**ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ**

**«ОПТИКО-МЕХАНІЧНИЙ ФАХОВИЙ КОЛЕДЖ КИЇВСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ ІМЕНІ ТАРАСА ШЕВЧЕНКА»**

Циклова комісія програмування та інформаційних технологій

**КУРСОВА РОБОТА**

**з дисципліни «Інструментальні засобі візуального програмування»**

на тему: «Проектування прикладного програмного забезпечення «Крилаті

вислови»»

|  |  |  |
| --- | --- | --- |
| ЗАХИЩЕНО З ОЦІНКОЮ  Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_  Кількість балів \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Керівник  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Таран В.В.  (підпис) |  | ВИКОНАВ |
| студент 3-го курсу гр. ІПЗ-31 спеціальність 121 «Інженерія програмного забезпечення» |
| Козак Іван Федорович  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис виконавця) |

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Таран В.В.

(підпис)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Заболотний В.О.

(підпис)

м. Київ — 2022

**ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ**

**«ОПТИКО-МЕХАНІЧНИЙ ФАХОВИЙ КОЛЕДЖ КИЇВСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ ІМЕНІ ТАРАСА ШЕВЧЕНКА»**

ЗАТВЕРДЖУЮ

В.о. директора

\_\_\_\_\_\_\_\_\_\_\_ Гаприндашвілі Б.В.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2022р.

|  |  |  |
| --- | --- | --- |
| **Дисципліна:** | Інструментальні засоби візуального програмування | |
| **Спеціальність:** | 121 «Інженерія програмного забезпечення» | |
| **Курс ІІІ** | **Група IПЗ-31** | **Семестр** **6** |

**ЗАВДАННЯ  
на курсову роботу студента**

|  |  |  |
| --- | --- | --- |
| Козака Івана Федоровича | | |
| **1.Тема курсової роботи:**  Проектування прикладного програмного забезпечення «Крилаті вислови» | | |
| Затверджена протоколом циклової комісії №06 від 25 січня 2022 р. | | |
| **2.Технічні вимоги:** ЄСПД, ЄСКД, ДСТУ 3008-95 | | |
| **3.Термін здачі студентом закінченої роботи: «**\_\_» червня 2022 р. | | |
| **4.Вихідні дані до виконання роботи:** | | |
| * Проаналізувати предметну область, виконати опис проектованої системи | | |
| і правил її функціонування; | | |
| * Розробити графічний інтерфейс користувача для роботи з його | | |
| елементами: головне вікно програми, система меню форм, довідкова система та інформація про програму; | | |
| **5.Зміст пояснювальної записки** (перелік питань, які підлягають розробці): | | |
| вступ; постановка задачі; характеристика апаратно-програмної частини; | | |
| опис мови і середовища програмування; розробка алгоритму та його опис; | | |
| створення та тестування програми; висновки; перелік умовних позначень; | | |
| список літератури; програмний лістинг. | | |
| **6. Дата видачі завдання:** «05» квітня 2022 р. | | |
| Керівник роботи |  | Таран Вікторія Валеріївна | |
| Завдання прийняв до виконання |  | Козак Іван Федорович | |

«05» квітня 2022 р.

**КАЛЕНДАРНИЙ ПЛАН-ГРАФІК**

**виконання курсової роботи з дисципліни**

**«Інструментальні засоби візуального програмування»**

|  |  |  |
| --- | --- | --- |
| **№ з/п** | **Етапи** | **Термін виконання** |
|  | Обрання та затвердження теми, попередня постановка задачі | 01.04-05.04.2022 |
|  | Підбір літератури, веб-джерел, ознайомлення зі стандартами | 06.04-12.04.2022 |
|  | Вивчення та аналіз предметної області об'єкту курсової роботи | 13.04-19.04.2022 |
|  | Підготовка оглядової частини | 20.04-26.04.2022 |
|  | Розробка архітектури та загальної структури системи | 27.04-03.05.2022 |
|  | Програмна реалізація системи | 04.05-10.05.2022 |
|  | Оформлення роботи, передача її на рецензування керівникові | 11.05-17.05.2022 |
|  | Рецензування роботи, уточнення назви, висновків, підготовка презентаційних матеріалів | 18.05-24.05.2022 |
|  | Попередній захист курсової роботи | 25.05-07.06.2022 |
|  | Захист курсової роботи | 8.06.2022 |

|  |  |  |
| --- | --- | --- |
| Керівник роботи |  | Таран Вікторія Валеріївна |
| Студент |  | Козак Іван Федорович |
|  |  | «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2022 р. |

**Зміст**

[**Реферат** 5](#_Toc11986430)

[**Скорочення** 6](#_Toc11986431)

**Вступ**  [7](#_Toc11986432)

[**Розділ 1.Опис напрямків дослідження** 9](#_Toc11986433)

[**Розділ 2. Проектування структури додатку, діаграма класів** 10](#_Toc11986436)

**Розділ 3. Проектування інтерфейсу додатку**11

[**Тестування програми**  15](#_Toc11986440)

[**Висновок** 17](#_Toc11986442)

[**Список використаної літератури** 20](#_Toc11986441)

[**Додатки** 21](#_Toc11986443)

# **Реферат**

Тема курсової роботи: «Проектування прикладного програмного забезпечення «Крилаті вислови»». Робота складається з 22 сторінок.

# Курсова робота складається із вступу, чотирьох розділів, висновків, списку використаних джерел (10 найменувань) та одного додатку. Робота містить дванадцять ілюстрацій. Загальний обсяг роботи становить 22 сторінок, основний текст викладено на 13 сторінках.

# Ключові слова: ПРОГРАМА, ДАНІ, ВИСЛІВ, КЛАС , АЛГОРИТМ, ІНТЕРФЕЙС, ДОДАТОК .

# Об’єктом дослідження є розробка ПЗ «Крилаті вислови» на мові програмування С#.

# Мета роботи – розробити ПЗ «Крилаті вислови»

Методика (метод дослідження): комп’ютерна реалізація та комп’ютерне тестування.

Результати досліджень можна використовувати як застосунок для перегляду крилатих висловів

# **Скорочення**

ПЗ – програмне забезпечення

ООП – об’єктно-орієнтовне програмування

# **ВСТУП**

В даній курсовій роботі розглядається розробка прикладного програмного забезпечення на прикладі розробки додатку «Довідник крилатих фраз» на мові програмування Microsoft Visual Studio С#.

Стоїть задача, використовуючи принципи об’єктно-орієнтованого програмування (ООП), створити базовий клас і на основі базового класу будувати похідні класи.

Клас – це визначений користувачем тип, який складається з полів, методів та членів, які оперують цими даними. Всі разом поля даних класу представляють стан екземпляра класу (інакше називають його об’єктом).

Основні положення об'єктно-орієнтованого програмування: інкапсуляція, спадкування, поліморфізм.

Інкапсуляція – це механізм, який об’єднує дані і код, а також захищає дані і код від стороннього втручання. Інкапсуляція – приховування даних. Інкапсуляція означає, що зовнішнім об’єктам і користувачу прямий доступ до даних заборонено. Це робиться для підтримки цілісності даних і їх несуперечливості.

Поліморфізм – це властивість коду вести себе по-різному в залежності від ситуації, яка виникає в момент виконання. Поліморфізм – це не стільки характеристика об’єктів, скільки характеристика функцій-членів класу і проявляється, зокрема, у можливості використання одного імені функції-члена для функцій, які мають різні типи аргументів, якщо виконувані функцією дії визначаються типом переданих їй аргументів. Це називається перевантаження функції.

Спадкування – це процес, в результаті якого один об’єкт може набути властивості іншого. На основі уже існуючих класів можна будувати похідні класи. При цьому похідний клас (клас-нащадок) наслідує дані-елементи і функції-елементи від своїх батьківських класів (класів-предків), добавляючи до них свої, характерні тільки йому, властивості та методи.

Використання спадковості та поліморфізму у класах дозволяє уникнути повторюваності програмного коду та зручно впорядкувати складні виклики методів, що об’єднані між собою в ланцюг.

Клас визначає формат (опис) деяких даних та роботу (поведінку) над цими даними. З оголошення класу можна отримати різну кількість об’єктів класу (змінних типу “клас”). Кожен об’єкт класу визначається конкретним (на даний момент) значенням внутрішніх даних, який називають станом об’єкту.

У класі оголошуються дані (внутрішні змінні, властивості) та методи (функції), що оперують цими даними (виконують роботу над даними).

Клас може бути успадкований від інших класів верхнього рівня. Це означає, що клас може використовувати частину коду інших класів верхнього рівня.

Також клас може бути батьківським для інших класів, які успадковують його програмний код.

Масив– набір змінних однакового типу. Доступ до цих змінних здійснюється з допомогою одного й того ж імені. Це ім’я називається іменем масиву.

Масиви використовуються для групування зв’язаних змінних між собою.

# **Розділ 1. Опис напрямків дослідження**

## Розробити додаток прикладного програмного забезпечення „Довідник крилатих фраз“.

## При запуску програми буде виводитися вікно, яке буде містити параметри такі як:

## • Автори.

## • Тип.

## • Тема.

## • Джерела.

## • Кнопки.

## • Поле в якому виводитиметься текс.

## Користувачу дається можливість вибирати параметри або робити пошук по ключовому слову.

## В додатку містяться три кнопки, які відповідають за пошук, очистку і вихід з програми.

## При виборі афоризмів чи висловів блокується панель тем і джерел. Щоб розблокувати ці панелі натискаємо кнопку “Назад” або вибираємо Приказки чи Прислів’я.

## Щоб зробити пошук по слову потрібно написати слово і натиснуть „Пошук“.

## Додаток має виконуватись у Windows Form.

# 

# **Розділ 2. Проектування структури додатку, діаграма класів**

Структура нашого додатку складається із базового класу Vislovi та похідних класів Autor, Tipe, Themcs, Djerelo.

Зв’язок між базовим класом Phrases та похідними класами Autor, Tipe, Themcs, Djerlo можна спостерігати на діаграмі класів додатку «Крилаті фрази»

Так базовий (батьківський) клас Abstract містить поля: ukr, eng, knowledge, family, friend, language, work, proverb, sayings, aphorisms, vislovi і тд.

Похідні класи Autor, Tipe, Themcs, Djerelo успадковують від батьківського класу властивості, які автоматично реалізовані (Рис.1).

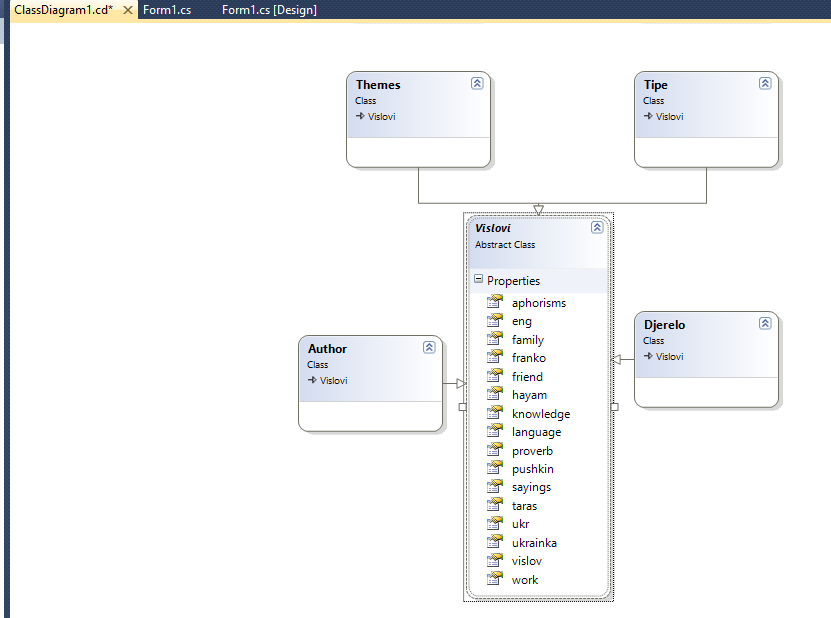
****

Рисунок 1 Діаграма класів

**Розділ 3. Проектування інтерфейсу додатку**

Початок роботи в Microsoft Visual Studio C# здійснюється запуском на виконання файлу devenv.exe ("C:\Program Files (x86)\Microsoft Visual Studio 10.0\Common7\IDE\devenv.exe").

Як правило, використовується для цього Головне меню: Пуск / Все программы / Microsoft Visual Studio 2010 / Microsoft Visual Studio 2010.

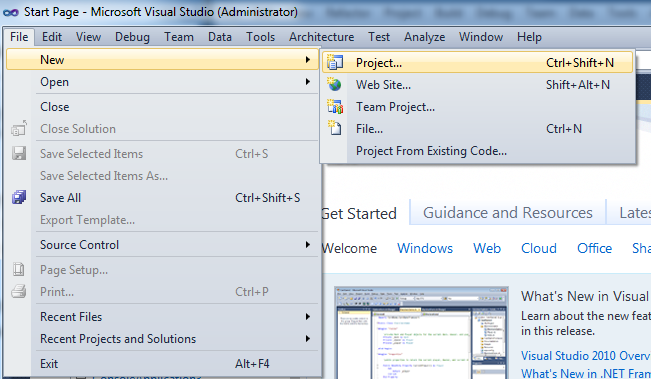
У головному меню вибираємо: File / New / Project… (Рис. 2).

Рисунок 2 Створення проекту

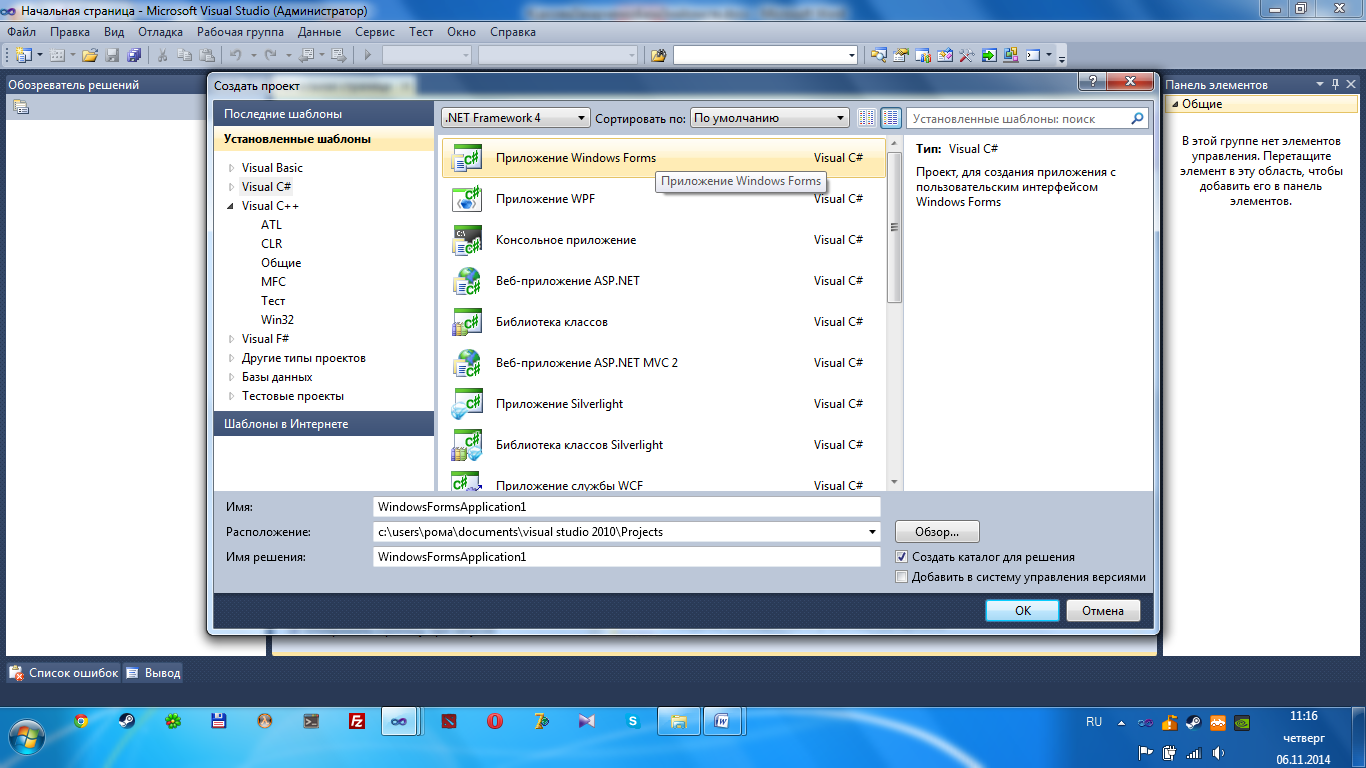
З’явиться діалогове вікно „New Project“, в якому слід вибрати мову Visual C#, тип додатку Windows Form Application та вказати назву проекту та його розміщення (Рис.2.1).

Рисунок 2.1 Вибір мови програмування у діалоговому вікні

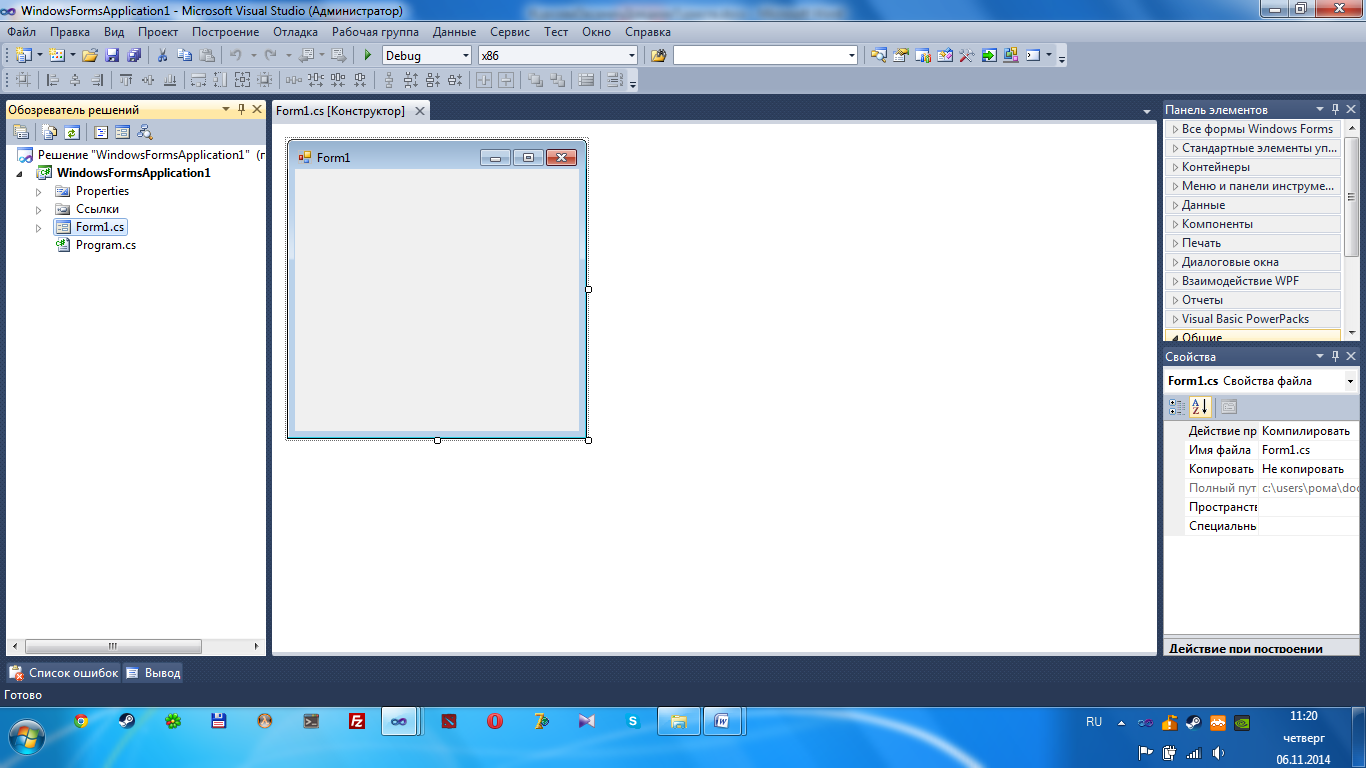
Коли ми все це зробили появиться таке вікно (Рис. 2.2)

Рисунок 2.2. Вигляд вікна у конструкторі

В курсовій роботі використовувалися такі елементи: button, richTextBox, groupBox, radioButton.

Щоб добавити якийсь елемент потрібно вибрати Вид/Панель элементов та вибрати певний елемент і перетягнути його на форму (Рис. 2.3)

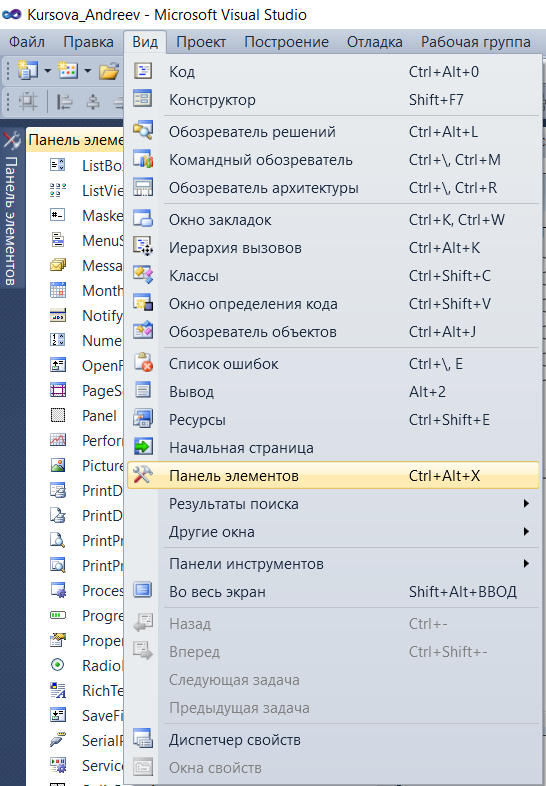


Рисунок 2.3 Додавання потрібного елементу

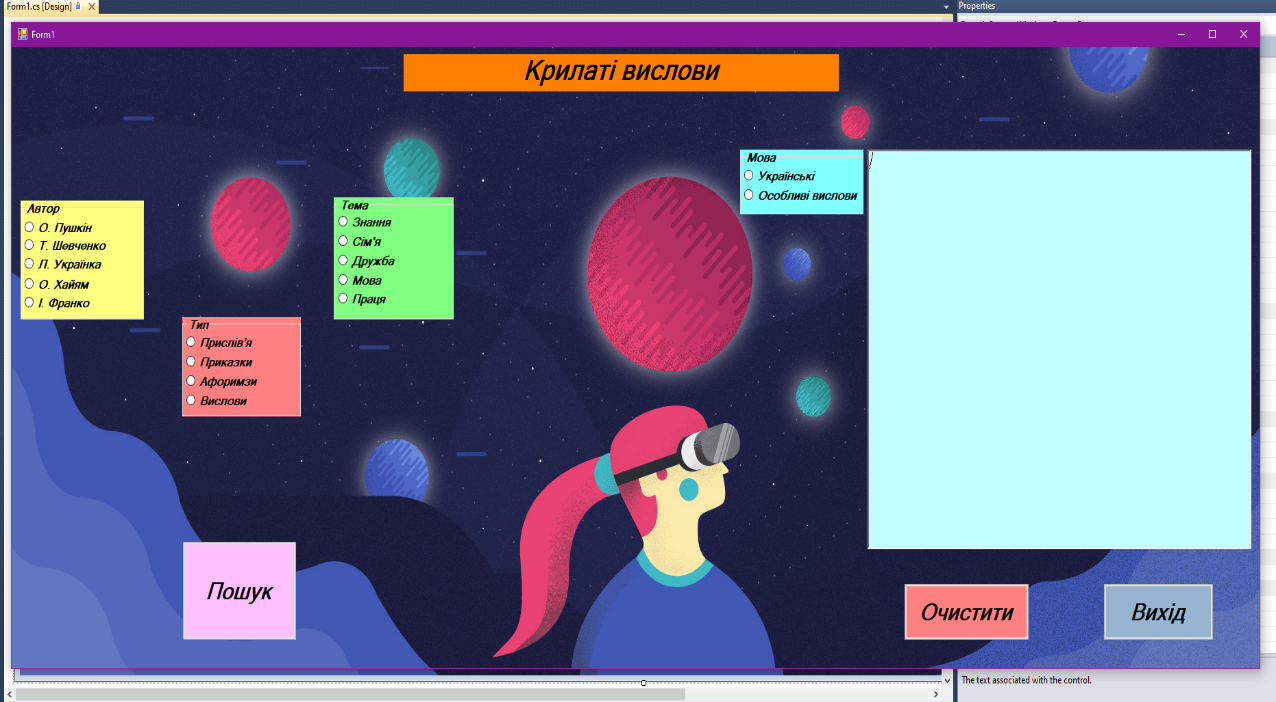
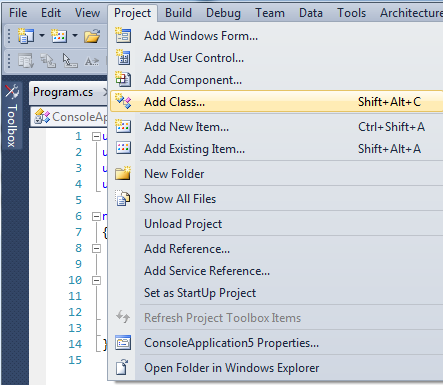
На моїй формі 4 groupBox, 16 radioButton, 3 button, , 1 richTextBox (Рис.2.4)

Рис.2.4 Як виглядає моя форма

Далі додаємо класи (Рис.2.5)

Рисунок 2.5 Додавання класів

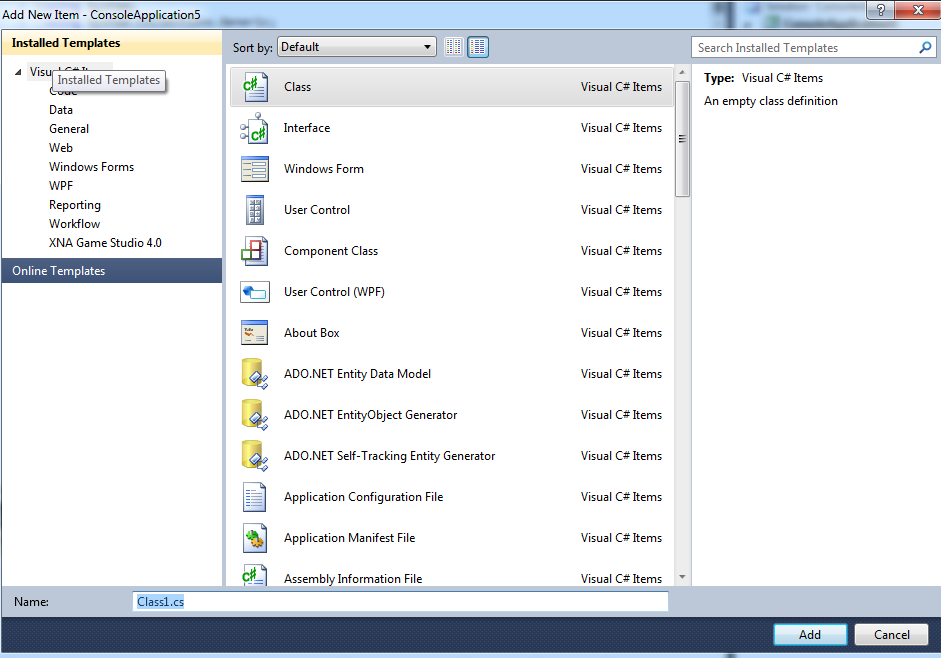
В діалоговому вікні задаємо назви класів (Рис.2.6)

Рисунок 2.6 Задаємо назви класів

**Тестування програми**

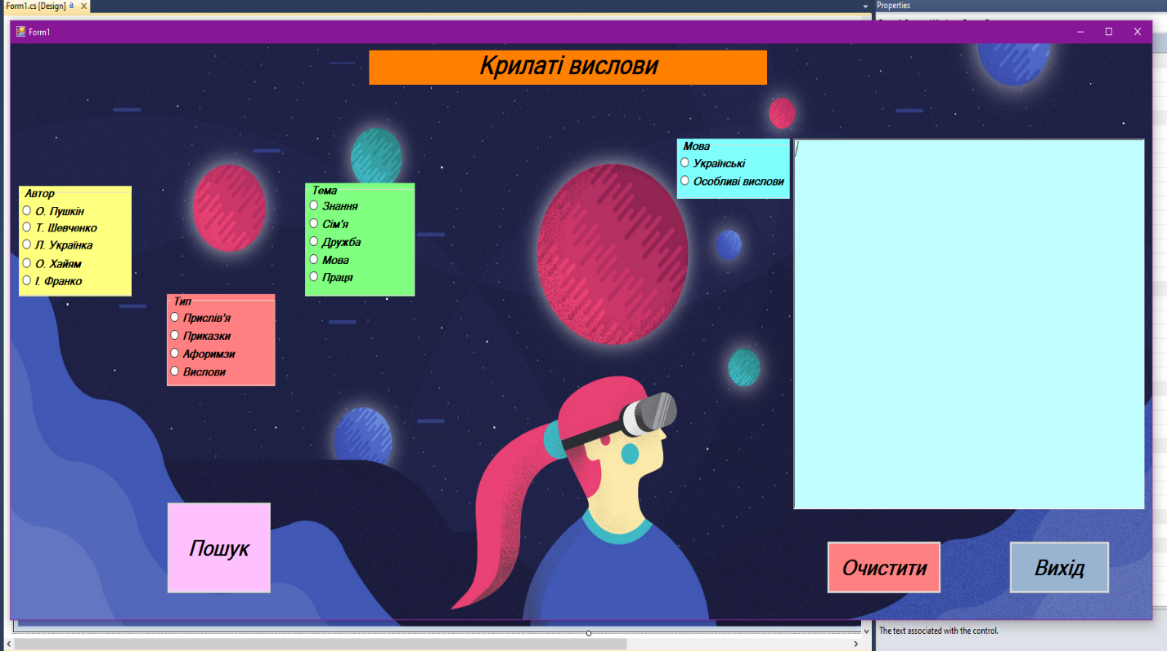
При запуску програми виводиться на екран таке вікно (Рис.3):

Рис.3 Виведення вікна при запуску програми

Рисунок 3 Виведення вікна при запуску програми

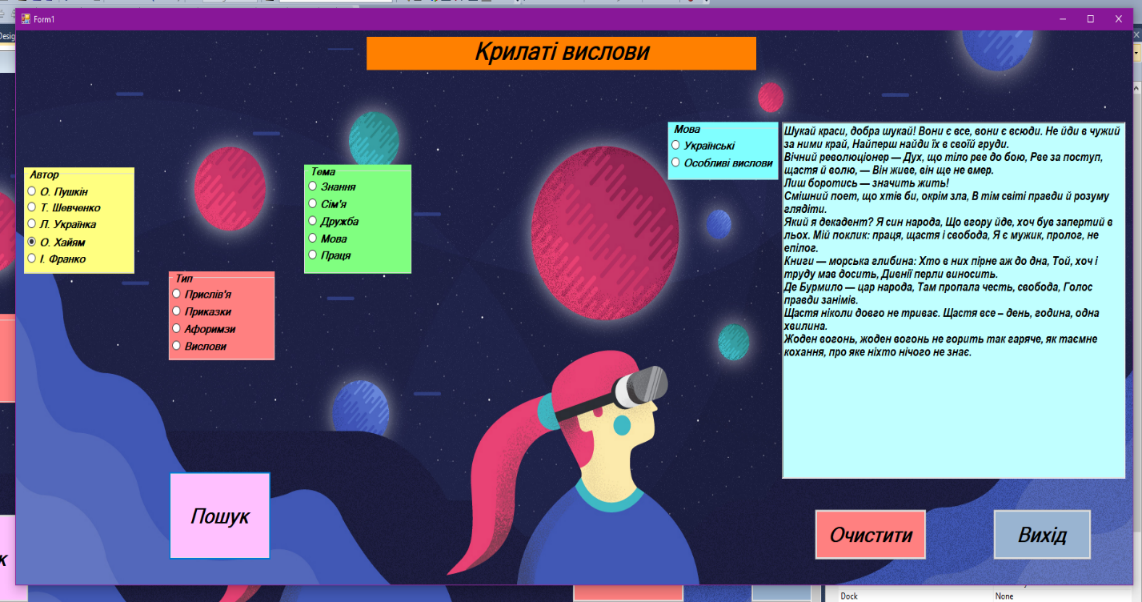
Якщо ми виберемо один або декілька параметрів і натиснемо кнопку „Пошук“, то виведе потрібний текст (Рис.3.1).

Рисунок 3.1 Функціонал кнопки «Пошук»

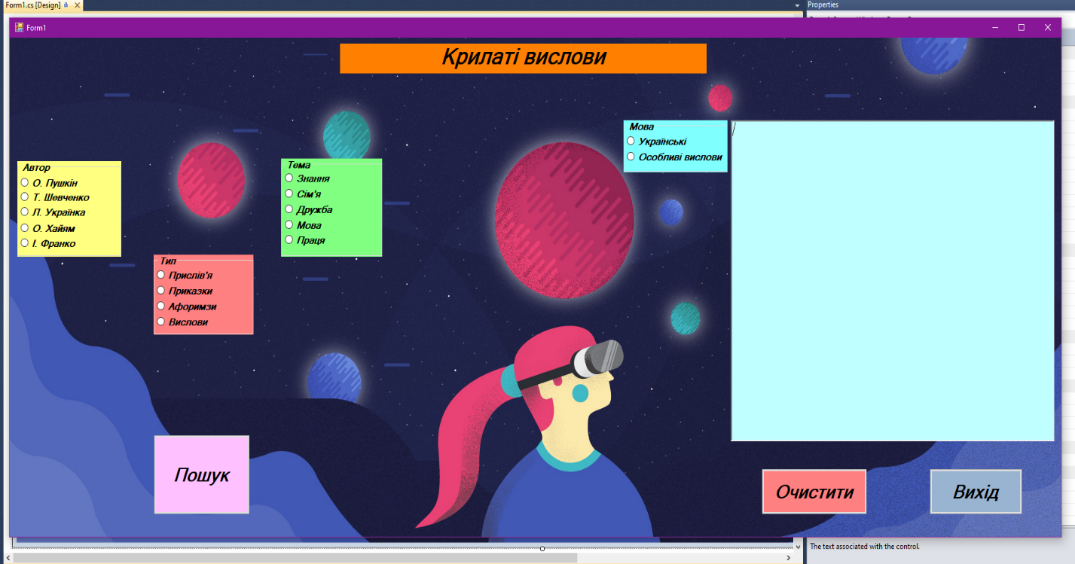
При натисканні кнопки „Очистити“, зникнуть ваші обрані параметри (Рис.3.2)

Рисунок 3.2 Функціонал кнопки «Очистити»

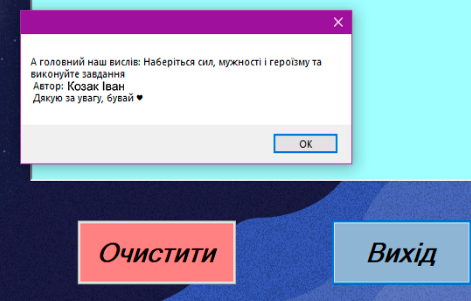
При натисканні кнопки „Вихід“ відбувається вихід з програми(Рис.3.3).

Рисунок 3.2 Функціонал кнопки «Вихід»

**Висновок**

У результаті виконання курсової роботи мною було опрацьовано ряд джерел інформації, одержані більш глибокі знання з мови програмування Microsoft Visual Studio C#. Створено прикладне програмне забезпечення “Крилаті вислови”, в якому застосовуються основні принципи об’єктно-орієнтованого програмування. Основні положення об'єктно-орієнтованого програмування: інкапсуляція, спадкування, поліморфізм.

Вдалось відокремити фундаментальні поняття (принципи), використані у переважній більшості визначень об'єктно-орієнтованого програмування. До них належить:

[***Клас***](https://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F))

Клас визначає абстрактні характеристики деякої сутності, включно з характеристиками самої сутності (її атрибутами або властивостями) та діями, які вона здатна виконувати (її поведінкою, [методами](https://uk.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) або можливостями). Властивості та методи класу, разом називаються його членами.

[***Об'єкт***](https://uk.wikipedia.org/wiki/%D0%9E%D0%B1%27%D1%94%D0%BA%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F))

Окремий екземпляр класу (створюється після запуску програми і ініціалізації полів класу).

[***Метод***](https://uk.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F))

Можливості об'єкта. В межах програми, використання методу має впливати лише на один об'єкт

[***Обмін повідомленнями***](https://uk.wikipedia.org/wiki/%D0%9E%D0%B1%D0%BC%D1%96%D0%BD_%D0%BF%D0%BE%D0%B2%D1%96%D0%B4%D0%BE%D0%BC%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F%D0%BC%D0%B8)

«Передача даних від одного процесу іншому, або надсилання викликів методів.»

[***Успадкування***](https://uk.wikipedia.org/wiki/%D0%A3%D1%81%D0%BF%D0%B0%D0%B4%D0%BA%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F))***(наслідування)***

Клас може мати «підкласи», спеціалізовані, розширені версії надкласу. Можуть навіть утворюватись цілі дерева успадкування. Так, наприклад, в Java дозволене лише одинарне успадкування, а в С і те і інше.

[***Приховування інформації***](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D1%85%D0%BE%D0%B2%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D1%96%D1%97)***(інкапсуляція)***

Приховування деталей про роботу класів від об'єктів, що їх використовують або надсилають їм повідомлення. Кожен об'єкт надає кожному іншому класу певний [інтерфейс](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) — члени, доступні іншим класам. Інкапсуляція потрібна для того, аби запобігти використанню користувачами інтерфейсу тих частин реалізації, які, швидше за все, будуть змінюватись. Це дасть змогу полегшити внесення змін без потреби змінювати і користувачів інтерфейсу. Деякі мови програмування йдуть ще далі: [Java](https://uk.wikipedia.org/wiki/Java_(%D0%BC%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) використовує ключове слово private для обмеження доступу, що буде дозволений лише з методів того самого класу, protected — лише з методів того самого класу і його нащадків та з класів із того ж самого пакету, [C#](https://uk.wikipedia.org/wiki/C_Sharp_(%D0%BC%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) та [VB.NET](https://uk.wikipedia.org/wiki/Visual_Basic_.NET) відкривають деякі члени лише для класів із тієї ж збірки шляхом використання ключового слова internal (C) або Friend (VB.NET), а [Eiffel](https://uk.wikipedia.org/wiki/Eiffel_(%D0%BC%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) дозволяє вказувати які класи мають доступ до будь-яких членів.

[***Абстрагування***](https://uk.wikipedia.org/wiki/%D0%90%D0%B1%D1%81%D1%82%D1%80%D0%B0%D0%B3%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F))

Спрощення складної дійсності шляхом моделювання класів, що відповідають проблемі, та використання найприйнятнішого рівня деталізації окремих аспектів проблеми.

***Поліморфізм***

Означає залежність поведінки від класу, в якому ця поведінка викликається, тобто, два або більше класів можуть реагувати по-різному на однакові повідомлення. На практиці - це реалізовується шляхом реалізації ряду підпрограм (функцій, процедур, методи тощо) з однаковими іменами, але з різними параметрами. Залежно від того, що передається, і вибирається відповідна підпрограма.

Я розробив додаток прикладного програмного забезпечення «Крилаті вислови», який дає можливість вибирати параметри які ви бажаєте. При запуску програми буде виводитися вікно, яке буде містити такі параметри : Автори; Тип; Тема; Джерела; Кнопки; Поле в якому виводитиметься текс. Коли ми виберемо один або декілька параметрів і натиснемо кнопку „Пошук“, то виведе задовільний текст. Додаток виконується у Windows Form.

# **С****писок використаної літератури**

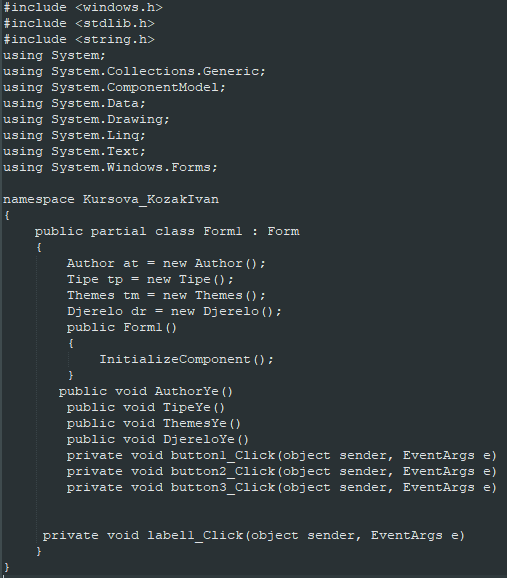
1. Visual Basic 2010 на прикладах (+ CD-ROM) Виктор Зиборов
2. П. Голд. Мова програмування C#. Класика Computers Science. 4-те видання 2014
3. Герберт Шілдт. C# 4.0: повне керівництво 2010
4. Крістіан Нейгел, Карлі Вотсон та ін. Visual C# 2010: повний курс

**Інтернет ресурси**

1. <https://ru.stackoverflow.com/questions/570092>
2. <https://msdn.microsoft.com/ru-ru/library/system.drawing.fontfamily.getlinespacing(v=vs.110).aspx>
3. <https://uk.wikiquote.org/wiki/%D0%A3%D0%BA%D1%80%D0%B0%D1%97%D0%BD%D1%81%D1%8C%D0%BA%D1%96_%D0%BD%D0%B0%D1%80%D0%BE%D0%B4%D0%BD%D1%96_%D0%BF%D1%80%D0%B8%D1%81%D0%BB%D1%96%D0%B2%27%D1%8F_%D1%82%D0%B0_%D0%BF%D1%80%D0%B8%D0%BA%D0%B0%D0%B7%D0%BA%D0%B8:_%D0%BF%D1%80%D0%B0%D1%86%D1%8F,_%D0%B3%D0%BE%D1%81%D0%BF%D0%BE%D0%B4%D0%B0%D1%80%D1%81%D1%82%D0%B2%D0%BE>
4. <http://traditions.in.ua/usna-narodna-tvorchist/pryslivia-ta-prykazky>
5. <http://www.zoda.gov.ua/news/24558/krilati-vislovi-y-aforizmi-z-poeziy-t.g.-shevchenka.html>
6. <https://sites.google.com/site/linakostenkopysarenko/krilati-vislovi>
7. <http://patrioty.org.ua/society/lini-kostenko-86-krashchi-tsytaty-pro-liubov-i-zhyttia-112357.html>
8. https://ru.wikipedia.org/wiki/C\_Sharp

# **Додаток**

# Код головної форми

**Опис алгоритму програми та розробка програмного коду**

**Похідні класи**

**Головний (похідний) клас - Vislovi**

**namespace Kursova\_KozakIvan**

**{**

**abstract class Vislovi**

**{**

**public String ukr { get; set; } //Українські**

**public String eng { get; set; } //особливі**

**//Тема**

**public String knowledge { get; set; } // Знання**

**public String family { get; set; } // Сім'я**

**public String friend { get; set; } // Друзі**

**public String language { get; set; } // Мова**

**public String work { get; set; } // Праця**

**//Тип**

**public String proverb { get; set; } // Прислів'я**

**public String sayings { get; set; } // Приказки**

**public String aphorisms { get; set; } // Афоризми**

**public String vislov { get; set; } // Вислови**

**//Автори**

**public String pushkin { get; set; } //О.Пушкін**

**public String taras { get; set; } //Т.Шевченко**

**public String ukrainka { get; set; } //Л.Українка**

**public String hayam { get; set; } //О. Хайям**

**public String franko { get; set; } //І. Франко**

**}**

**}**