

 \* EventArgs e

 RETURNS.......: void

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 private void Form1\_Load(object sender, EventArgs e)

 {

 comboBox1.Text = "Виберіть групу...";

 }

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: comboBox1\_SelectedValueChanged

 DESCRIPTION...: Selected group and creates for there new forms

 ATTRIBUTES....: private

 ARGUMENTS.....:

 \* object sender

 \* EventArgs e

 RETURNS.......: void

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 private void comboBox1\_SelectValueChanged(object sender,EventArgs e)

 {

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 \* Select musical\_group from list

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 if (comboBox1.Text == "Фортепіано")

 {

 Piano\_Form pianino = new Piano\_Form();

 pianino.Show();

 }

 if (comboBox1.Text == "Гітара")

 {

 Guitar\_Form gitara = new Guitar\_Form();

 gitara.Show();

 }

 if (comboBox1.Text == "Скрипка")

 {

 Violin\_Form skrypka = new Violin\_Form();

 skrypka.Show();

 }

 }

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: button2\_Click

 DESCRIPTION...: Create new form for teachers

 ATTRIBUTES....: private

 ARGUMENTS.....:

 \* object sender

 \* EventArgs e

 RETURNS.......: void

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 private void button2\_Click(object sender, EventArgs e)

 {

 Teacher\_Form teacher = new Teacher\_Form();

 teacher.Show();

 }

 }

 }

/\*\* (END OF FILE : Main\_Form.cs)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**Guitar\_Form.cs**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

FILE..........: Guitar\_Form.cs

AUTHOR........: Sergiy Popil

DESCRIPTION...: The form where you may make changes in group of guitar

METHOD........: Guitar\_Form(),button1\_Click(), button2\_Click(),Guitar\_Form\_Load()

SWITCHES......: .NET - if defined, 32-bit version is

compiled, otherwise 16-bit edition is compiled.

COPYRIGHT.....: Copyright (c) 2010, KI-21.

HISTORY.......: DATE COMMENT

----------------------------------------------------------------

 \* 05-19-10 Created - Sergiy

 \* 05-20-10 Modified - Sergiy

 \* 05-22-10 Modified - Sergiy

\\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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;

namespace musical\_school

{

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 CLASS…......: Guitar\_Form.

 DESCRIPTION…: Guitar\_Form, show information about group and make some changes.

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 public partial class Guitar\_Form : Form

 {

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: Guitar\_Form

 DESCRIPTION...: initicialization object

 ATTRIBUTES....: Public

 ARGUMENTS.....: none

 RETURNS.......: none

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 public Guitar\_Form()

 {

 InitializeComponent();

 }

 CPupil[] clas = new CPupil[35];

 CPupil pupil;

 int i = 0;

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: button1\_Click

 DESCRIPTION...: Cheked if the data is entry and add pupils

 ATTRIBUTES....: private

 ARGUMENTS.....:

 \* object sender

 \* EventArgs e

 RETURNS.......: void

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 private void button1\_Click(object sender, EventArgs e)

 {

 pupil = new CPupil();

 pupil.szName = textBox1.Text;

 clas[i] = pupil;

 if (textBox1.Text == "")

 {

 MessageBox.Show("Error!No data");

 }

 else

 {

 listBox1.Items.Add(" - " + pupil.szName);

 textBox1.Text = "";

 }

 }

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: button2\_Click

 DESCRIPTION...: Create button to delete pupil

 ATTRIBUTES....: private

 ARGUMENTS.....:

 \* object sender

 \* EventArgs e

 RETURNS.......: void

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 private void button2\_Click(object sender, EventArgs e)

 {

 pupil = new CPupil();

 pupil.szName = textBox1.Text;

 clas[i] = pupil;

 i++;

 listBox1.Items.Remove(listBox1.SelectedItem);

 }

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: Guitar\_Form\_Load

 DESCRIPTION...: Inicialization form

 ATTRIBUTES....: private

 ARGUMENTS.....:

 \* object sender

 \* EventArgs e

 RETURNS.......: void

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 private void Guitar\_Form\_Load(object sender, EventArgs e)

 {

 }

 }

}

/\*\* (END OF FILE : Guitar\_Form.cs)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**Piano\_Form.cs**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

FILE..........: Piano\_Form.cs

AUTHOR........: Sergiy Popil

DESCRIPTION...: The form where you may make changes in group of piano

METHOD........: Piano\_Form(),button1\_Click(),listBox1\_SelectedIndexChanged(),textBox1\_TextChanged()

 \* button2\_Click(), Piano\_Form\_Load()

SWITCHES......: .NET - if defined, 32-bit version is

compiled, otherwise 16-bit edition is compiled.

COPYRIGHT.....: Copyright (c) 2010, KI-21.

HISTORY.......: DATE COMMENT

----------------------------------------------------------------

 \* 05-19-10 Created - Sergiy

 \* 05-20-10 Modified - Sergiy

 \* 05-22-10 Modified - Sergiy

\\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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;

namespace musical\_school

{

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 CLASS…......: Piano\_Form.

 DESCRIPTION…: Piano\_Form, show information about group and make some changes.

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 public partial class Piano\_Form : Form

 {

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: Piano\_Form

 DESCRIPTION...: initicialization object

 ATTRIBUTES....: Public

 ARGUMENTS.....: none

 RETURNS.......: none

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 public Piano\_Form()

 {

 InitializeComponent();

 }

 CPupil[] clas = new CPupil[35];

 CPupil pupil;

 int i = 0;

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: button1\_Click

 DESCRIPTION...: Cheked if the data is entry and add pupils

 ATTRIBUTES....: private

 ARGUMENTS.....:

 \* object sender

 \* EventArgs e

 RETURNS.......: void

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 private void button1\_Click(object sender, EventArgs e)

 {

 pupil = new CPupil();

 pupil.szName = textBox1.Text;

 clas[i] = pupil;

 i++;

 if (textBox1.Text == "")

 {

 MessageBox.Show("Error!No data");

 }

 else

 {

 listBox1.Items.Add(" - " + pupil.szName);

 textBox1.Text = "";

 }

 }

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: listBox1\_SelectedIndexChanged

 DESCRIPTION...: inicialization list box for pupils of group

 ATTRIBUTES....: private

 ARGUMENTS.....:

 \* object sender

 \* EventArgs e

 RETURNS.......: void

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 private void listBox1\_SelectedIndexChanged(object sender, EventArgs e)

 {

 }

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: textBox1\_TextChanged

 DESCRIPTION...: inicialization textbox for add or delete pupil

 ATTRIBUTES....: private

 ARGUMENTS.....:

 \* object sender

 \* EventArgs e

 RETURNS.......: void

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 private void textBox1\_TextChanged(object sender, EventArgs e)

 {

 }

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: button2\_Click

 DESCRIPTION...: Create button to delete pupil

 ATTRIBUTES....: private

 ARGUMENTS.....:

 \* object sender

 \* EventArgs e

 RETURNS.......: void

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 private void button2\_Click(object sender, EventArgs e)

 {

 pupil = new CPupil();

 pupil.szName = textBox1.Text;

 clas[i] = pupil;

 i++;

 listBox1.Items.Remove(listBox1.SelectedItem);

 }

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: Piano\_Form\_Load

 DESCRIPTION...: Inicialization form

 ATTRIBUTES....: private

 ARGUMENTS.....:

 \* object sender

 \* EventArgs e

 RETURNS.......: void

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 private void Piano\_Form\_Load(object sender, EventArgs e)

 {

 }

 }

}

/\*\* (END OF FILE : Piano\_Form.cs)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**Teachers\_Form.cs**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

FILE..........: Teachers\_Form.cs

AUTHOR........: Sergiy Popil

DESCRIPTION...: The form where you may get information about teacher

METHOD........: Guitar\_Form(),button1\_Click(), button2\_Click(),Guitar\_Form\_Load()

SWITCHES......: .NET - if defined, 32-bit version is

compiled, otherwise 16-bit edition is compiled.

COPYRIGHT.....: Copyright (c) 2010, KI-21.

HISTORY.......: DATE COMMENT

----------------------------------------------------------------

 \* 05-19-10 Created - Sergiy

 \* 05-20-10 Modified - Sergiy

 \* 05-22-10 Modified - Sergiy

\\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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;

namespace musical\_school

{

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 CLASS…......: Teacher\_Form.

 DESCRIPTION…: Teacher\_Form, show information about teacher.

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 public partial class Teacher\_Form : Form

 {

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: Teacher\_Form

 DESCRIPTION...: initicialization object

 ATTRIBUTES....: Public

 ARGUMENTS.....: none

 RETURNS.......: none

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 public Teacher\_Form()

 {

 InitializeComponent();

 }

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: comboBox1\_SelectedIndexChanged

 DESCRIPTION...: Selected teacher and show information about he

 ATTRIBUTES....: private

 ARGUMENTS.....:

 \* object sender

 \* EventArgs e

 RETURNS.......: void

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

 {

 for (int i = 0; i < 6; i++)

 {

 if (comboBox1.SelectedItem == kolectuv.teach[i].szName)

 {

 textBox1.Text = kolectuv.teach[i].iYear\_born.ToString();

 textBox2.Text = kolectuv.teach[i].szSpecialization;

 textBox3.Text = kolectuv.teach[i].szContact\_Inform;

 }

 }

 }

 public CTeachers\_group kolectuv = new CTeachers\_group();

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: Teachers\_Form\_Load

 DESCRIPTION...: selected teacher

 ATTRIBUTES....: private

 ARGUMENTS.....:

 \* object sender

 \* EventArgs e

 RETURNS.......: void

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 private void Teachers\_Form\_Load(object sender, EventArgs e)

 {

 comboBox1.Text = "Виберіть викладача...";

 for (int i = 0; i < 6; i++)

 {

 comboBox1.Items.Add(kolectuv.teach[i].szName);

 }

 }

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: textBox1\_TextChanged

 DESCRIPTION...: inicialization

 ATTRIBUTES....: private

 ARGUMENTS.....:

 \* object sender

 \* EventArgs e

 RETURNS.......: void

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 private void textBox1\_TextChanged(object sender, EventArgs e)

 {

 }

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: textBox2\_TextChanged

 DESCRIPTION...: inicialization

 ATTRIBUTES....: private

 ARGUMENTS.....:

 \* object sender

 \* EventArgs e

 RETURNS.......: void

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 private void textBox2\_TextChanged(object sender, EventArgs e)

 {

 }

 }

}

/\*\* (END OF FILE : Teachers\_Form.cs)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**Violin\_Form.cs**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

FILE..........: Violin\_Form.cs

AUTHOR........: Sergiy Popil

DESCRIPTION...: The form where you may make changes in group of violin

METHOD........: Guitar\_Form(),button1\_Click(), button2\_Click(),Guitar\_Form\_Load()

SWITCHES......: .NET - if defined, 32-bit version is

compiled, otherwise 16-bit edition is compiled.

COPYRIGHT.....: Copyright (c) 2010, KI-21.

HISTORY.......: DATE COMMENT

----------------------------------------------------------------

 \* 05-19-10 Created - Sergiy

 \* 05-20-10 Modified - Sergiy

 \* 05-22-10 Modified - Sergiy

\\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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;

namespace musical\_school

{

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 CLASS…......: Violin\_Form.

 DESCRIPTION…: Violin\_Form, show information about group and make some changes.

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 public partial class Violin\_Form : Form

 {

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: Violin\_Form

 DESCRIPTION...: initicialization object

 ATTRIBUTES....: Public

 ARGUMENTS.....: none

 RETURNS.......: none

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 public Violin\_Form()

 {

 InitializeComponent();

 }

 CPupil[] clas = new CPupil[35];

 CPupil pupil;

 int i = 0;

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: Violin\_Form\_Load

 DESCRIPTION...: Inicialization form

 ATTRIBUTES....: private

 ARGUMENTS.....:

 \* object sender

 \* EventArgs e

 RETURNS.......: void

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 private void Violin\_Form\_Load(object sender, EventArgs e)

 {

 }

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: button1\_Click

 DESCRIPTION...: Cheked if the data is entry and add pupils

 ATTRIBUTES....: private

 ARGUMENTS.....:

 \* object sender

 \* EventArgs e

 RETURNS.......: void

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 private void button1\_Click(object sender, EventArgs e)

 {

 pupil = new CPupil();

 pupil.szName = textBox1.Text;

 clas[i] = pupil;

 i++;

 if (textBox1.Text == "")

 {

 MessageBox.Show("Error!No data");

 }

 else

 {

 listBox1.Items.Add(" - " + pupil.szName);

 textBox1.Text = "";

 }

 }

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

 METHOD........: button2\_Click

 DESCRIPTION...: Create button to delete pupil

 ATTRIBUTES....: private

 ARGUMENTS.....:

 \* object sender

 \* EventArgs e

 RETURNS.......: void

 \\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 private void button2\_Click(object sender, EventArgs e)

 {

 pupil = new CPupil();

 pupil.szName = textBox1.Text;

 clas[i] = pupil;

 i++;

 listBox1.Items.Remove(listBox1.SelectedItem);

 }

 }

}

/\*\* (END OF FILE : Violin\_Form.cs)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/