Міністерство освіти і науки України

Харківський національний університет імені В.Н. Каразіна

Факультет комп’ютерних наук

**Пояснювальна записка**

до курсової роботи з дисципліни «Об’єктно-орієнтоване програмування»

на тему: Бібліотека та управління реєстром видань в ній

Виконав студент 2 курсу

групи КС-21

Клочко Андрій Володимирович

Перевірив:

к.т.н., доц. Нагорний К.А.

Харків – 2020

**РЕФЕРАТ**

Звіт про виконання курсової роботи: кількість сторінок с., кількість рисунків рис.

Ключові слова: ООП, JAVA, SOLID, IDE, CSV, ПРЕДМЕТНА ОБЛАСТЬ.

Пропонується розробити java-додаток для автоматизації процесу управління реєстром видань в бібліотеці. Спочатку треба визначити предметну область, проаналізувати вимоги, створити модель програмного продукту, написати програмну реалізацію.

Даний застосунок дозволить керувати реєстром публікацій в бібліотеці, а також реєстром користувачів за допомогою CLI. Модель програмного продукту буде побудована за допомогою UML-діаграми класів.

ЗМІСТ

[СПИСОК ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ 4](#_Toc59291144)

[ВСТУП 5](#_Toc59291145)

[РОЗДІЛ 1 ПОСТАНОВА ЗАДАЧІ 6](#_Toc59291146)

[1.1 Область визначення прикладного рішення 6](#_Toc59291147)

[1.2 Основні можливості системи 6](#_Toc59291148)

[РОЗДІЛ 2 ПРОЕКТУВАННЯ 8](#_Toc59291149)

[2.1 Опис предметної області 8](#_Toc59291150)

[2.2 Класи предметної області 8](#_Toc59291151)

[2.3 UML-діаграма класів 9](#_Toc59291152)

[РОЗДІЛ 3 РЕЗУЛЬТАТИ РОБОТИ 13](#_Toc59291153)

[3.1 Тестування 13](#_Toc59291154)

[3.2 Інструкція для користувача 16](#_Toc59291155)

[ВИСНОВОК 18](#_Toc59291156)

[ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ 19](#_Toc59291157)

# СПИСОК ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

ООП – об’єктно-орієнтовне програмування.

ПрО (предметна область) – частина реального світу, що розглядається в межах певного контексту.

UML (англ. Unified Modeling Language) – уніфікована мова моделювання.

JDK (англ. Java Development Kit) – безкоштовний розповсюджуваний Oracle комплект розробника застосунків на мові Java.

SOLID – абревіатура п'яти основних принципів проектування в об'єктно-орієнтованому програмуванні – Single responsibility, Open-closed, Liskov substitution, Interface segregation і Dependency inversion. У перекладі на українську: принципи єдиної відповідальності, відкритості / закритості, підстановки Барбари Лісков, поділу інтерфейсу і інверсії залежностей).

IDE (Integrated Development Environment) – система програмних засобів, яка використовується програмістами для розробки програмного забезпечення.

CSV – текстовий формат, призначений для представлення табличних даних. Рядок таблиці відповідає рядку тексту, яка містить одне або кілька полів, розділених комами.

# ВСТУП

У ході роботи буде розроблений застосунок, який дозволить користувачам бібліотеки взаємодіяти із реєстром видінь, що зберігаються у бібліотеці.

При проектуванні програмного забезпечення, з використанням парадигми ООП, принципів проектування SOLID, моделі будуть побудовані за допомогою діаграм мови UML. Для побудування UML-діаграм класів та Use Case-діаграм буде використовуватись онлайн ресурс creately [1].

У ході розробки буде використовуватись мова програмування java, а також IDE, а саме Intellij IDEA від компанії JetBrains [2].

# РОЗДІЛ 1 ПОСТАНОВА ЗАДАЧІ

## 1.1 Область визначення прикладного рішення

**Мета**: Розробка прототипу програмної системи на базі основних принципів об’єктно-орієнтованого дизайну та сучасних практик програмування із використанням об’єктно-орієнтованої мови програмування Java.

**Предметна область:** бібліотека.

**Призначення програмної системи:** управління реєстром видань у бібліотеці.

**Задання**: Розробити прототип програмної системи (ПС) управління виданнями у бібліотеці, яка надає можливість користувачу маніпулювати бібліотечним реєстром. Інтерфейс користувача – консоль, до якої має бути виведений список усіх можливих команд. ПС має працювати до введення команди виходу із системи. Потрібно реалізувати збереження даних таких як публікації та користувачі, в файл у форматі CSV (coma separated values). Описати відповідну структуру класів розробленого програмного рішення в нотації UML у термінах діаграми класів (class diagram).

## 1.2 Основні можливості системи

До переліку можливостей системи входять такі обов’язкові пункти:

1. Додати видання до реєстру
2. Видалити видання з реєстру
3. Отримати список усіх видань, що містяться у реєстрі
4. Отримати список усіх видань за мовою якою вони надруковані
5. Видати обране видання користувачеві бібліотеки, при цьому видання не може бути видалене із реєстру до моменту повернення до бібліотеки
6. Зберегти реєстр до сховища (текстовий файл формату CSV)
7. Отримати реєстр зі сховища (текстового файлу формату CSV)

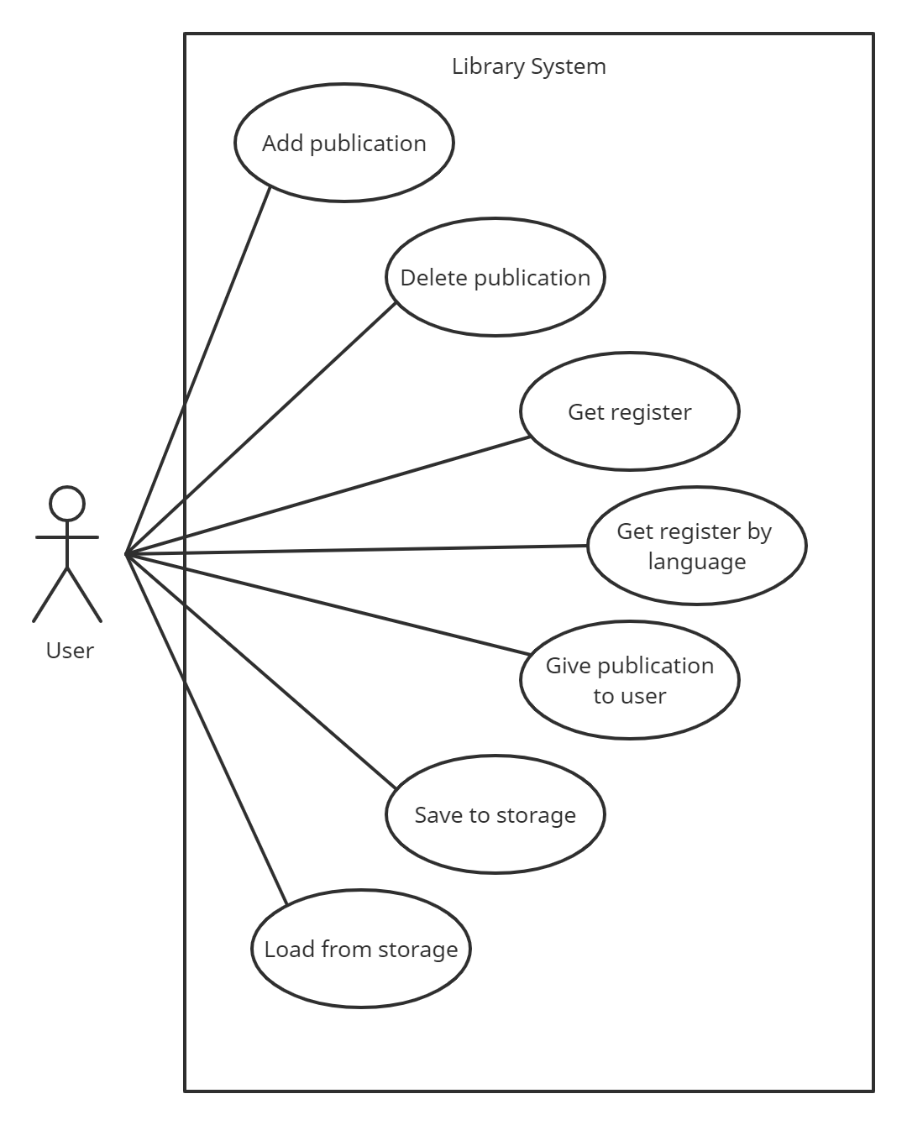


Рисунок 1.1 – Use Case-діаграма

# РОЗДІЛ 2 ПРОЕКТУВАННЯ

## 2.1 Опис предметної області

До основних сутностей предметної області (ПрО) входять такі сутності:

1. Бібліотека. Має назву, адресу, мастить реєстр видань
2. Книга – неперіодичне видання, яке має назву, автора, кількість сторінок та мову друку. У книги є видавець, який видає книгу у відповідному році.
3. Журнал – періодичне видання, яке має назву, автора статті, кількість статей та загальну кількість сторінок та мову друку. У журналу є видавець, який видає журнал у відповідному році.
4. Автор – людина, яка створила книгу або приймала участь у створенні журналу.
5. Користувач бібліотеки – має ім’я та прізвище, рік народження та номер бібліотечного квитка

## 2.2 Класи предметної області

Library – головний клас, має в собі реєстр публікацій та користувачів.

Publication – абстрактний клас, предок класів Book та Magazine.

Book, Magazine – різновид публікації, являються нащадками абстрактного класу Publication.

PublicationType – клас перерахувань типів публікацій.

PublicationRegister – реєстр публікацій, список всіх публікацій які знаходяться або знаходились в бібліотеці.

User – клас користувача системи.

UserRegister – реєстр користувачів, список всіх користувачів бібліотеки.

FileWriter – клас запису даних в файл.

OutputDataFormer – клас формування даних для запису у вихідний файл.

FileReader – клас зчитування даних з файлу.

InputDataParser – клас парсер даних з вхідного файлу.

## 2.3 UML-діаграма класів

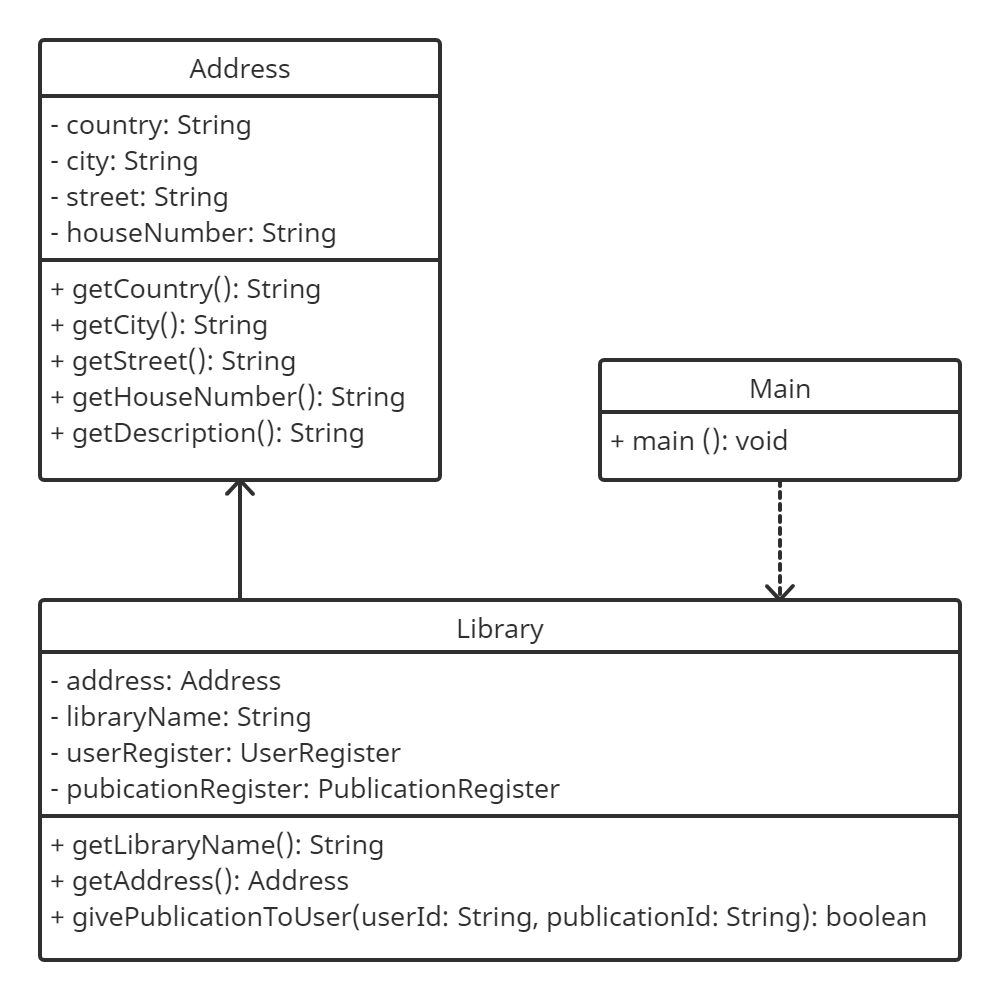


Рисунок 2.1 – UML-діаграма класу Library

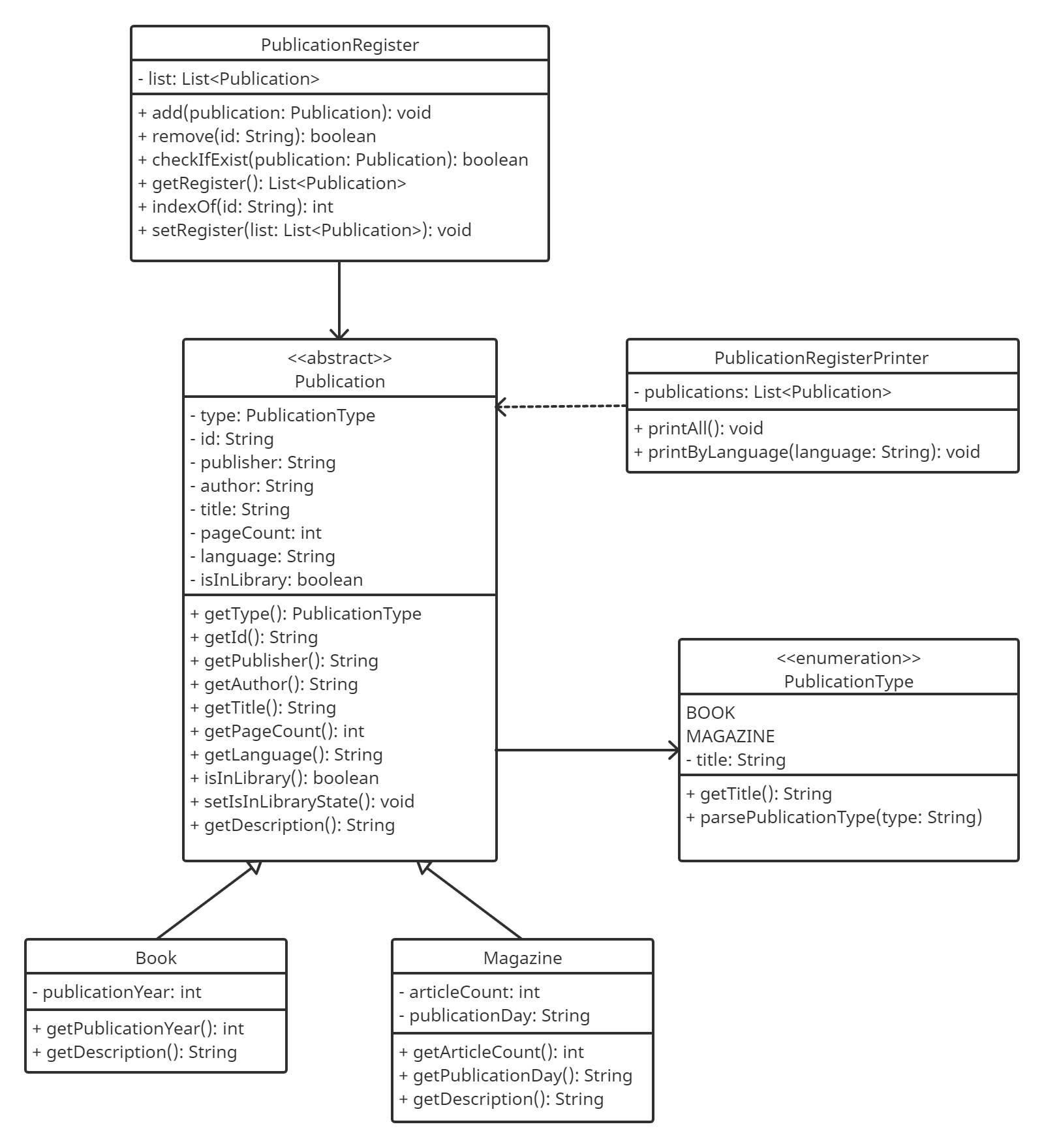


Рисунок 2.2 – UML-діаграма класу PublicationRegister

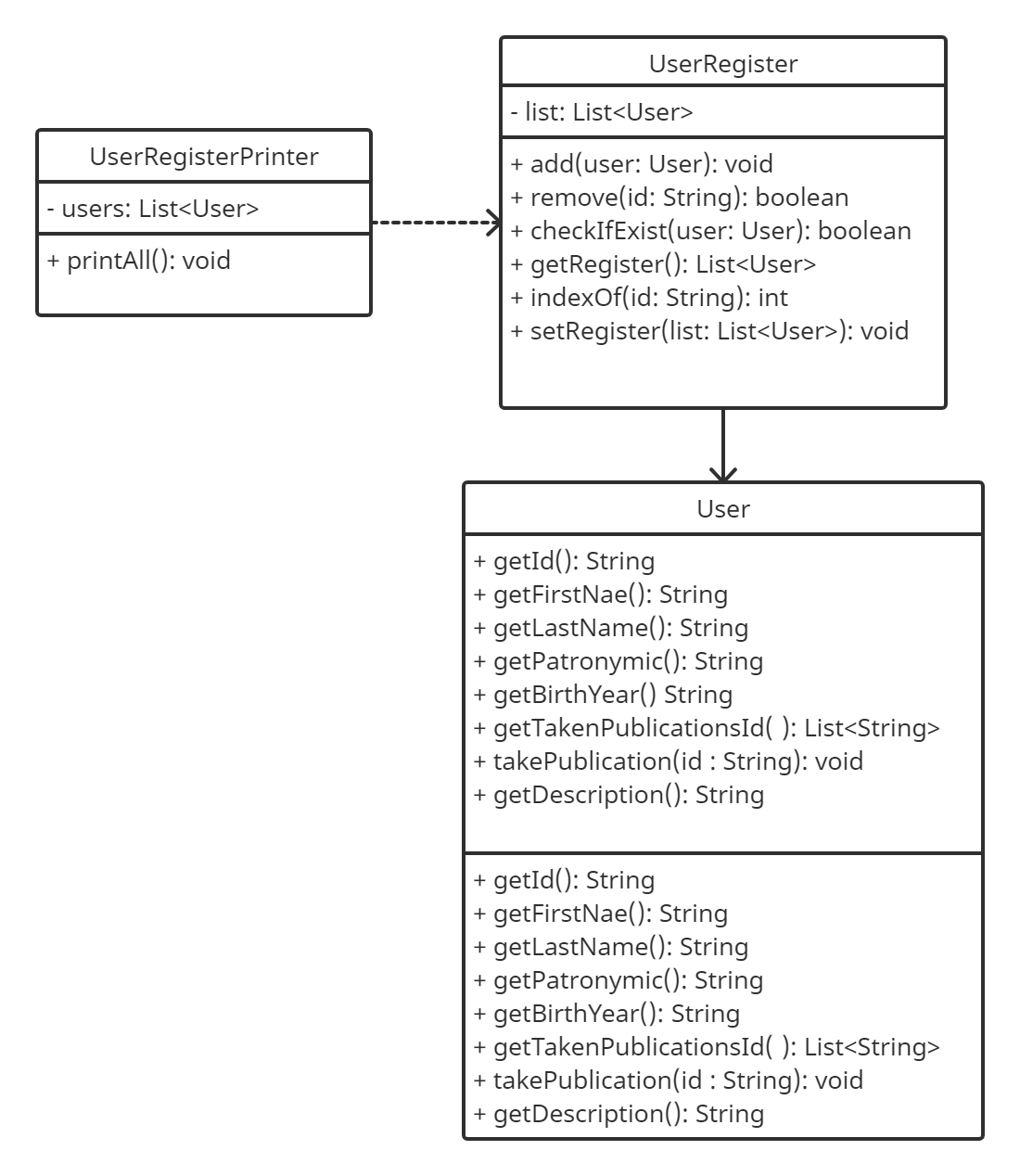


Рисунок 2.3 – UML-діаграма класу UserRegister

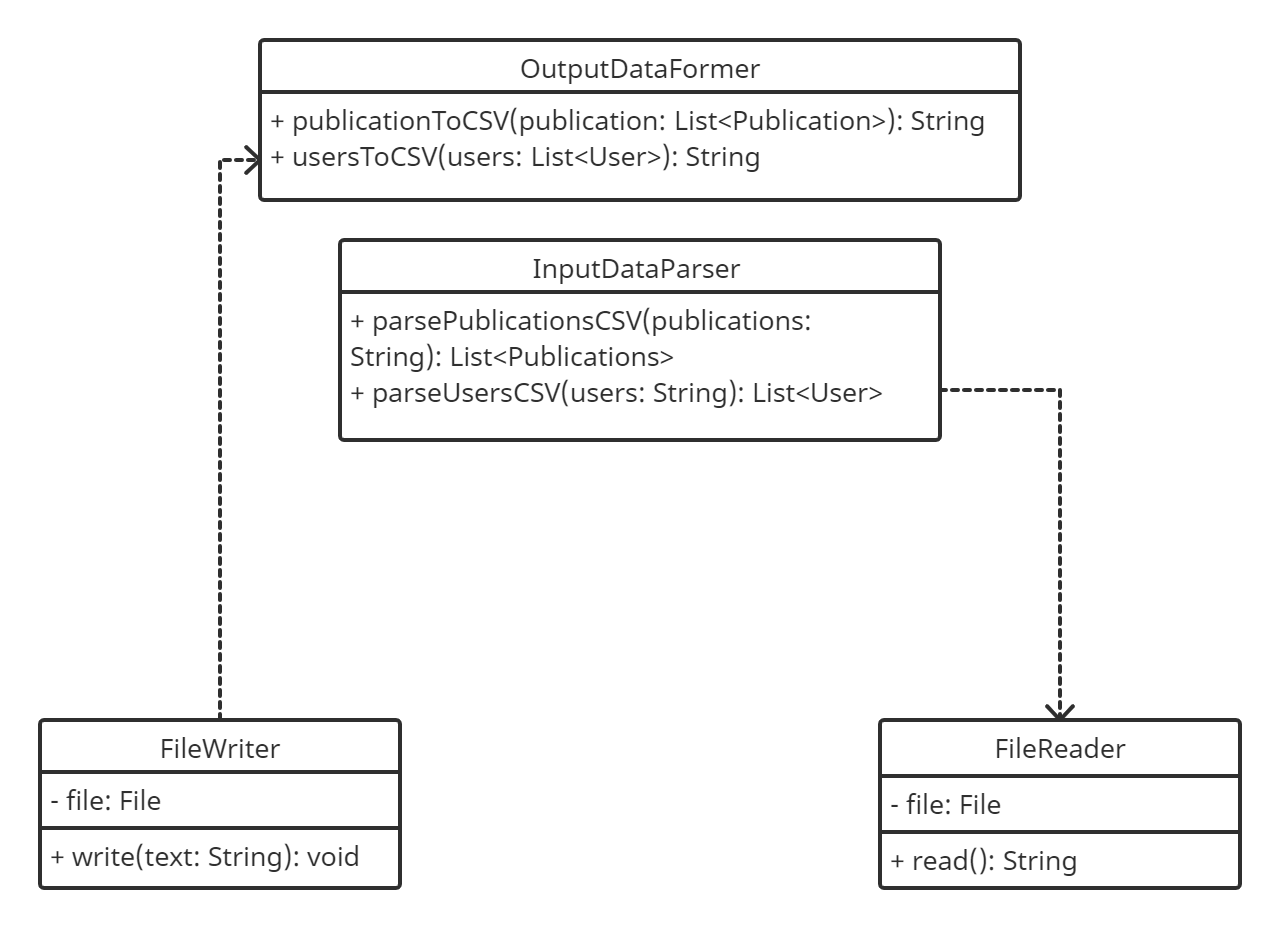


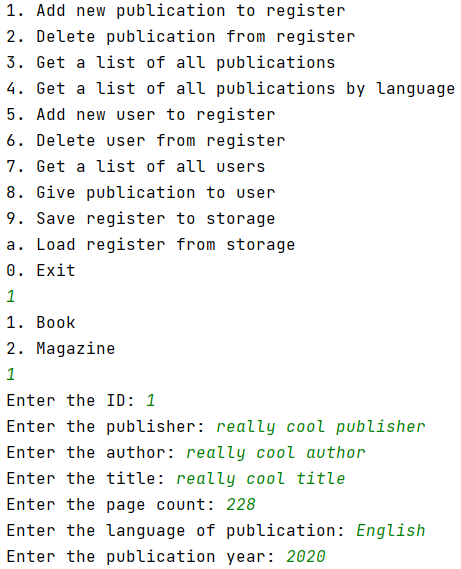
Рисунок 2.3 – UML-діаграма класів формування та парсингу даних з файлів

Повну UML-діаграму класів можна подивитися за посиланням 2 в переліку посилань [2].

# РОЗДІЛ 3 РЕЗУЛЬТАТИ РОБОТИ

## 3.1 Тестування

Одним із найважливіших етапів в розробці програмного забезпечення є тестування продукту. В ході тестування помилок виявлено не було. Результати тестування приведені на рисунках 3.1 – 3.9.



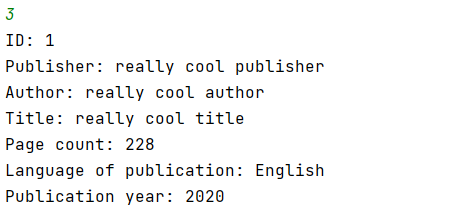
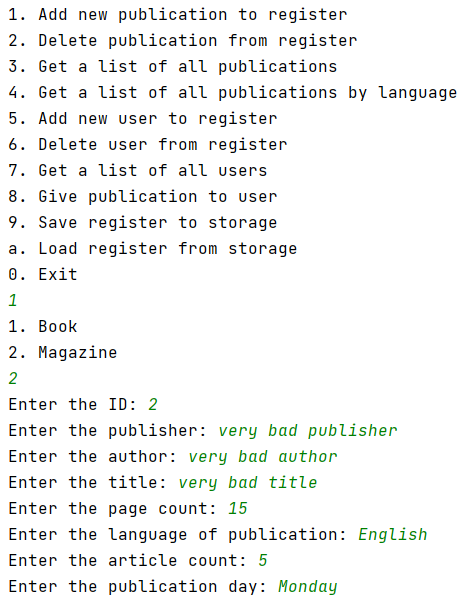


Рисунок 3.1 – Додавання книги в реєстр



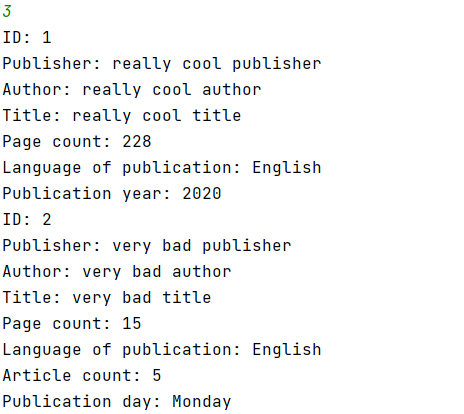
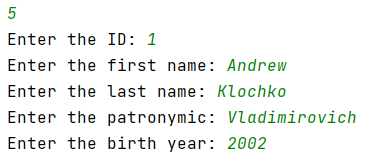


Рисунок 3.2 – Додавання журналу в реєстр



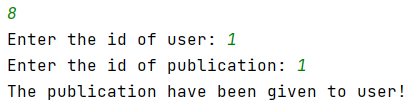


Рисунок 3.3 – Створення нового користувача та видача йому книги

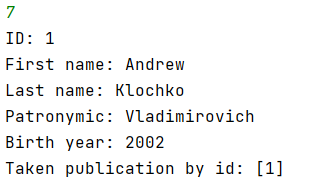


Рисунок 3.4 – Відображення реєстру користувачів бібліотеки

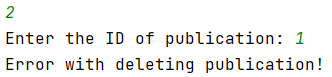


Рисунок 3.5 – Спроба видалення публікації, яка знаходиться не в бібліотеці



Рисунок 3.6 – Збереження реєстру користувачів до файлу users.csv



Рисунок 3.7 – Збереження реєстру публікацій до файлу publications.csv

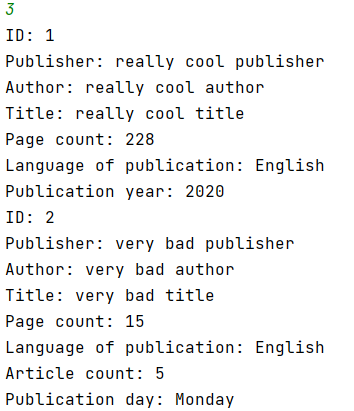


Рисунок 3.8 – Відображення всіх публікацій після завантаження з файлу

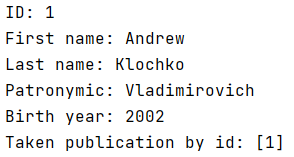


Рисунок 3.9 – Відображення всіх користувачів після завантаження з файлу

## 3.2 Інструкція для користувача

В CLI даної програми присутні такі пункт меню:

1. Add new publication to register
2. Delete publication from register
3. Get a list of all publications
4. Get a list of all publications by language
5. Add new user to register
6. Delete user from register
7. Get a list of all users
8. Give publication to user
9. Save register to storage
10. Load register from storage
11. Exit

Add new publication to register – Додавання нової публікації до реєстру. Не дозволяється додавати публікації з однаковими ID.

Delete publication from register – Видалення існуючої публікацію з регістру. Видалення відбувається по ID публікації. При невірному ID ніяка публікація не видалиться.

Get a list of all publications – Відображення всіх публікацій що містяться в регістрі.

Get a list of all publications by language – Відображення всіх публікацій що містяться в регістрі відфільтровані по мові друку.

Add new user to register – Додавання нового користувача до реєстру. Не дозволяється додавати користувачів з однаковими ID.

Delete user from register – Видалення існуючого користувача з регістру. Видалення відбувається по ID користувача. При невірному ID ніякий користувач не видалиться.

Get a list of all users – Відображення всіх користувачів що містяться в регістрі.

Give publication to user – Видача публікації користувачу. Видача відбувається по ID публікації та ID користувача. При невірних ID видача не відбудеться.

Save register to storage – Збереження реєстру до файлу в форматі .csv.

Load register from storage – Загрузка реєстру з файлу формату .csv.

Exit – Вихід з програми. При виході з програми, не відбувається автозбереження даних до реєстру.

# ВИСНОВОК

В ході виконання курсової роботи було розроблений прототип системи, а саме бібліотеки. Програма містить форми, які дають змогу користувачеві додавати, видаляти, та отримувати необхідну інформацію. Для збереження даних було створено файл, який має розширення .csv, з якого ми отримували та записували інформацію в відповідні файли.

# ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. // creately. URL: https://creately.com/. Дата звернення: 16.12.2020
2. // jetbrains. URL: https://www.jetbrains.com/. Дата звернення: 16.12.2020
3. UML-діаграма класів бібліотеки. URL: https://app.creately.com/diagram/bWOikmTkZyj/edit. Дата звернення: 16.12.2020