КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

Кафедра інженерії програмного забезпечення та кібербезпеки

|  |
| --- |
| *Захищено на кафедрі інженерії програмного забезпечення та кібербезпеки*  *«\_14\_»\_\_\_\_травня\_\_\_\_\_\_\_\_\_\_\_\_2020р.*  *з оцінкою\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*  *Підпис членів комісії:*  *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*  *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*  *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* |

**КУРСОВА РОБОТА**

**З ДИСЦИПЛІНИ**

**«ОБ’ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ»**

**НА ТЕМУ:**

**«Розробка програмного додатку домашня фільмотека мовою програмування С#»**

|  |  |
| --- | --- |
|  | **Виконав:** студента факультету інформаційних технологій  2 курсу 12 групи  **Голинський Андрій Миколайович.**  **Науковий керівник:**  асистент кафедри інженерії програмного забезпечення та кібербезпеки  **Гнатченко Дмитро Дмитрович** |

Київ 2021

Київський національний торговельно-економічний університет

Кафедра інженерії програмного забезпечення та кібербезпеки

**КАРТКА**

**завдання та контролю за ходом виконання курсової роботи**

1. Голинський А.М.
2. Факультет інформаційних технологій(ФІТ)
3. Курс, група 2 курс 12 група
4. Форма навчання денна
5. Номер залікової книжки №34/19-275
6. Номер та назва теми курсової роботи Розробка програмного додатку домашня фільмотека мовою програмування C#

1. Дата погодження теми з науковим керівником  21.02.2021
2. Дата затвердження плану роботи 21.02.2021
3. Дата подання виконаної роботи на кафедрі 07.05.2021
4. Дата захисту роботи 13.05.2021
5. Тему, план і строки прийняв до виконання

(підпис студента)

1. Науковий керівник  Гнатченко Дмитро Дмитрович

(прізвище та підпис)

“  ”  20  р.

Зміст

[**Вступ** 4](#_Toc71749218)

[**РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.** 6](#_Toc71749219)

[**1.1.** **Статистика кінематографу.** 6](#_Toc71749220)

[**1.2.** **Визначення структури та особливостей пошуку медіа продукту на просторах інтернет.** 9](#_Toc71749221)

[**1.3.** **Власні дослідження.** 10](#_Toc71749222)

[**Висновок розділ 1** 11](#_Toc71749223)

[**РОЗДІЛ 2. ОЗНАЙОМЛЕННЯ ІЗ ОСНОВОЮ ПРОГРАМУВАННЯ.** 12](#_Toc71749224)

[**2.1.** **Мова програмування, її особливості.** 12](#_Toc71749225)

[**2.2.** **Основні вимоги до розробки програмного додатку.** 19](#_Toc71749226)

[**2.3.** **Середовище та інструменти розробки.** 21](#_Toc71749227)

[Висновок до розділу 2. 24](#_Toc71749228)

[**РОЗДІЛ 3.РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ ДОМАШНЯ ФІЛЬМОТЕКА.** 25](#_Toc71749229)

[**3.1** **. Налаштування віртуального середовища розробки.** 25](#_Toc71749230)

[**3.2** **. Технології використання розробленого програмного продукту.** 26](#_Toc71749231)

[**Висновок до розділу 3** 33](#_Toc71749232)

[**ВИСНОВОК** 34](#_Toc71749233)

[**СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ** 35](#_Toc71749234)

[**ДОДАТКИ** 37](#_Toc71749235)

# **Вступ**

**Анотація:** Загальний обсяг роботи склав 34 сторінки, де знаходиться 22 рисунки і 5 додатків. Розглядаються такі основні поняття як: об’єктно – орієнтоване програмування, базові поняття об’єктно – орієнтованого програмування, мови об’єктно – орієнтованого програмування, мова програмування C#, Visual Studio. Сутність роботи полягає в розробці додатку для, пошуку, збереження та відтворення фільмів.

Моя курсова робота присвячена розробці домашньої фільмотеки мовою програмування С#, адже в сучасному світі, а особливо в період пандемії, хочеться розслабитись та відволіктись від проблем буденності, пірнути в тепле покривало з чашкою гарячого какао, та поринути в світ кіно. На мою думку, кожному потрібна така розрядка. Та існує проблема, яка заважає нам насолоджуватись дарами сучасної кінематографі.

Проблема полягає в тому, що ми витрачаємо безліч дорогоцінного часу на пошук ресурсу, який дасть змогу подивитись улюблений фільм без реклами, в хорошій якості та в приємному перекладі, якщо фільм зарубіжний. Адже, щоб сповна насолодитись переглядом, потрібно відключитись від зовнішнього світу і щоб нічого не відволікало. Адже в кожному фільмі зкладений не тільки розважальний сенс, але й повчальний. На мою думку, найбільш повчальними є фільми про техногенні катастрофи, на приклад серіал « Чорнобиль» знятий одним із найбільших телевізійних гігантів каналом HBO, або, фільм «Хмарочос»який відноситься до ряду наукової фантастики, і показує нам**,** що не можна стрибнути вище голови, а також що навіть в накращих програмах може бути помилка, і тут як ніде можна побачити важливість такої спеціальності, як «Кібербезпека». Адже саме через похибку в програмі, хакери змогли взяти контроль і зруйнувати все. Також надзвичайно крутий фільм що вартий уваги це «Небезпечна правда» фільм який вийшов в 2001 році має правдоподібні кадри створнення програм зі справжніми консольними командами та демонстрацією складної і довгої роботи, замість звичайного тикання по клавішам.

І не просто так я торкнувся телеканалу HBO адже вони зараз в процесі створення онлайн-кінотеатру, тобто своєї фільмотеки.

Завдання курсової роботи полягає у розробці програмного продукту, що, забезпечить комфортний перегляд фільмів в онлайн режимі.

**Метою дослідження даного курсового проекту** є розробка програмного додатку домашня фільмотека.

**Актуальність обраної тематики** дослідження заключається в необхідності отримання нової інформації та саморозвитку через додаткові ресурси такі як медіа, а також зменшення витрат часу на пошук потрібного медіа продукту в задовільних рамках.

**Практичне значення** курсової роботи полягає у розробці ресурсу що дозволить ознайомитись з жанрами і роками випуску фільмів та серіалів а також готове посилання для зручного прегляду користувачами, як в оригіналі з субтитрами, для вивчення іноземних мов, так в приємному вуху перекладі.

**Інформаційною базою** дослідження є науково популярні статті, доклади асоціацій кіновиробництва, а також інтернет ресурси, дані з популярних стрімингових сервісів.

# **РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.**

* 1. **Статистика кінематографу.**

Кінематограф- це синтетичний за своє природою вид мистецтва, який поєднує в собі елементи літератури, театру, живопису, музики, хореографії. Кіномистецтво дає можливість безпосередньо відображати дійсність у її просторово-часовій єдності, зображає дійсність рухомою, динамічною.[1]

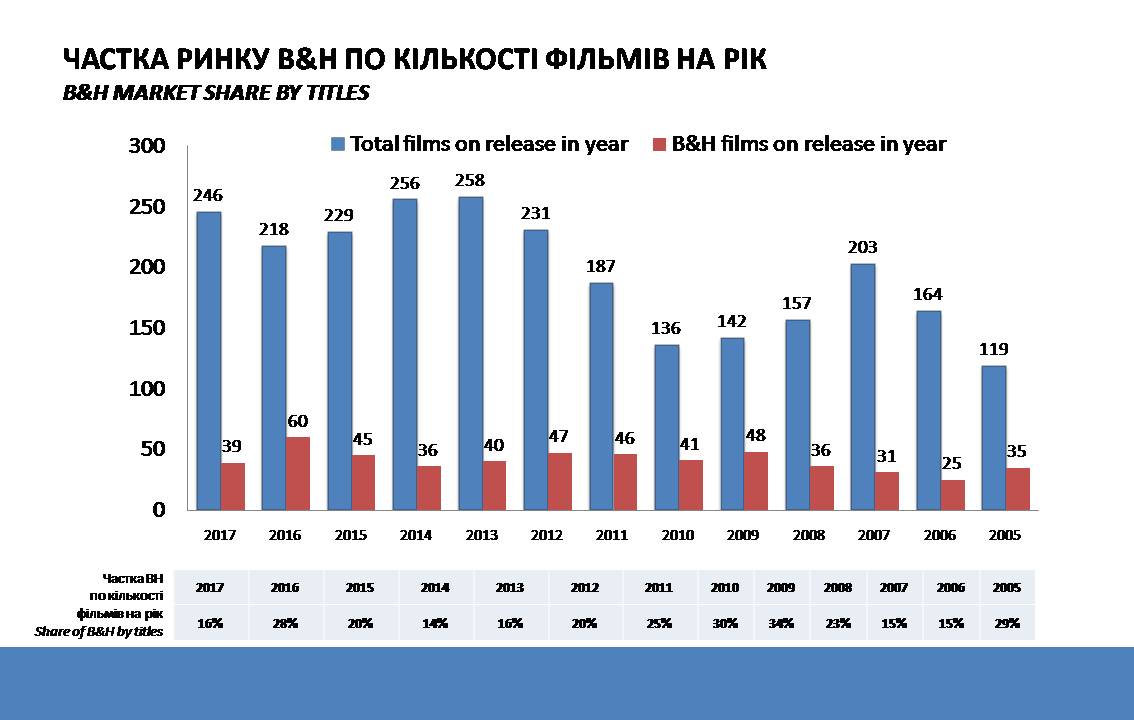
Появу кіно стимульовали соціальні особливості новітнього часу, потреба відобразити та художньо осмислити широкий розмах народних рухів, утягування в творення історії мільйонних мас, загальне зростання динаміки життя, розширення та поглиблення взаємозалежності різних процесів (швидке переміщення дії в географічному просторі; зв’язок подій, що відбулися в одному кінці земної кулі, з подіями в інших її кінцях; взаємодія людини з різноманітними сферами дійсності), а також потребою в фотогенічності та портретуванні духовного життя людини. На виражальні можливості кіно вплинули також досягнення традиційних мистецтв.[1]

Також словом кінематографія позначають інколи такі терміни

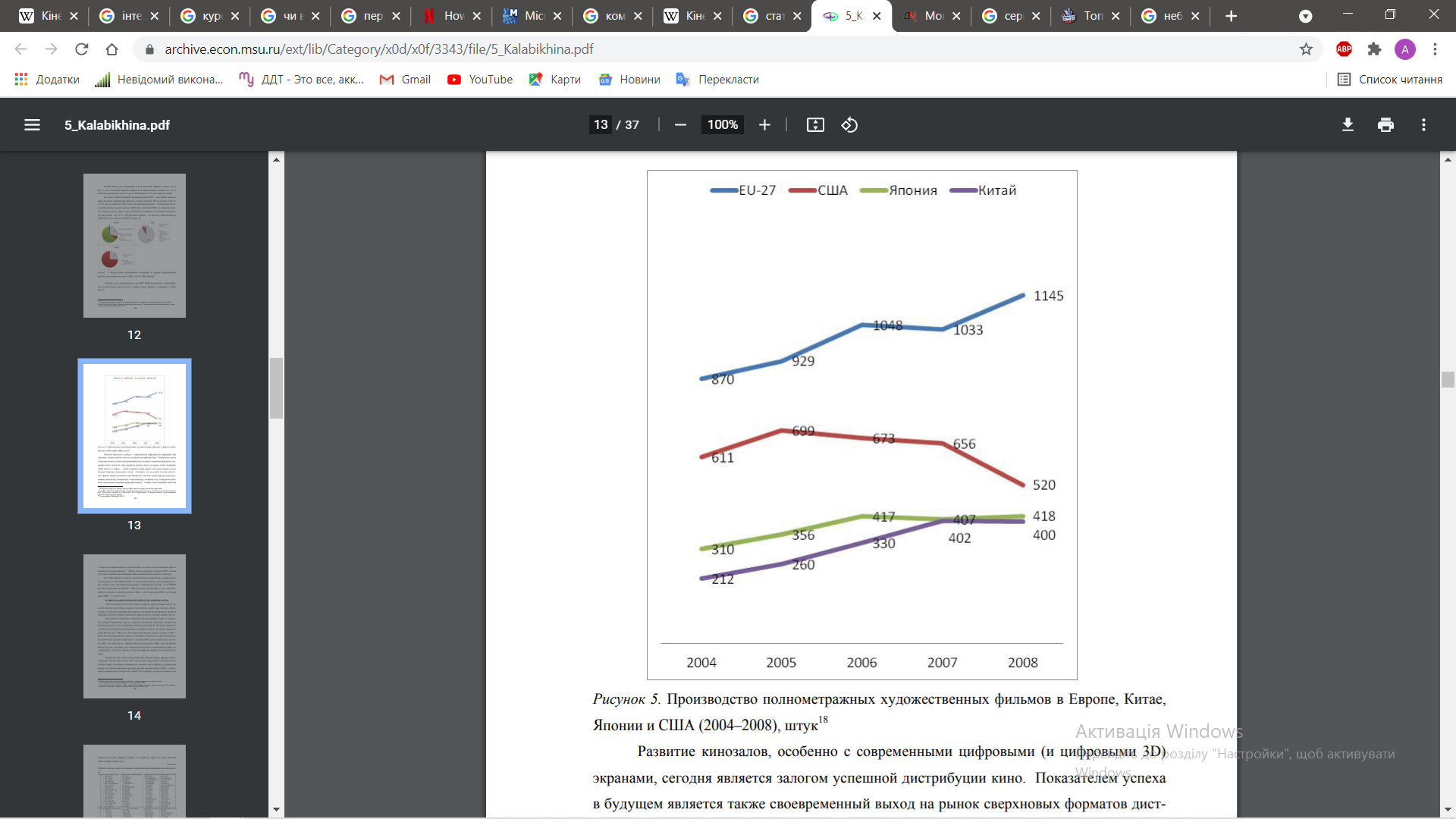
* Кіномистецтво — вид художнього мистецтва, що з’явився у 19 столітті і став найпопулярнішим у 20 столітті.
* Кінознімання — збір пристроїв та методів, що використовуються для знімання рухомих об'єктів на кіноплівку, а також для відображення відзнятих кадрів, шляхом проєктування їх на екран.

Так само термін кінематографія можуть називати сінематографія, сінематограф та кінематограф.

Щодня зарубіжний та український кінематографи створюють нові проекти, їх кількість надзвичайно різко зростає і це можна побачити на прикладі діаграми на рисунку 1.1. та 1.2.



**Рис. 1.1. Кількість фільмів на рік, що випускаються в Україні.[2]**



**Рис. 1.2. Кількість фільмів на рік що випускаються в світі[3]**

Та все таки в наш час, а особливо в період пандемії надзвичайно стрімко почали набувати популярність фільмотеки та онлайн-кінотеатри. На рисунку 1.3. показані топ 10 в країнах СНГ.



**Рис. 1.3. Топ 10 онлан-кінотеатрів СНГ.[4]**

На даний момент найпопулярнішими в світі:

1 місце – Netflix на даний момент у нього понад 1 мільярд завантажень в Play мвркет.

2 місце - Amazon video.

3 місце ділять між собою – Apple TV+ та Disney+.

Та це сервіси на яких можна побачити продукт власного виробництва.

Тому в Україні зараз найпопулярнішими є «Megogo» та «Sweat TV».

На даних сервісах можна переглядати будь-які фільми та серіали, а також ТВ канали. Та в них є один значний недолік, на жаль не можна здійснювати сортування за жанром чи за роком випуску.

* 1. **Визначення структури та особливостей пошуку медіа продукту на просторах інтернет.**

Досліджуючи дану область можна скласти таку структуру дій:

* Спочатку потрібно визначитись з тим який медіа продукт обрати, і тут є декілька підпунктів:

1. Рік випуску.
2. Жанр.

Завдяки цим пунктам здійснюється перша частина процесу.

* Потім в роботу входить зоровий спектр, ми завжди опираємось на те що ми бачимо, і візуальна частина дуже важлива, отож, якщо нам не сподобається обкладинка фільму, малоймовірно, що ми будемо його дивитись.
* На третьому місці у нас стоїть опис фільму, адже він, як і початок книги повинен заінтересувати нас.

Після цих пунктів ми починаємо пошук ресурсу на якому є потрібний продукт.

І тут виникає найбільше проблем, адже існує понад 1000 веб-сайтів що дають нам можливість дивитись фільми і серіали. Потрібно вибрати найзручніший, з найменшою кількістю реклами, а також той де буде хороший переклад.

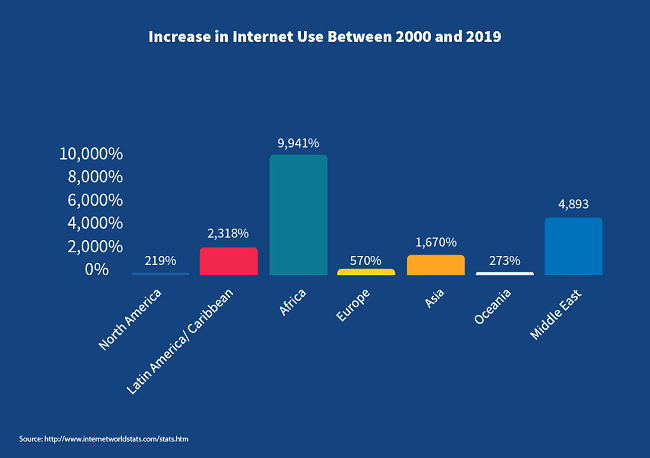
На даний час гаджети стали невід’ємною частиною нашого життя, завдяки ним ми спілкуємось, навчаємось, саморозвиваємось, робимо фото, в них ми можемо записати те зберегти гігабайти інформації, та що там зараз наручний годинник від компанії Apple може рахувати не тільки спалені калорії, кількість кроків, пройдену відстань та пульс, він може визначати рівень кисню в крові що дуже важливо в період бушування Коронавірус 2019-nCoV. Сучасні смартфони дозволяють нам вийти в мережу Інтернет де б ми не знаходились, чи то на Говерлі, чи в Метрополітені. І ми проводимо в ньому безліч часу.

ВВП країн, як видається, сильно співвідноситься з проникненням Інтернету, при цьому найбагатші країни мають вищу пропускну здатність в Інтернеті і навпаки. Різниця між користувачами Інтернету в Північній Кореї та Катарі становить 98,4%.[5]

Незважаючи на розміри та населення, Північна Корея має проникнення Інтернету всього 0,06%, а в Катарі та Об’єднаних Арабських Еміратах аж 99% громадян мають доступ.[5]

За останні 18 років в Африці спостерігається значне збільшення майже на 10 000%, а в Північній Америці – на 219%.[5]

Дана інформація наведена на рисунку 1.4.

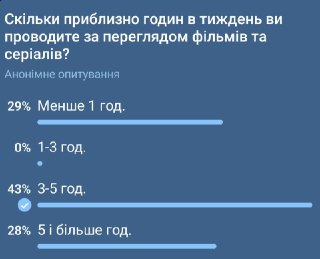


**Рис. 1.4. Збільшення використання Інтернету між 2000 та 2019 рр.[5]**

Використання Інтернету є найпоширенішим у Північній Америці, а також у Північній та Західній Європі, де рівень поширення становить більше 90%, а в Центральній та Східній Африці залишається низьким, менше 20%.Головним чином це пов’язано з меншою доступністю інфраструктури.[5]

* 1. **Власні дослідження.**

На жаль, на даму тему майже не проводилось досліджень, тому при вивченні літератури що можливо знайти на просторах мережі Інтернет, вимальовується картина високого попиту на фільмотеки. За даними мого опитування найбільше 43% людей проводять за переглядом фільмів та серіалів 3-5 годин на тиждень, і ще 28% більше 5 годин. Рисунок 1.5.



**Рис.1.5. Кількість годин на тиждень, які студенти КНТЕУ витрачають на перегляд фільмів.**

*Джерело: побудоване автором ( знімок екрану)*

Також я провів дослідження по зручності роботи з фільмотеками, і майже всі вони відрізняються, на приклад Netflix, Disney+, Apple TV дають нам змогу переглядати тільки медіа продукт власного виробництва, на відміну від додатку Zona. Даний додаток дозволяє вибирати нам серед продуктів будь якої компанії, та на жаль це піратська програма, яка має надзвичайно низький рівень захисту, і може стати причиною виведення з ладу вашого пристрою, за допомогою вірусів. Можливо, серед всіх схожих ресурсів, я б назвав Megogo, даний ресурс потребує не велику щомісячну оплату, та недоліками, я б назвав великий інтернет трафік, та досить не велику базу фільмів та серіалів, звісно ж Sweet TV, та Київстар ТВ нічим не відрізняються але зручніший і приємніший інтерфейс на мою думку саму у Megogo.

## **Висновок розділ 1**

Підведемо підсумки, в Україні як і в усьому світі, кількість медіа продукту, що створюється, постійно збільшується, пошук потрібного стає складнішим. І це все створює велику кількість «неораної землі» з якою так і хочеться, працювати.

# **РОЗДІЛ 2. ОЗНАЙОМЛЕННЯ ІЗ ОСНОВОЮ ПРОГРАМУВАННЯ.**

* 1. **Мова програмування, її особливості.**

Мова програмування є основною в написанні будь якої програми, це як і в спілкування не знаючи мови не можливо щось сказати щось так щоб тебе зрозуміли, так і в програмуванні, не знаючи мови система не зрозуміє задачу, яку потрібно виконати.

Мова програмування — це штучна мова, створена для передачі команд машинам, зокрема комп’ютерам. Мови програмування використовуються для створення програм, котрі контролюють поведінку машин, та запису алгоритмів.[10]

Більш строге визначення: мова програмування — це система позначень для опису алгоритмів та структур даних, певна штучна формальна система, засобами якої можна виражати алгоритми. Мову програмування визначає набір лексичних, синтаксичних і семантичних правил, що задають зовнішній вигляд програми і дії, які виконує виконавець (комп’ютер) під її управлінням.[10]

З часу створення перших програмованих машин було створено понад дві з половиною тисячі мов програмування. Щороку їх кількість поповнюється новими. Деякими мовами вміє користуватись тільки невелике число їх власних розробників, інші стають відомі мільйонам людей. Професійні програмісти зазвичай застосовують в своїй роботі декілька мов програмування.[10]

Без мов програмування світ в нинішньому вигляді просто не існував би. Уявіть, що ми живемо без комп’ютерів, цифрових технологій, інтернету з Instagram і TikTok. Уявити таке дуже складно, якщо взагалі можливо.[11]

Але скільки мов програмування існує взагалі? Щоб відповісти на це питання, потрібно провести серйозне дослідження. Причому варто відразу відокремити зерна від полови, тобто визначити, що є мовою програмування (МП), а що ні. Залежно від цього різні джерела дають найрізноманітніші відповіді. Якщо коротко, то від 1000 до 10 тисяч. Все залежить від того, що визнавати мовою програмування і чи визначати різновиди (якщо завгодно, форми) однієї МП як різні мови. Але навіть якщо так не робити, то нарахувати можна близько 700 мов: від найстаріших до сучасних, від практичних і використовуваних до езотеричних, створюваних просто для фана.[11]

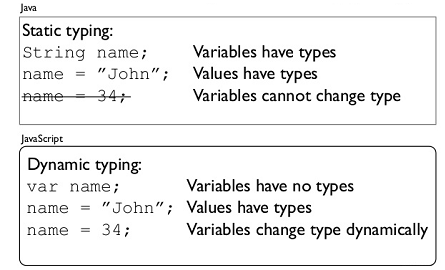
Більш того, нові МП з’являються із регулярністю, якій можна позаздрити – багато компаній розробляють власні мови. Експериментують з цим і окремі розробники.[11]

У всіх мов своє призначення. Якісь застосовуються для написання мобільних додатків, інші – для роботи з мікроконтролерами, треті – для створення комп’ютерних програм або аналітики. Але є і випадки, коли одну задачу можна вирішити за допомогою різних мов.[11]

**Які бувають мови програмування**

Мови програмування класифікують за різними ознаками: функціональністю, способом опису сутностей, способом реалізації та деякими іншими.Рисунок 2.1.[11]

**Статична і динамічна типізація**

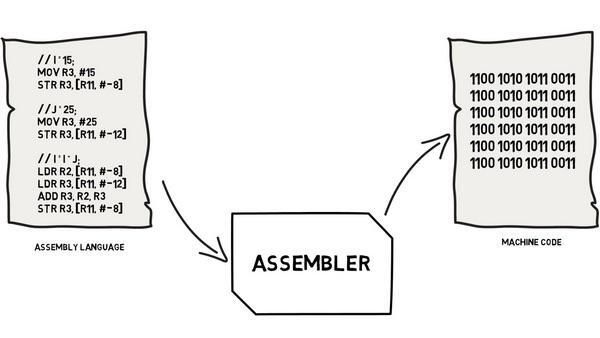


**Рис. 2.1. Різниця властивостей МП Java та JavaScript.[11]**

Тут мова йде про процес перевірки відповідності типів. Система типів – це набір правил, які присвоюють властивість («тип») ряду сутностей: змінним, виразам, функціям, модулям. У статичній типізації змінним типи присвоюються в момент оголошення – так працюють Go, С ++, C #, Java і ряд інших мов. При динамічній типізації це відбувається в момент надання значення, як в PHP, Python, JavaScript, Ruby і подібних МП.[11]

**Високорівневі та низькорівневі мови**

Тут все простіше. Як правило, до низькорівневих мов відносяться МП перших поколінь. Це машинний код і асемблери, які (Рис.2.2.) з’явилися на зорі цифрової ери. Тоді гори були високими, дерева зеленими, а ЕОМ займали цілі будівлі.[11]



**Рис. 2.2. Перші мови програмування.[11]**

Але в міру технічного прогресу завдання ускладнювалися. Щоб їх вирішувати, люди вдосконалювали обчислювальні пристрої. Тоді й було розроблено високорівневі мови, що дозволили прискорити створення ПЗ і збільшити ефективність роботи ЕОМ. Однією з перших була Fortran.

Зараз виділяють ще й надвисокорівневі мови з дуже високим рівнем абстракції. До них відносять, наприклад, Python і Perl.[11]

**Інтерпретовані та компільовані мови**

Це розділення вельми умовне. Головна відмінність – компільованим мовам для виконання програми потрібен машинний код, а інтерпретованим – ні. Замість цього програму порядково виконують інтерпретатори. Раніше компільовані мови працювали швидше інтерпретованих, але зараз цей розрив поступово скорочується завдяки таким технологіям, як JIT-компіляція.[11]

Приклади компільованих мов: Pascal, Basic, Visual Basic, Cobol, Swift, Go, Haskell, C#, Kotlin. Інтерпретовані: PHP, JavaScript, Python, Lava, Perl.[11]

В останні роки, професія програміста стало однією із найбільш популярних, і що року в ВНЗ вікривається все більше нових спеціальностей, збільшується кількість абітурієнтів, а отже зростає попит на розширення робочої та навчальної області.

Та через надзвичайно велику кількість МП важко визначитись, яку вивчати,

Ось список 5 найпопулярніших мов програмування.

**Java,** **Javascript,** **HTML та CSS,** **РНР,** **C++**

До даного списку, я ще б додав:

**С**

Це мова програмування загального призначення, що включає економiю представлення, сучасне керування потоком i структурою даних i багатий набiр операторiв. C не є мовою «дуже високого рiвня», нi «великою» мовою, i не призначена для певної областi застосування. Але вiдсутнiсть в нiй обмежень та її загальнiсть роблять її зручнiшою i ефективнiшою для багатьох завдань, у порiвняннi з мовами, що вважаються потужнiшими.[13]

**С#**

проста, потужна, статично типізована, об'єктно орієнтована мова програмування від компанії Microsoft. C # входить в сімейство мов програмування C, синтаксис мови здасться знайомим програмістам, які працювали з C, C ++, Java і JavaScript.[14]

Перша версія мови C # була створена в 1998-2001 роках, групою інженерів Microsoft під керівництвом Андреса Хейлсберг і Скотта Вільтаумота, як основна мова програмування платформи Microsoft .Net.[14]

C # увібрав в себе кращі властивості мов C, C ++, Modula, Object Pascal, спираючись на досвід їх практичного використання. Деякі проблемні моделі, які до цього використовувалися в мовах програмування, наприклад множинне спадкування класів (використовується в С ++), були свідомо вимкнені.[14]

Багато в чому мова C # дуже схожа на Java, ця подібність помітно в синтаксисі і основні поняття цих мов програмування.[14]

C # є об'єктно-орієнтованою мовою програмування і відноситься до родини мов з C-подібним синтаксисом, з них його синтаксис найбільш близький до C ++ і Java. Перейнявши багато від своїх попередників - мов C ++, Pascal, Модула, Smalltalk і особливо Java, С # при цьому відмовляється від деяких проблемних моделей (не можна використовувати множинне спадкування класів, яке може привести до семантичної невизначеності).[18]

На даний момент актуальна версія мови - C # 7.0, але в рамках розглянутого нижче програмного забезпечення використовується більш стара версія - C # 3.0.

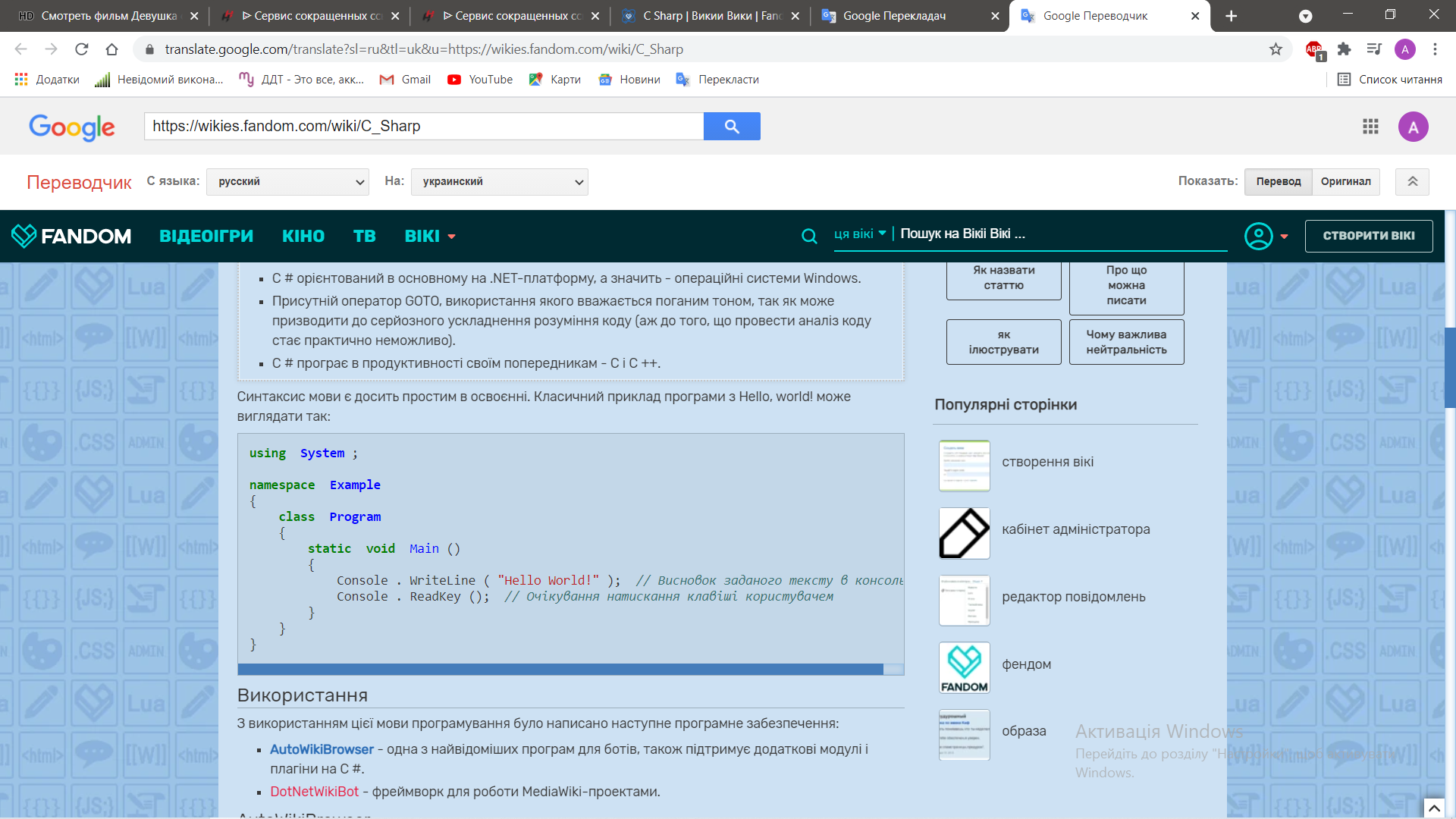
**плюси:**

* Невисокий поріг входження: для новачків дана мова програмування є одним з найбільш перспективних.
* Велика кількість «синтаксичного цукру» - конструкцій, які створені для полегшення написання і розуміння коду.
* Присутня можливість працювати з пам'яттю безпосередньо (що, втім, може бути і мінусом).
* Висока популярність мови і велика спільнота розробників.

**мінуси:**

* C # орієнтований в основному на .NET-платформу, а значить - операційні системи Windows.
* Присутній оператор GOTO, використання якого вважається поганим тоном, так як може призводити до серйозного ускладнення розуміння коду (аж до того, що провести аналіз коду стає практично неможливо).
* C # програє в продуктивності своїм попередникам - C і C ++.[18]

Синтаксис мови є досить простим в освоєнні. Класичний приклад програми з Hello, world! Дивитись рисунок 2.3.[18]



**Рис.2.3. Один із класичних прикладів програми.[18]**

**Використання**

З використанням цієї мови програмування було написано наступне програмне забезпечення:

* **AutoWikiBrowser** - одна з найвідоміших програм для ботів, також підтримує додаткові модулі і плагіни на C #.
* DotNetWikiBot - фреймворк для роботи MediaWiki-проектами.[18]

AutoWikiBrowser

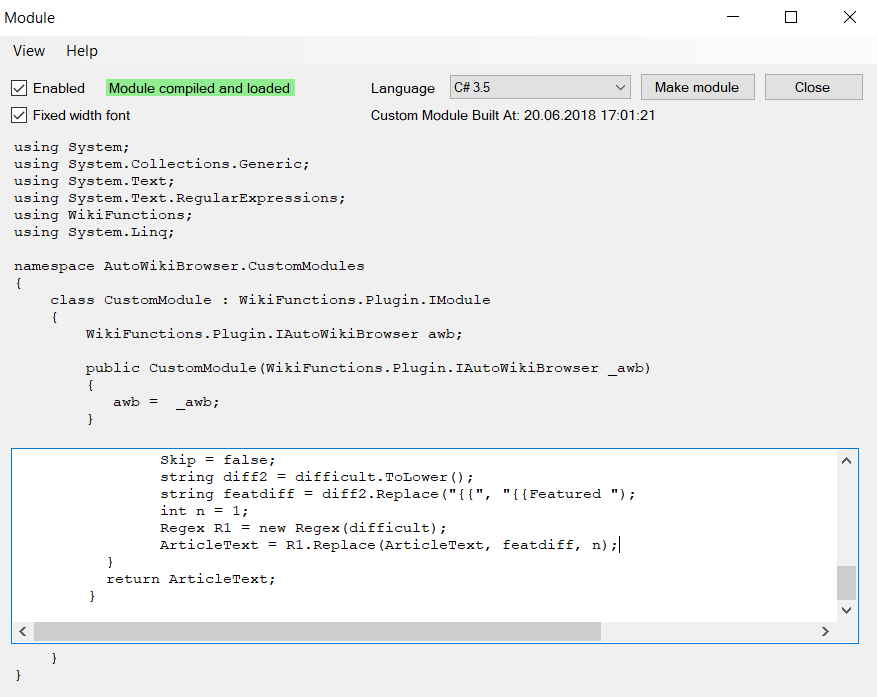
Як вже було сказано раніше, програма написана на мові програмування C # з використанням .NET Framework 3.5 (який необхідний для роботи програми).[18]

У AutoWikiBrowser'е використовуються наступні бібліотеки (набір класів, що містять певний код):

* **WikiFunctions** - бібліотека, яка містить велику кількість різноманітних класів. Містить в собі великий функціонал для самих різних моментів при роботі з програмою - методи для входу в обліковий запис, генерація списків, форматування тексту для подальшої роботи і багато іншого.[18]
* **Newtonsoft.JSON** - бібліотека, призначена для формування JSON (текстовий формат обміну даними, заснований на JavaScript) і подальшої роботи з ним, наприклад його перетворення або валідації.[18]

Модулі

AutoWikiBrowser дозволяє завантажувати додаткові **плагіни** , які дають нові можливості при використанні програми. При скачуванні останньої версії AWB з офіційного репозиторію  в окремій папці вже є кілька плагінів. Для того щоб їх використовувати, потрібно або помістити їх в папку, де здійснюється запуск AWB, або ж через додаток вибрати потрібні для завантаження плагіни, дивитись на рисунку 2.4. [18]

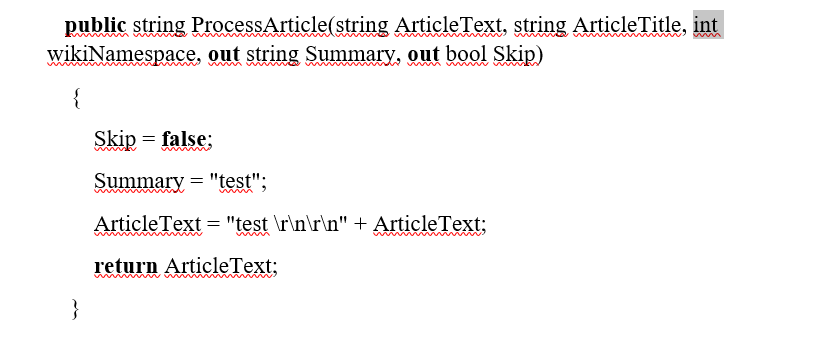


**Рис.2.4. Вибір потрібних до завантаження плагінів[18]**

Приклад модуля в AutoWikiBrowser

При необхідності учасник може сам написати необхідний йому плагін. Для цього учаснику знадобиться середовище розробки, яка підтримує C # (наприклад Visual Studio), а також бібліотека **WikiFunctions**.[18]

Якщо оператору бота необхідно реалізувати свій спосіб обробки статей, він може використовувати **модулі** . Оператору необхідно написати свій код (на мові C # 3.5 або VB .NET 2.0), який буде по-своєму змінювати текст статті. За замовчуванням модуль містить наступний код, дивитись на рисунку 2.5.[18]



**Рис.2.5. Код модуля.[18]**

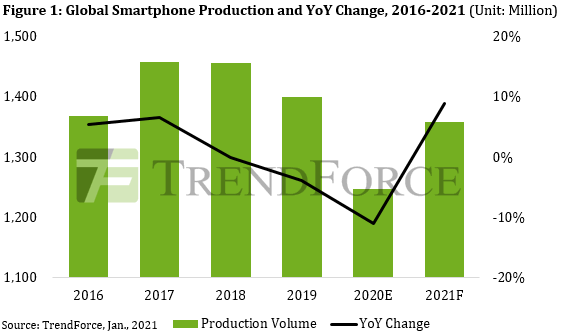
Ключовим у цьому коді є метод **ProcessArticle** , на вхід якого подається текст статті (ArticleText), її назва (ArticleTitle), номер простору імен (wikiNamespace). Усередині методу ми можемо поставити опис правки (використовуючи параметр Summary) і вказати, чи потрібно пропустити цю статтю, чи ні (використовуючи параметр Skip). В результаті роботи буде сформовано новий текст статті на основі старого (в прикладі вище на самому початку буде додано слово test і два розриву рядків) і переданий на наступні етапи обробки.[18]

* 1. **Основні вимоги до розробки програмного додатку.**

Якщо ще декілька століть тому людей охоплювала Золота лихоманка, то зараз світ цілком і повністю охоплений мобільною лихоманкою.зараз не знайдеться ні одної людини в оточенні, яка б не користувалась смартфоном.

Та що там навіть компанія Tesla, на чолі з Ілоном Маском, запустили проект свого смартфона. Новий пристрій від компанії тесла переверне весь світ, тхнологія Neuralink зможе допомогти людям отримавшим травму головного мозку, і інтерфейс мозк-комп’ютер зможе навіть замінити деякі функцію мозку.

Що дня кількість продажу смартфонів зростає, хоча в 2020 році світовий ринок смартфоні різко упав, і причиною цьому став жорсткий локдавн. Дану статистику ми можемо побачити на рисунку 2.6.



**Рис.2.6. Статистика продажу смартфонів.[15]**

При створенні мобільного додатку у розробника є основні завдання:

* створення ТЗ (технічного завдання) на розробку мобільного додатку;
* обговорення з замовником етапів і процесу роботи проекту;
* побудова архітектури додатку;
* безпосередньо програмування;
* робота з дизайнерами;
* підтримка мобільних додатків;
* робота з тестувальниками над налагодженням і тестуванням додатків;
* допомога в створенні інструкцій по роботі з готовим додатком;
* оформлення документації;
* розміщення додатків в AppStore і Google Play Market, Amazon Appstore, Opera Mobile Store і інших магазинах мобільних додатків.[16]

Розробник комп’ютерних програм створює програми для виконання конкретних завдань, які ставляться замовником програми.[17]

Метою роботи розробника програм є створення та підготовка до використання на практиці розробленої програми. Для розробки комп’ютерної програми він аналізує технічну документацію, функціональні та експлуатаційні вимоги до програми, розробляє нові або використовує існуючі алгоритми та структури даних.[17]

У розробника комп’ютерних додатків основні завдання відрізняються ніж у розробника мобільних.[17]

**Професійні обов’язки**

* вивчення припущень, функціональних та експлуатаційних вимог до програми;
* аналіз замовленого проекту та додатків до нього, підготовка та аналіз алгоритмів;
* створення нових та використання існуючих структур даних;
* створення та зміна коду програми у вибраному середовищі та мовою програмування;
* тестування коду для виявлення та усунення помилок та забезпечення надійності та безпеки комп’ютерної програми;
* підготовка технічної документації та інструкцій для користувача;
* підготовка додатків до алгоритму встановлення та запуску розробленої програми;
* організація робочого місця відповідно до принципів охорони праці, техніки безпеки, протипожежного захисту та охорони навколишнього середовища.[17]
  1. **Середовище та інструменти розробки.**

Я вже розповідав, що саме фільми, такі як «Небезпечна правда» та «Дівчина з татуюванням дракона» спонукали мене поступити на спеціальність пов’язану з **IT**, отож моя спеціальність **Кібербезпека**, а модель програмування ООП (Об’єктно-орієнтовне програмування).

1. Що таке Кібербезпека?

**Кібернетична безпека (кібербезпека)** – стан захищеності критичних об'єктів національної інформаційної інфраструктури та окремих її складових, за якого забезпечується їх стале функціонування і розвиток, своєчасне виявлення, запобігання і нейтралізація кібернетичних загроз в інтересах людини, суспільства, держави.[6]

**Комп'ютерна безпека** — це сукупність проблем у галузі телекомунікацій та інформатики, пов'язаних з оцінкою і контролюванням ризиків, що виникають при користуванні комп'ютерами та комп'ютерними мережами і розглядуваних з точки зору конфіденційності, цілісності і доступності.[7]

Закон України «Про основні засади забезпечення **кібербезпеки** України» дає таке визначення: Закон України «Про основні засади забезпечення **кібербезпеки** України» дає таке визначення: «**кібернетична безпека (кібербезпека)** – це стан захищеності життєво важливих інтересів людини і громадянина, суспільства та держави в кіберпросторі; кібернетичний простір (кіберпростір) – це середовище, яке виникає в результаті функціонування на основі єдиних принципів і за загальними правилами інформаційних, телекомунікаційних та інформаційно-телекомунікаційних систем».[7]

Цей Закон визначає правові та організаційні основи забезпечення захисту життєво важливих інтересів людини і громадянина, суспільства та держави, національних інтересів України у кіберпросторі, основні цілі, напрями та принципи державної політики у сфері кібербезпеки, повноваження державних органів, підприємств, установ, організацій, осіб та громадян у цій сфері, основні засади координації їхньої діяльності із забезпечення кібербезпеки.[8]

Випускникам цієї спеціальності надається досить великий вибір посад на яких вони зможуть працювати після закінчення ВНЗ.

**Об'єктно - орієнтоване програмування (ООП)** – це модель програмування яка базується на стверджені того, що програма це сукупність об’єктів які взаємодіють між собою. Кожен об’єкт в цій моделі є незалежним, і він здатний отримувати, обробляти дані та відправляти ці дані іншим об’єктам. В ООП використано моделі успадкування, модульності, поліморфізму та інкапсуляції.[9]

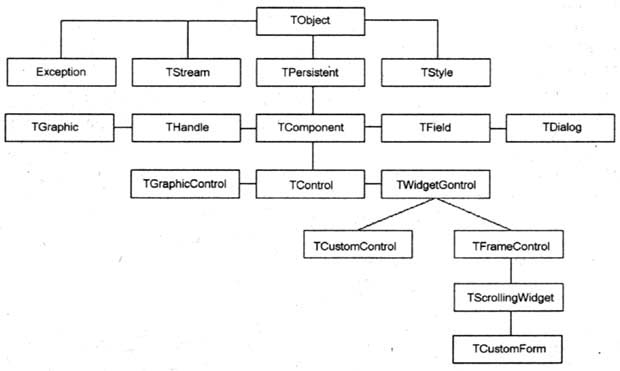
Основним поняттям ООП є об'єкт. **Об'єкт** можна визначити як певну сукупність даних(характеристик об'єкта) та методів роботи з ними. Для класифікації об'єктів у ООП використовують класи. **Клас** служить зразком для створення обєкту, тобто об'єкт є нічим іншим, ніж копією класу.[9]

Кожен об'єкт має процедури і функції(те що він уміє виконувати, наприклад,завантажувати файл, відображати картинку і т.д.), які служать для роботи з даними об'єкта. Ці процедури і функції називаються **методами**.[9]

Існування ООП можливе завдяки трьом основним парадигмам **на яких базується саме ООП:**

-**Інкапсуляція**. Також відома як приховування даних. Зміст інкапсуляції полягає у приховуванні від зовнішнього користувача деталей реалізації об'єкта, замість цього надаючи інтерфейс взаємодії з ним.[9]

-**Успадкування.** Це означає, що об'єкти (класи) можуть переймати деякі властивості у своїх прабатьків. Як? Це залежить від тієї мови, на якому пишеться програма. Однак у будь-якому випадку картина та ж: це призводить до повторного використання вже написаного одного разу коду. Підкласи успадковують атрибути та поведінку своїх батьківських класів, і можуть мати нові власні атрибути. Тобто утворюється ієрархія з класів, де від основного класу(так званого, предка) походять усі інші класи. Приклад такого розгалуженого «дерева» зображено на схемі.[9] Рис.2.7.

[](http://programming.in.ua/images/stories/articles/basis_progm/082.jpg)

**Рис. 2.7. Розгалуження підкласів за успадкуванням.[9]**

-**Поліморфізм** означає залежність поведінки від класу, в якому ця поведінка викликається, тобто, два або більше класів можуть реагувати по різному на однакові повідомлення. Це спричинене зміною в одного з класів якогось методу(процедури, функції), шляхом запису іншого алгоритму. Як приклад, деяка комп'ютерна програма при натисканні клавіші Esc завершить роботу, інша ж програма після натискання кнопки Esc тільки відкриє меню даної програми.[9]

## Висновок до розділу 2.

Ознайомившись із МП, середовищем розробки та вимогами розробника, вибравши мову, та платформу, а також вид програмування, можна сміло приступати до написання програмного додатку, адже все необхідне вже є в пам’яті.

# **РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ ДОМАШНЯ ФІЛЬМОТЕКА.**

1. **. Налаштування віртуального середовища розробки.**

Перш ніж почати роботу над проектом потрібно налаштувати простір, в якому буде створюватись наш програмний додаток.

Працювати над своїм проектом я буду в програмі яку створила компанія Nicrosoft ще в далекому 1997 році, це Visual Studio 2019.

**Microsoft Visual Studio** - повнофункціональна інтегрована середа розробки (IDE) з підтримкою популярних мовних програм програмування, серед яких C, C ++, VB.NET, C #, F #, JavaScript, Python.[19]

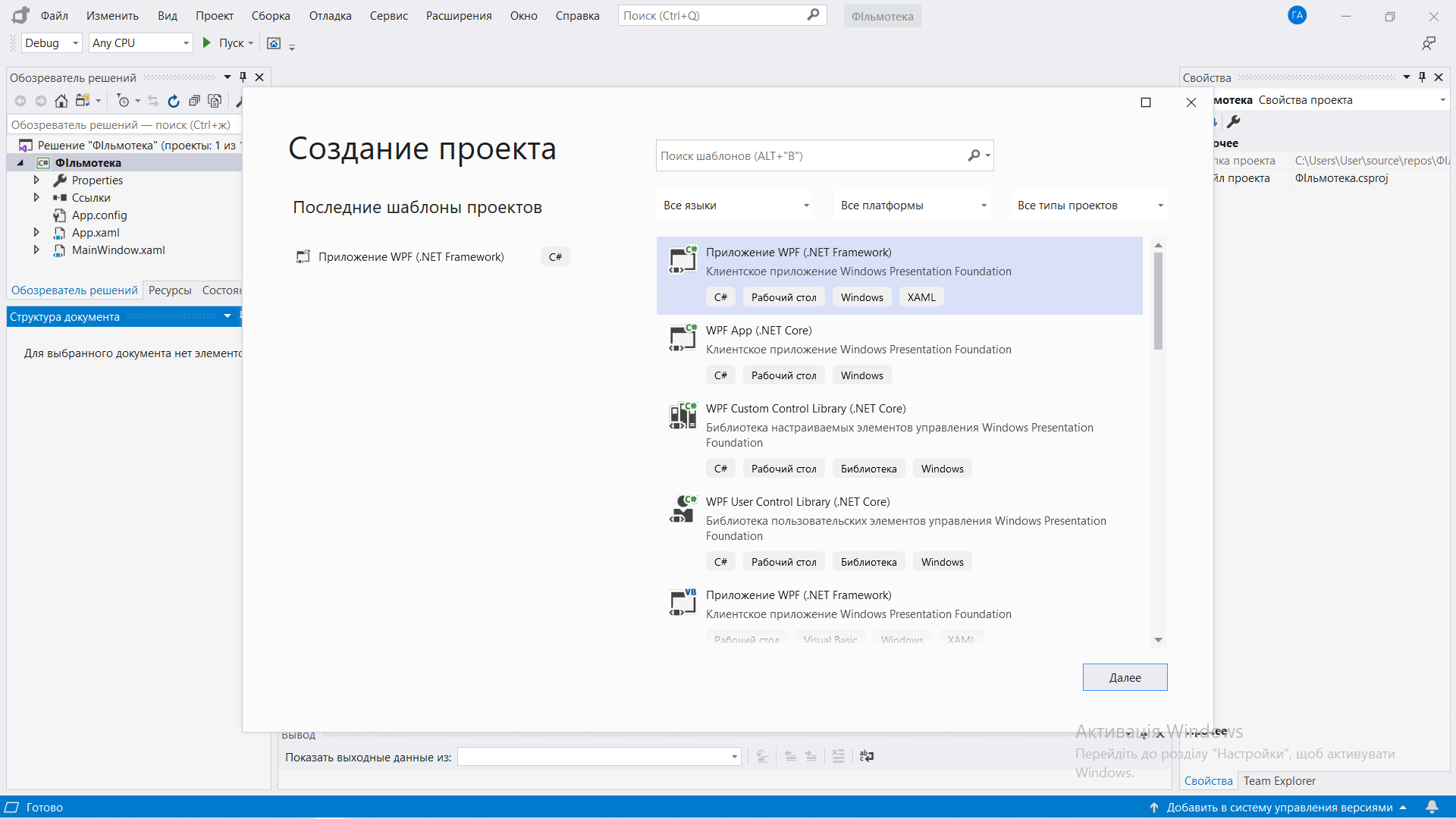
Функціональність Visual Studio охоплює всі етапи розробки програмного забезпечення, надання сучасних інструментів для написання коду, проектування графічних інтерфейсів, збірки, відладки та тестування додатків. Можливості Visual Studio можуть бути доповнені за допомогою підключення необхідних розширень.[19]

Редактор коду Visual Studio підтримує підсвічування синтаксису, вставку фрагментів коду, відображення структур та пов'язаних функцій. Суттєво прискорити роботу допомагає технологія IntelliSense - автозавершення коду за мережею вводу.[19]

Вбудований відладчик Visual Studio використовується для пошуку та виправлення помилок у вихідному коді, у тому числі на низькому рівні пристрою. Інструменти діагностики дозволяють оцінити якість коду з точки зору продуктивності та використання пам'яті.[19]

Дизайнер форм Visual Studio незаленим від розробника програм з графічним інтерфейсом, що допомагає спроектувати зовнішній вигляд майбутнього додатків та роботу кожного елемента інтерфейсу.[19]

Щоб почати роботу над проектом потрібно виконати декілька кроків. Для початку в меню File вибираємо пункт New а потім Projekt. В діалоговому вікні що відкриєтся потрібно вибрати необхідний шаблон, в даному випадку це WPF App (.NET Framework),(рис.3.1.).



**Рис.3.1. Створення проекту і вибір шаблону.**

*Джерело: побудоване автором ( знімок екрану)*

Також необхідно не забути про коректне ім’я проекту і все, можна приступати до розробки програмного додатку.

1. **. Технології використання розробленого програмного продукту.**

Даний програмний продукт побудовано із застосуванням шаблону WPF App (.NET Framework).

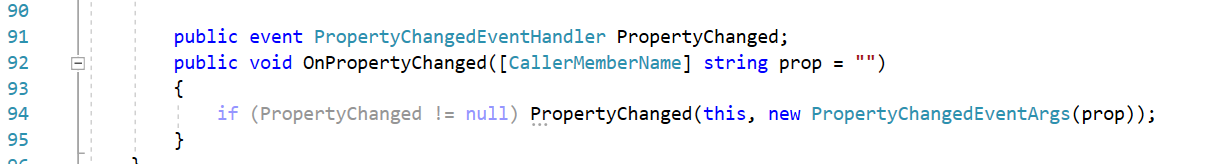
Шаблони проектів та елементів створюють багаторазові заглушки, які надають користувачам деякий базовий код та структуру, що можна налаштувати для своїх цілей.

В розробці програмного продукту я використовував 3 класи:

**Library.cs**

Це база даних в якій вказується **Private, Public** (рисунок 3.2.). Команда **Private** робить так звані поля, межі застосування, **Public** властивості, щоб при завантаженні (**Set**) встановити значення, а також викликати метод **PropertyChangedEventHandler** (рисунок 3.2.). Який передає інтерфейсу інформацію про зміну данних і про те, що потрібно вивести іншу інформацію. В даній бібліотеці використовуються 2 методи **SaveToXML** і **LoadFromXML**, вони записують та зберігають потрібну нам інформіію форматом **XML** в бібліотеці, а також вивантажують її з потрібної нам папки. Тобто за допомогою **XML** парсера відновлює дані по рядках, потім створюється аналогічна ієрархія даних яка була спочатку. І вже в кінці методу **LoadFromXML** застосовуються **tempLib** в якому збирається вся інформація з файлу та відбувається заміна. Таким чином ми виводимо потрібну інформацію на екран замінюючи її без лишніх дій.

**Рис.3.2. Застосування методу PropertyChangedEventHandler**

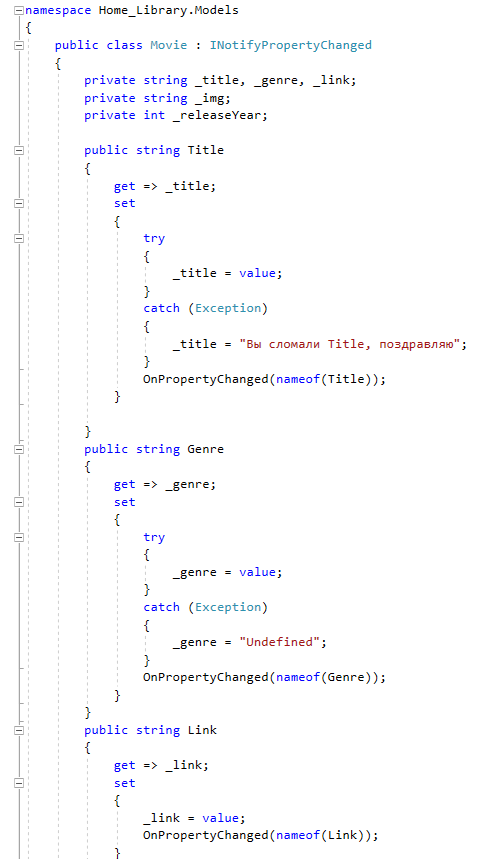


2

*Джерело: побудоване автором ( знімок екрану)*

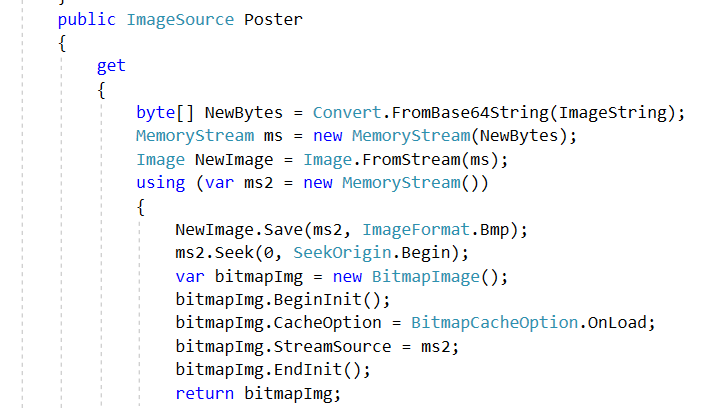
**Movie.cs**

В класі **Movie.cs** прописується інформація даного медіа продукту. Тобто назви полів та вид інформації, що буде в них вказуватись (**String**). В програмі буде 4 рядки в яких буде відображатись назва продукту, жанр, рік випуску та посилання (рис.3.3.). А завдяки **ImageSource Poster** (рис.3.4.) програма розпізнає постер фільму, також форматує зберігає і коли потрібно відображає його на екрані з іншою потрібною інформацією.



**Рис.3.3. Запис даних про вид інформації в рядках.**

*Джерело: побудоване автором ( знімок екрану)*

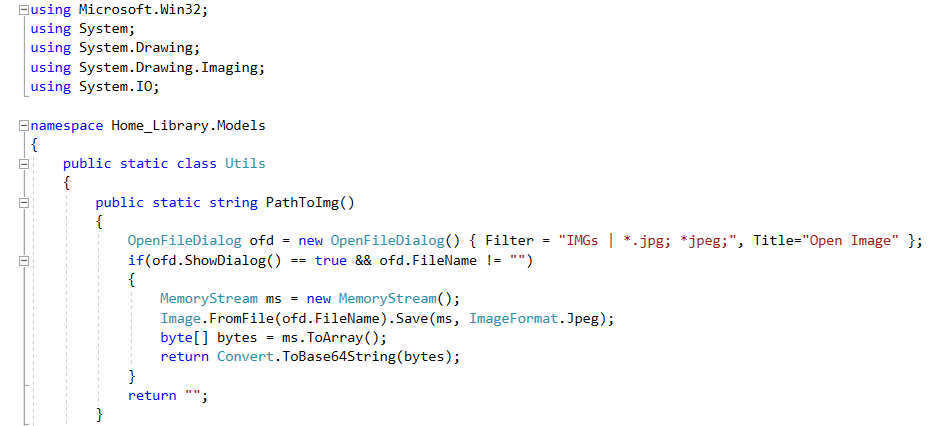


**Рис.3.4. Застосування ImageSource Poster**

*Джерело: побудоване автором ( знімок екрану)*

**Utils.cs**

В даному класі, я використовував один метод це **PathToImg,** (рис.3.5.)**,** якийкодує інформацію фільму в набір символів для його переформатування та збереження в папці, стислим варіантом.



**Рис. 3.5.Використання методу PathToImg**

*Джерело: побудоване автором ( знімок екрану)*

**XMAL**

XAML (eXtensible Application Markup Language) - мова розмітки, яка використовується для ініціалізації об'єктів в технологіях на платформі .NET. Стосовно до WPF (а також до Silverlight) дана мова використовується перш за все для створення призначеного для користувача інтерфейсу декларативним шляхом. Хоча функціональність XAML тільки графічними інтерфейсами не обмежується: дана мова також використовується в технологіях WCF і WF, де вона ніяк не пов'язана з графічним інтерфейсом. Тобто її область ширше. Стосовно до WPF ми будемо говорити про неї найчастіше саме як про мову розмітки, яка дозволяє створювати декларативним шляхом інтерфейс, на зразок HTML в веб-програмуванні. Однак знову ж таки повторюся, зводити XAML до одного інтерфейсу було б неправильно.[20]

**AddMoviePage.xaml.**

За допомогою ми прописуємо область взаємодії, тобто розсташування кнопок, входів, все дуже легко організовується за допомогою сітки, та прив’язки до її окремої ділянки. Рисунок 3.6.

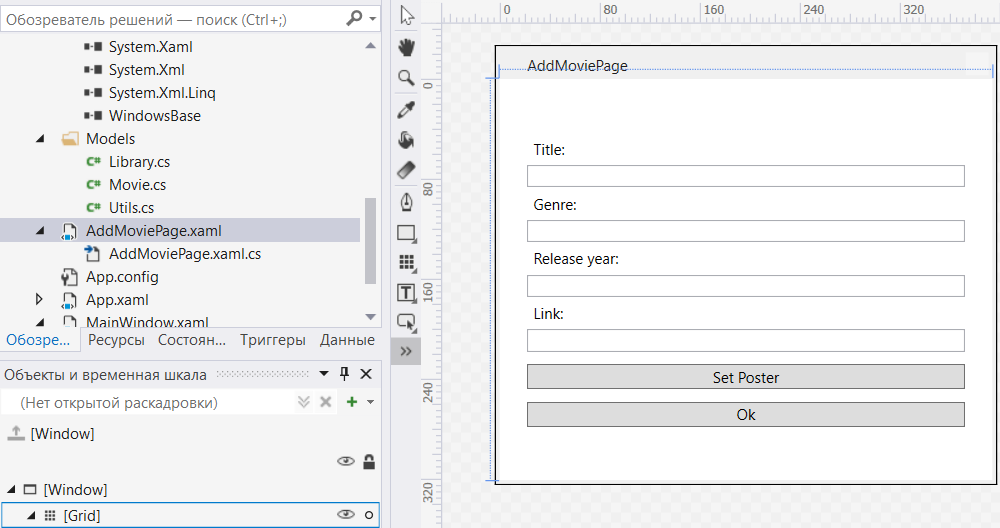
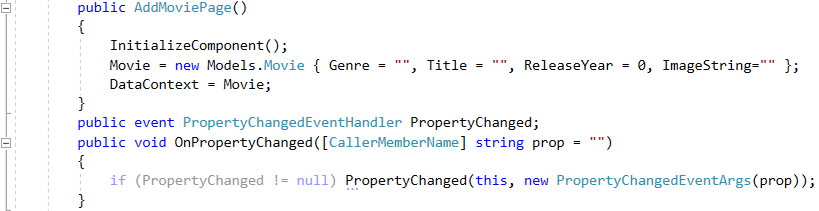


Рисунок 3.6. Побудова інтерфейсу.

*Джерело: побудоване автором(знімок екрану)*

**AddMoviePage.xaml.cs**

Коли дане вікно ініціалізується, в ньому створюється приватне поле кіно і публічне співвідносно, і йому задається те, щоб не було пустих даних передається фільм з такими даними.(рис.3.7.)

****

**Рисунок 3.7. Передача даних.**

*Джерело: побудоване автором ( знімок екрану)*

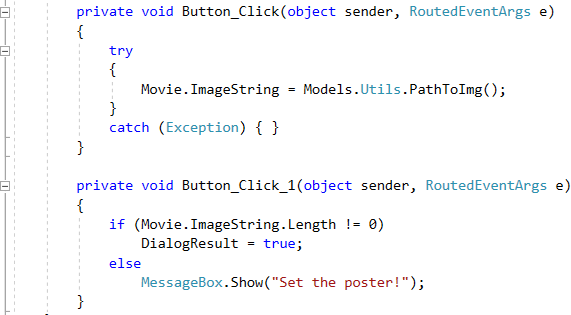
*\*

Button\_Click

Це виклик методу завдання рядка з постером із класу Utils.cs який преобразує постер в набір символів, для його збереження.(рис.3.8)

Button\_Click\_1

Даний метод передає інформацію. інтерфейсу, що можна закрити діалогове вікно, і те що всі дані завантажені успішно.(рис.3.8.)



**Рисунок 3.8. Методи.**

*Джерело: побудоване автором ( знімок екрану)*

В MainWindow.xaml створюється початковий екран програмного додатку (рис.3.9.), де завдяки сітці прив’язується розташування інформації, кнопок, розміру, і дій. Також створюється загальний вид форма, колір. Вся задачність і дії прописуються вже загально в класі MainWindow.xaml.cs,(рис.3.10) так само є публічні і приватні бібліотеки.

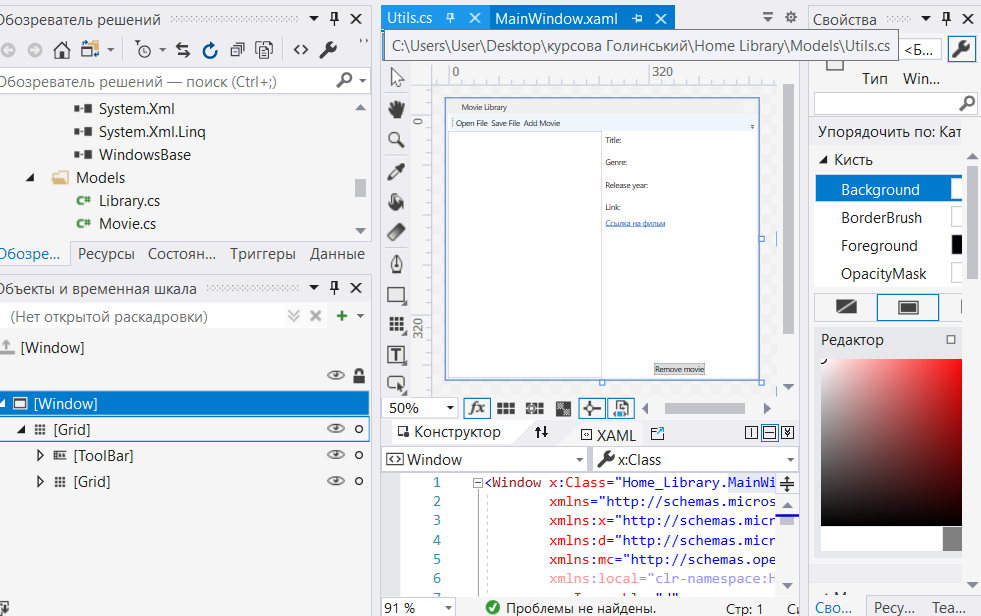


Рисунок 3.9. Розробка інтерфейсу.

*Джерело: побудоване автором ( знімок екрану)*



Рисунок 3.10. Прив’язка клавіш.

*Джерело: побудоване автором ( знімок екрану)*

## **Висновок до розділу 3**

WPF App (.NET Framework) середовище інтерфейсу користувача, яке не залежить, від роздільної здатності та використовує механізм візуалізації на основі векторів. WPF існує як підмножина типів .NET яка зазвичай знаходиться в просторі імен.

# **ВИСНОВОК**

У період пандемії, коли природа землі почала відновлюватись, «вірус століття» позитивно вплинув, на екологічну ситуацію на всій планеті. Лише в Китаї де рівень забрудненості повітря вважався найвищим викиди парникових газів зменшились на 25%, а оксиду азоту на 37%. А в Венецію повернулись дельфіни. Це все відбулось тому, що всі сиділи дома, адже кількість летальних випадків постійно збільшувалась, і в цей період, ще коли світ не адаптувався, людям потрібно було чимось себе розважити. І саме в цей період великої поулярності набули фільиотеки, та онлайн кінотеатри, і навіть зараз вони стають ще більш популярними, про що тільки говорить статистика підключень Netflix. Саме тому мене так сильно захопила дана тематика створити щось своє в даній галузі. Фільмотека мого варіанту відрізняється від тих що можна знайти в мережі Інтернет. В даному програмному додатку зібрано те що на мою думку потрібно знати про фільм щоб його подивитись, і одразу ж посилання, на WEB-сторінку, де на мою думку його найзручніше переглянути, і дізнатись більше інформації.

# **СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ**

1. Художня Культура <https://sites.google.com/site/hudoznakultura15/mistectvo-ak-fenomen-kulturi-genezis-mistectva-formuvanna-jogo-vidovoie-strukturi-ponatta-pro-aktualnij-vid-mistectva/kinematograf-1>
2. Монополія на кіно: кінопрокатники та державні органи обговорили розвиток кінопрокатного ринку. <https://detector.media/rinok/article/138158/2018-06-04-monopoliya-na-kino-kinoprokatnyky-ta-derzhavni-organy-obgovoryly-rozvytok-kinoprokatnogo-rynku/>
3. Розвиток та фінансування кіновидавництва в країнах з розвиненою кіноіндустрією. Калабихина Є.А. МГУ. <https://archive.econ.msu.ru/ext/lib/Category/x0d/x0f/3343/file/5_Kalabikhina.pdf>
4. Топ-10 онлайн-кінотеатрів по популярності в соц. медіа. <https://br-analytics.ru/blog/top-10-online-cinema/>
5. 23 Дивовижні Статистика в Інтернеті і соціальних медіа в 2021 році. <https://uk.wizcase.com/blog/%D0%B4%D0%B8%D0%B2%D0%BE%D0%B2%D0%B8%D0%B6%D0%BD%D1%96-%D1%81%D1%82%D0%B0%D1%82%D0%B8%D1%81%D1%82%D0%B8%D0%BA%D0%B8-%D0%B2-%D1%96%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82%D1%96/>
6. Стратегія забезпечення кібернетичної безпеки України. <https://niss.gov.ua/sites/default/files/2013-11/kiberstrateg.pdf>
7. "Кібербезпека" - спеціальність майбутнього стає сьогоденням! <https://nubip.edu.ua/node/55475>
8. Верховна Рада України. <https://zakon.rada.gov.ua/laws/show/2163-19#Text>.
9. Об’єктно-орієнтоване програмування. .<http://programming.in.ua/programming/basisprogramming/25-oop.html>
10. Інформатика. <https://sites.google.com/site/informatikaforeve/programuvanna/mova-programuvanna>.
11. Об’єктно-орієнтоване програмування. <http://programming.in.ua/programming/basisprogramming/25-oop.html>.
12. 5 найпопулярніших мов програмування. <https://yolo.if.ua/blog/view/5-naipopuliarnishikh-mov-programuvannia>.
13. Програмування по-українськи. <http://programming.in.ua/programming/c-language.html>.
14. Введення в C#.<https://programm.top/c-sharp/tutorial/introduction/>
15. . Tadviser. <https://www.tadviser.ru/index.php/>.
16. Розробка мобільного додатку від А до Я. <https://dan-it.com.ua/uk/rozrobka-mobilnih-dodatkiv-vid-a-do-ja-povnij-gajd/>.
17. Розробка комп’ютерних програм. <https://osvita.ua/proforientation/profession/74835/>.
18. C#.

<https://wikies.fandom.com/wiki/C_Sharp>.

1. Microsoft Visual Studio. <https://uk.wikipedia.org/wiki/Microsoft_Visual_Studio>.
2. METANIT.COM <https://metanit.com/sharp/wpf/2.php>

**ДОДАТКИ**

Додаток А

**Код класу Library.cs**

using Microsoft.Win32;

using System;

using System.Collections.ObjectModel;

using System.ComponentModel;

using System.Runtime.CompilerServices;

using System.Xml.Linq;

namespace Home\_Library.Models

{

public class Library : INotifyPropertyChanged

{

public Library()

{

Movies = new ObservableCollection<Movie>();

}

private ObservableCollection<Movie> movies;

public ObservableCollection<Movie> Movies

{

get => movies;

set

{

movies = value;

OnPropertyChanged(nameof(Movies));

}

}

public void SaveToXML()

{

try

{

XDocument xDoc = new XDocument();

XElement Library = new XElement("Library");

foreach(var movie in Movies)

{

Console.WriteLine("2");

XElement Movie = new XElement("Movie");

Movie.Add(

new XAttribute("Title", movie.Title),

new XAttribute("ReleaseYear", movie.ReleaseYear),

new XAttribute("ImageString", movie.ImageString),

new XAttribute("Genre", movie.Genre),

new XAttribute("Link", movie.Link)

);

Library.Add(Movie);

}

xDoc.Add(Library);

SaveFileDialog saveFileDialog = new SaveFileDialog() {

Filter = "XML | \*.xml",

Title = "Сохранение новой библиотеки",

FileName = "Lib.xml"

};

if(saveFileDialog.ShowDialog() == true)

{

xDoc.Save(saveFileDialog.FileName);

}

}

catch (Exception) { }

}

public void LoadFromXML()

{

try

{

OpenFileDialog openFileDialog = new OpenFileDialog()

{

Filter = "XML | \*.xml",

Title = "Чтение библиотеки"

};

openFileDialog.ShowDialog();

if (openFileDialog.FileName != "")

{

var xDoc = XDocument.Load(openFileDialog.FileName);

var tempLib = new ObservableCollection<Movie>();

foreach(var movie in xDoc.Element("Library").Elements("Movie"))

{

var tempMovie = new Movie

{

Link = movie.Attribute("Link").Value,

Genre = movie.Attribute("Genre").Value,

ReleaseYear = Int32.Parse(movie.Attribute("ReleaseYear").Value),

ImageString = movie.Attribute("ImageString").Value,

Title = movie.Attribute("Title").Value

};

tempLib.Add(tempMovie);

}

Movies = tempLib;

}

}

catch (Exception) { }

}

public event PropertyChangedEventHandler PropertyChanged;

public void OnPropertyChanged([CallerMemberName] string prop = "")

{

if (PropertyChanged != null) PropertyChanged(this, new PropertyChangedEventArgs(prop));

}

}

}

Додаток Б

**Код класу Movie.cs**

using System;

using System.ComponentModel;

using System.Drawing;

using System.Drawing.Imaging;

using System.IO;

using System.Runtime.CompilerServices;

using System.Windows.Media;

using System.Windows.Media.Imaging;

namespace Home\_Library.Models

{

public class Movie : INotifyPropertyChanged

{

private string \_title, \_genre, \_link;

private string \_img;

private int \_releaseYear;

public string Title

{

get => \_title;

set

{

try

{

\_title = value;

}

catch (Exception)

{

\_title = "Вы сломали Title, поздравляю";

}

OnPropertyChanged(nameof(Title));

}

}

public string Genre

{

get => \_genre;

set

{

try

{

\_genre = value;

}

catch (Exception)

{

\_genre = "Undefined";

}

OnPropertyChanged(nameof(Genre));

}

}

public string Link

{

get => \_link;

set

{

\_link = value;

OnPropertyChanged(nameof(Link));

}

}

public string ImageString

{

get => \_img;

set

{

\_img = value;

OnPropertyChanged(nameof(Poster));

}

}

public ImageSource Poster

{

get

{

byte[] NewBytes = Convert.FromBase64String(ImageString);

MemoryStream ms = new MemoryStream(NewBytes);

Image NewImage = Image.FromStream(ms);

using (var ms2 = new MemoryStream())

{

NewImage.Save(ms2, ImageFormat.Bmp);

ms2.Seek(0, SeekOrigin.Begin);

var bitmapImg = new BitmapImage();

bitmapImg.BeginInit();

bitmapImg.CacheOption = BitmapCacheOption.OnLoad;

bitmapImg.StreamSource = ms2;

bitmapImg.EndInit();

return bitmapImg;

}

}

}

public int ReleaseYear

{

get => \_releaseYear;

set

{

try

{

\_releaseYear = Convert.ToInt32(value);

}

catch (Exception)

{

\_releaseYear = 5005;

}

OnPropertyChanged(nameof(ReleaseYear));

}

}

public event PropertyChangedEventHandler PropertyChanged;

public void OnPropertyChanged([CallerMemberName] string prop = "")

{

if (PropertyChanged != null)

PropertyChanged(this, new PropertyChangedEventArgs(prop));

}

}

}

Додаток В

**Код класу Utils.cs**

using Microsoft.Win32;

using System;

using System.Drawing;

using System.Drawing.Imaging;

using System.IO;

namespace Home\_Library.Models

{

public static class Utils

{

public static string PathToImg()

{

OpenFileDialog ofd = new OpenFileDialog() { Filter = "IMGs | \*.jpg; \*jpeg;", Title="Open Image" };

if(ofd.ShowDialog() == true && ofd.FileName != "")

{

MemoryStream ms = new MemoryStream();

Image.FromFile(ofd.FileName).Save(ms, ImageFormat.Jpeg);

byte[] bytes = ms.ToArray();

return Convert.ToBase64String(bytes);

}

return "";

}

}

}

Додаток Г

Код класу AddMoviePage.xaml.cs

using System;

using System.ComponentModel;

using System.Runtime.CompilerServices;

using System.Windows;

namespace Home\_Library

{

/// <summary>

/// Логика взаимодействия для AddMoviePage.xaml

/// </summary>

public partial class AddMoviePage : Window, INotifyPropertyChanged

{

private Models.Movie movie;

public Models.Movie Movie

{

get => movie;

set

{

movie = value;

OnPropertyChanged(nameof(Movie));

}

}

public AddMoviePage()

{

InitializeComponent();

Movie = new Models.Movie { Genre = "", Title = "", ReleaseYear = 0, ImageString="" };

DataContext = Movie;

}

public event PropertyChangedEventHandler PropertyChanged;

public void OnPropertyChanged([CallerMemberName] string prop = "")

{

if (PropertyChanged != null) PropertyChanged(this, new PropertyChangedEventArgs(prop));

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

try

{

Movie.ImageString = Models.Utils.PathToImg();

}

catch (Exception) { }

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

if (Movie.ImageString.Length != 0)

DialogResult = true;

else

MessageBox.Show("Set the poster!");

}

}

}

Додаток Д

Код класу MainWindow.xaml.cs

using System;

using System.Collections.ObjectModel;

using System.ComponentModel;

using System.Diagnostics;

using System.Runtime.CompilerServices;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Documents;

namespace Home\_Library

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window, INotifyPropertyChanged

{

private Models.Library \_library;

private Models.Movie \_selected;

public Models.Library Library

{

get => \_library;

set

{

\_library = value;

OnPropertyChanged(nameof(Library));

}

}

public Models.Movie SelectedMovie

{

get => \_selected;

set

{

\_selected = value;

OnPropertyChanged(nameof(SelectedMovie));

}

}

public MainWindow()

{

InitializeComponent();

Library = new Models.Library() { Movies = new ObservableCollection<Models.Movie>() };

DataContext = this;

}

private void ListBox\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

if (lstbox.SelectedItem == null) return;

SelectedMovie = lstbox.SelectedItem as Models.Movie;

}

private void Remove\_Movie(object sender, EventArgs e)

{

if (lstbox.SelectedItem == null) return;

Library.Movies.Remove(lstbox.SelectedItem as Models.Movie);

OnPropertyChanged(nameof(SelectedMovie));

}

private void Open\_File(object sender, EventArgs e)

{

Library.LoadFromXML();

}

private void Save\_File(object sender, EventArgs e)

{

Library.SaveToXML();

}

private void Add\_Movie(object sender, EventArgs e)

{

var window = new AddMoviePage();

if(window.ShowDialog() == true) {

Library.Movies.Add(window.Movie);

}

}

public event PropertyChangedEventHandler PropertyChanged;

public void OnPropertyChanged([CallerMemberName] string prop = "")

{

if (PropertyChanged != null) PropertyChanged(this, new PropertyChangedEventArgs(prop));

}

private void Hyperlink\_Click(object sender, RoutedEventArgs e)

{

Process.Start((sender as Hyperlink).NavigateUri.ToString());

}

}

}

**Київський національний торговельно-економічний університет**

**Рецензія на курсову роботу (проект) і результат захисту**

Студента **\_\_** Голинський Андрій Миколайович

(прізвище, ім’я та по батькові)

\_\_\_2\_\_\_ курсу \_\_12\_\_\_\_ групи \_\_\_\_\_\_\_\_ФІТ\_\_\_\_\_\_\_факультету

Курсова робота (проект) з \_\_«Об’єктно-орієнтоване програмування»\_\_

(назва навчальної дисципліни)

Тема Розробка програмного додатку розвитку домашня фільмотека мовою програмування C#

Реєстраційний № \_\_\_\_, дата одержання 11.05.2021р.

Науковий керівник\_\_\_\_\_асистент кафедри інженерії програмного забезпечення та кібербезпеки Гнатченко Дмитро Дмитрович. (вчене звання, прізвище, ініціали)

**Зміст рецензії**

У курсовій роботі студента Голинського А.М. на тему «Розробка програмного додатку домашня фільмотека мовою програмування C#» зазначено актуальність дослідження, мету, об’єкт, предмет та задачі дослідження.

У теоретичній частини курсової роботи студент розглянув питання пов’язані з попитом на даний продукт та розробкою програмного додатку. Автором детально описано та охарактеризовано додаток «Домашня фільмотека». Також, для реалізації інтерфейсу, автор розробив подання для відповідних форм програмного модулю інтерфейсу. У третьому розділі подано детальний опис розробки додатку у середовищі Visual Studio. В цілому курсова робота студента Голинського А.М. на тему «Розробка програмного додатку домашня фільмотека мовою програмування C#» оформлена у відповідності до Вимог, може бути допущена до захисту та заслуговує на позитивну оцінку.

Допущено до захисту “11”\_\_\_\_05\_\_\_\_\_ 2021 р.

Захист планується о \_10,00\_\_\_\_\_ “\_\_14\_\_\_”\_\_\_\_\_05\_\_\_ 2021 р.

(час)

кафедра інженерії програмного забезпечення та кібербезпеки

(місце роботи комісії)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис наукового керівника)

Курсова робота захищена “\_14\_”\_\_\_05\_\_\_\_\_ 2021 р.

з оцінкою \_75\_\_\_\_ «» \_\_\_\_\_\_балів\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(за шкалою КНТЕУ, національною шкалою та шкалою ЄКТС)

**Комісія:**

1. \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_Бебешко Б.Т.\_\_\_\_\_\_\_\_

(підпис) (прізвище, ініціали)

1. \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_ Жирова Т.О.\_\_\_\_\_\_\_\_

(підпис) (прізвище, ініціали)

1. \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Хорольська К.В.\_\_\_\_

(підпис) (прізвище, ініціали)