МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

„КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. І. Сікорського”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**КУРСОВА РОБОТА**

з дисципліни «Об’єктно-орієнтоване програмування»   
на тему

**Шаблони проєктування в ООП. Комп’ютерна гра “Glory to Ukraine”**

Виконав студент

ІІ курсу групи КП-12   
Якубишин Анатолій Cергійович

Керівник роботи

доцент, к.т.н. Заболотня Т.М. Оцінка

(дата, підпис)

КИЇВ 2023

**ЗМІСТ**

[ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ 3](#_Toc135652687)

[ВСТУП 4](#_Toc135652688)

[1. ОПИС СТРУКТУРНО-ЛОГІЧНОЇ СХЕМИ ПРОГРАМИ 6](#_Toc135652689)

# **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ**

**ПЗ** – (Програмне забезпечення) сукупність програм і програмних документів, необхідних для експлуатації цих програм.

**С#** – Об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET.

**MVVM** – Model-View-ViewModel — [шаблон проєктування](https://uk.wikipedia.org/wiki/%D0%A8%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD%D0%B8_%D0%BF%D1%80%D0%BE%D1%94%D0%BA%D1%82%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%D0%BD%D0%BE%D0%B3%D0%BE_%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F), що застосовується під час проєктування архітектури [застосунків (додатків)](https://uk.wikipedia.org/wiki/%D0%97%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA" \o "Застосунок). Публічно вперше був представлений Джоном Госсманом (John Gossman) у 2005 році як модифікація шаблону Presentation Model. MVVM орієнтований на такі сучасні платформи розробки, як [Windows Presentation Foundation](https://uk.wikipedia.org/wiki/WPF) та [Silverlight](https://uk.wikipedia.org/wiki/Silverlight" \o "Silverlight) від компанії [Microsoft](https://uk.wikipedia.org/wiki/Microsoft).

**WPF** – **Windows Presentation Foundation** (WPF, кодова назва — **Avalon**) — графічна (презентаційна) підсистема (аналог WinForms), яка починаючи з [.NET Framework 3.0](https://uk.wikipedia.org/wiki/.NET_Framework_3.0) в складі цієї платформи. Має пряме відношення до [XAML](https://uk.wikipedia.org/wiki/XAML). WPF разом з .NET Framework 3.0 вбудована в [Windows Vista](https://uk.wikipedia.org/wiki/Windows_Vista), а також доступна для установки в [Windows XP](https://uk.wikipedia.org/wiki/Windows_XP) Service Pack 2 і [Windows Server 2003](https://uk.wikipedia.org/wiki/Windows_Server_2003).

**ППO – Протиповітряна оборона.**

# **ВСТУП**

Дана курсова робота присвячена розробці програмного забезпечення комп’ютерної гри “Glory to Ukraine” за допомогою шаблонів проєктування. Багатато людей і не тільки дітей люблять грати в комп’ютерні ігри. Окрім розважальної мети, розроблене ПЗ має привернути увагу людей до важкої роботи сил ППО України та дасть змогу приміряти на себе роль захисника у розважальному вигляді. Взагалі ігри розвивають нестандартне мислення, спритність, швидкість реакції. Дана тематика обрана для виконання курсової роботи тому, що результати абстрагування об’єктів у цій предметній галузі дозволяють застосувати вивчені принципи та методи об’єктно‑орієнтованого програмування для створення програмного забезпечення, зокрема шаблони проектування.

*Об’єктом* дослідження є процес гри в комп’ютерну гру “Glory to Ukraine”.

*Метою роботи* є розроблення програмного забезпечення комп’ютерної гри “Glory to Ukraine” з використанням шаблонів проєктування.

Для досягнення визначеної мети необхідно виконати такі *завдання*:

* абстрагувати об’єкти предметної галузі;
* розробити структурну організацію ПЗ за допомогою застосування основних принципів ООП та шаблонів проєктування;
* визначити та описати функціональні характеристики програми;
* обґрунтувати вибір шаблонів проєктування, використаних для побудови програми;
* розробити користувацький графічний інтерфейс;
* виконати реалізацію програмного забезпечення відповідно до вимог технічного завдання;
* виконати тестування розробленої програми;
* оформити документацію з курсової роботи.

Розроблене ПЗ cкладається з трьох основних типів модулів: моделі, види, та вид-моделі на базі шаблону MVVM.

Реалізовані шаблони проєктування: MVVM, Шаблоний метод, Стратегія, Фабричний метод, Команда, Легковаговик, Проксі, Спостерігач та міні шаблон – Одинак.

До функціональних можливостей програми належать: надання інформації щодо правил гри, зміна мови гри, надання характеристики щодо ракет ППО та ворожих літальних апаратів (літаки та дрони), запуск рівня та тренування.

Для функціювання розробленої програми (після встановлення з інсталятора, а не сам безпосередньо проект) потрібно 60 Мб вільного дискового простору.

Розроблене програмне забезпечення може бути використане людьми, що люблять грати в шутери та хочуть приємно провести час.

Пояснювальна записка складається зі вступу **ДОПИСАТИ!!!**

1. **ОПИС СТРУКТУРНО-ЛОГІЧНОЇ СХЕМИ ПРОГРАМИ**

**1.1. Модульна організація програми**

*Рис. 1.1.1. Модульна організація програми*

У програмі можна виділити декілька основних логічних модулів, що мають певну самостійність і обмінюються один з одним даними. Це модулі: Модель, Вид та Вид-Модель. Це є різновидом класичної багатошарової архітектури ПЗ, а саме його підвидом – MVVM.

**Моделі** – являють собою фундаментальні дані, що необхідні для роботи застосунку.

**Вигляд** — це графічний інтерфейс, тобто вікно, кнопки тощо. Cкладається з xaml розмітки та класу вікна у WPF.

**Вид-модель** з одного боку є абстракцією **Вигляду**, а з іншого надає обгортку даних з Моделі, які мають зв'язуватись. Тобто вона містить Модель, яка перетворена до Вигляду, а також містить у собі команди, якими може скористатися Вигляд для впливу на Модель. Фактично ViewModel призначена для того, щоб

* Здійснювати зв'язок між моделлю та вікном
* Відслідковувати зміни в даних, що зроблені користувачем
* Відпрацьовувати логіку роботи View (механізм команд)

Серед **Видів,**  з якими користувач може взаємодіяти, є головне вікно, інтсрукція, режим вибору рівня, вікно поля бою.

Для кожного **Виду** відповідно існує свій **Вид-Модель.**

## 1.2. Функціональні характеристики

Розроблене ПЗ дозволяє користувачу зручно керувати процесом гри за допомогою графічного інтерфейсу. Як тільки користувач заходить в гру , відкривається головне вікно, на якому можна змінити мову, при чому вибір користувача запам’ятається і при повторному запуску програми мова залишиться збереженою. Як тільки користувач встановлює гру у нього не пройдено жодного рівня і йому доступне тренування та перший рівень, також у користувача є можливість детально ознайомитися з графічної інструкцією. Під час проходження рівняння або тренування, користувач може натиснути на паузу (клавіша ESC) та з з’явленого вікна може зробити наступні дії: вийти з гри взагалі, вийти в меню, перезапустити рівень або продовжити гру. Також хочу зазначити, для того щоб пройти рівень потрібно знищити всі цілі, але при цьому якщо хоча б одна ціль була пропущена рівень не перезапускаєтьcя. Це було зроблено з метою, щоб людина мала змогу подивитися на порядок повітряних цілей та потренуватися збивати наступні цілі, але якщо людина не хоче витрачати свій час, знаючи, що вона вже програла, у неї можливість перезапустити рівень через меню.

Під час проходження рівняння користувачу відображаються дані про кількість ракет кожного типу, що наявні загалом, та кількість ракет, що залишилась. Також кількість ворожих цілей протягом рівня, та кількість збитих ворогів.

1. **ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ЗА ДОПОМОГОЮ ШАБЛОНІВ ПРОЕКТУВАННЯ** 
   1. **Обґрунтування вибору та опис шаблонів проектування для програмної реалізації гри “Glory to Ukraine”**
2. **MVVM**

*Визначення:*

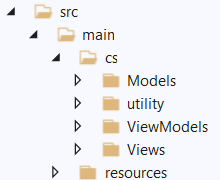
Model-View-ViewModel — [шаблон проєктування](https://uk.wikipedia.org/wiki/%D0%A8%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD%D0%B8_%D0%BF%D1%80%D0%BE%D1%94%D0%BA%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F), що застосовується під час проєктування архітектури [застосунків (додатків)](https://uk.wikipedia.org/wiki/%D0%97%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA" \o "Застосунок). Публічно вперше був представлений Джоном Госсманом (John Gossman) у 2005 році як модифікація шаблону Presentation Model. MVVM орієнтований на такі сучасні платформи розробки, як [Windows Presentation Foundation](https://uk.wikipedia.org/wiki/WPF) та [Silverlight](https://uk.wikipedia.org/wiki/Silverlight" \o "Silverlight) від компанії [Microsoft](https://uk.wikipedia.org/wiki/Microsoft).

*Обґрунтування використання шаблону:*

Оскільки цей шаблон є підвидом багатошарової архітектури для WPF, то до нього можна застосувати властивості цієї архітектури.

Основні випадки коли застосовується багатошарова архітектура:

* Є хорошим вибором для невеликих простих програм та вебсайтів
* як відправна точка, для ситуацій з дуже жорстким бюджетом і часовими обмеженнями.
* коли архітектор все ще аналізує бізнес-потреби і вимоги та не впевнений, який архітектурний стиль буде найкращим.

Випадок розроблення курсової роботи як раз підлягає цим критеріям.

*Рис. 2.1.1. Cтруктура MVVM*

Учасники шаблону:

* Види:
  + MainWindow.xaml і MainWindow.xaml.cs
  + ChooseLevel.xaml і ChooseLevel.xaml.cs
  + Guid.xaml і Guide.xaml.cs
  + Field.xaml і Field.xaml.cs

За допомогою прив’язки властивостей та прив’язки до команд відповідної Вид-Моделі взаємодіє з Вид-Моделлю.

* Вид-Моделі:
  + BaseViewModel – клас батько для всіх Вид-Моделей, реалізовує інтерфейс InotifyPropertyChanged
  + MainWindowViewModel
  + GuideViewModel
  + FieldViewModel

Також класи Вид-Моделей містять екземпляри класів команд,

Детальніше буде в описі шаблона Команда.

* Моделі:
  + Класи, що наслідують абстрактний клас AbstactDrone (Дрони)
  + Патрони, що наслідують абстрактний клас AbstractPatron
  + Літаки, що наслідують абстрактний клас AbstractPlain
  + GameStateSingltone, що відповідає за поточний стан ігрового поля
  + А також неосновні допоміжні класи такі як індекатор здоров’я ворога тощо.

**2)**