

Прикарпатський національний університет імені Василя Стефаника

Кафедра інформаційних технологій
(повна назва кафедри)

КУРСОВИЙ ПРОЄКТ

з дисципліни
OK19 Об'єктно-орієнтоване програмування
(шифр та назва навчальної дисципліни)

на тему: Розробка програми оброблення даних з файлів та занесення даних у файли
на тему "Книги"

Здобувачки вищої освіти
другого року навчання, групи ПЗ-24
освітньої програми
Інженерія програмного забезпечення
спеціальності
121 Інженерія програмного забезпечення
першого (бакалаврського) рівня вищої освіти

Анни ШКУРАТ
(Власне ім'я та ПРІЗВИЩЕ здобувача вищої освіти)

Керівник професор кафедри інформаційних
технологій
(посада, вчене звання, науковий ступінь)

Микола КУЗЬ
(Власне ім'я та ПРІЗВИЩЕ)

Національна шкала: _____
Університетська шкала: _____
Оцінка ECTS: _____

Члени комісії: _____ Микола КУЗЬ
(підпис) (Власне ім'я та ПРІЗВИЩЕ)

_____ Марія ДУТЧАК
(підпис) (Власне ім'я та ПРІЗВИЩЕ)

м. Івано-Франківськ – 2024 рік

ЗАВДАННЯ НА КУРСОВИЙ ПРОЄКТ

Анни ШКУРАТ

(Власне ім'я та ПРІЗВИЩЕ здобувача вищої освіти)

Кафедра інформаційних технологій

Дисципліна ОК19 Об'єктно-орієнтоване програмування

Спеціальність 121 Інженерія програмного забезпечення

Освітня програма Інженерія програмного забезпечення

Рік навчання 2 Група ІПЗ- 24 Семестр 4

Перший (бакалаврський) рівень вищої освіти

1. Тема проєкту: Розробка програми оброблення даних з файлів та занесення даних у файли на тему «Книги»

2. Перелік питань, які підлягають розробці: розробити програму оброблення даних з існуючих файлів та занесення оброблених даних у файли; утворити файл, до якого занести книги ціною, не більше 10.0; утворити файл, до якого занести книги ціною в межах від 10.01 до 19.0; - утворити файл, до якого занести книги ціною, більшою за 19.01; провести сортування у вихідних файлах по двох варіантах – по автору, при рівності – по назві, при рівності – по ціні, та по ціні, при рівності – по автору, при рівності – по назві; для перетворення текстового рядка у масив Strings використати метод split() з класу String; для перетворення текстових змінних у числові використати методи parseDouble() з класу Double та parseInt() з класу Integer.

3. Дата видачі завдання 15.02.2024

4. Термін подачі до захисту 14.06.2024

5. Здобувачка вищої освіти _____
(підпис) (Власне ім'я та ПРІЗВИЩЕ) Анна ШКУРАТ

6. Керівник _____
(підпис) (Власне ім'я та ПРІЗВИЩЕ) Микола КУЗЬ

КАЛЕНДАРНИЙ ПЛАН

Пор. №	Назва етапів підготовки курсового проєкту	Термін виконання	Форма контролю
1.	Затвердження теми роботи	15.02.2024	Затвердження на засіданні кафедри
2.	Узгодження завдання та плану курсової роботи з науковим керівником	21.02.2024	Сформовані завдання та план курсової роботи
3.	Обґрунтування актуальності, формулювання мети, завдання, предмету та об'єкту дослідження роботи	02.03.2024	Узгодження з науковим керівником
4.	Опрацювання джерел з теми роботи	16.03.2024	Підготовка огляду літератури
5.	Підготовка I розділу роботи	13.04.2024	Звіт на наук. семінарі кафедри
6.	Підготовка II розділу роботи	26.04.2024	Звіт на наук. семінарі кафедри
7.	Підготовка III розділу роботи	04.05.2024	Представлення наук. керівнику
8.	Виправлення зауважень. Підготовка вступу і висновків.	11.05.2024	Представлення наук. керівнику
9.	Оформлення роботи згідно вимог.	20.05.2024	Представлення наук. керівнику та нормоконтролеру
10.	Виправлення зауважень керівника та нормоконтролера. Відправка електронних версій на перевірку на плагіат.	29.05.2024	Представлення наук. керівнику.
11.	Подача видрукованої і підписаної роботи (завдання, календарний план, зміст) на кафедру.	14.06.2024	Наявність роботи із усіма необхідної підписами на кафедрі
12.	Оцінювання роботи	19.06.2024	Виставлення оцінки на титульній сторінці роботи і заповнення відомості

Здобувачка вищої освіти

Анна ШКУРАТ

(підпис)

(Власне ім'я та ПРІЗВИЩЕ)

Керівник

Микола КУЗЬ

(підпис)

(Власне ім'я та ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка: 33 сторінки, 12 рисунків, 8 джерел.

Ключові слова: МЕТОДИ, ПОЛЯ, ООП, ФП, ПП, КНИГИ, ІНСТРУМЕНТИ, ПРОГРАМА, РОЗРОБКА, ОБРОБКА, ФАЙЛИ.

Об'єктом розробки є програмне забезпечення для обробки та сортування даних про книги.

Мета роботи – розробка програми оброблення даних з файлів та занесення даних у файли на тему «Книги».

Стислий опис тексту пояснювальної записки:

У першому розділі описано предметну область та її особливості, а також наведено деякі інструменти розробки.

У другому розділі проводиться аналіз методів розв'язання поставленої задачі та засобів розробки програмного забезпечення, складаються вимоги до розроблюваного програмного забезпечення.

У третьому розділі описується вирішення поставленої задачі та демонструється робота розробленої програми.

ЗМІСТ

ВСТУП.....	6
1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІНСТРУМЕНТІВ РОЗРОБКИ.....	7
1.1 Опис предметної області та її особливості.....	7
1.2 Інструменти розробки та їх короткий опис	7
АНАЛІЗ МЕТОДІВ РОЗВ’ЯЗАННЯ ЗАДАЧІ ТА ЗАСОБІВ РОЗРОБКИ ПЗ.	
ВИМОГИ ДО ПЗ	10
2.1 Методи програмування.....	10
2.1.1 Процедурне програмування	10
2.1.2 Функціональне програмування.....	11
2.1.3 Об'єктно-орієнтоване програмування	12
2.2 Інструменти розробки.....	13
2.2.1 Sublime Text	13
2.2.2 IntelliJ IDEA	14
2.2.3 Git та GitHub	14
2.2.4 Stack Overflow та YouTube	14
2.3 Вимоги до розроблюваного програмного забезпечення	15
2 ОПИС ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ ТА ДЕМОНСТРАЦІЯ	
РОБОТИ ПРОГРАМИ	17
2.Опис вирішення задачі.....	17
3.1.1 Клас Book	17
3.1.2 Клас Controller	19
3.1.3 Клас Dispatcher	26
3.2 Демонстрація роботи програми	28
ВИСНОВКИ.....	32
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	33

					КП.ПЗ-421.ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата	<div>Розробка програми оброблення даних з файлів та занесення даних у файли на тему «Книги»</div>		
Розроб.	Анна ШКУРАТ						
Перевір.	Микола КУЗЬ						
Н. Контр.	Микола КУЗЬ						
Затверд.	Ігор ЛАЗАРОВИЧ						
					Літ.	Арк.	Акрушів
					н	5	33
					ПНУ ПЗ-24		

ВСТУП

Людська уява надзвичайно сильний інструмент для просування новітніх технологій, вигадкування кулінарних рецептів та навіть написання книг, під час ознайомлення з якими читачі можуть занурюватися у фантастичний чи цілком реальний світ разом з персонажами улюбленої історії. Зараз видати власну книгу не є великою проблемою, тому люди масовіше стали втілювати власні ідеї на папері та продавати їх.

З популяризацією читання та зростанням кількості літератури виникла потреба у її систематизації для полегшення пошуку. Книжкові каталоги стали невід'ємною частиною бібліотечних систем, електронних платформ для читання та книжкових магазинів. Вони допомагають упорядковувати книги за різними критеріями, такими як автор, жанр, рік видання, ціна тощо.

У контексті книжкових магазинів, каталогізація книг є ключовим елементом ефективного управління асортиментом. Добре організований каталог дозволяє швидко знайти потрібну книгу, отримати інформацію про новинки та популярні видання, а також здійснити покупки без зайвих труднощів. Зважаючи на це, важливо розробити ефективну систему обробки файлів, яка забезпечить якісну каталогізацію книжкових товарів.

Програма, розроблена в рамках курсового проєкту, дозволить ефективно обробляти дані про книги, розподіляючи їх за цінovими категоріями; матиме функцію сортування книг за різними критеріями, що дозволить користувачам легко знаходити потрібну інформацію.

					КП.ІПЗ-421.ІЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІНСТРУМЕНТІВ РОЗРОБКИ

1.1 Опис предметної області та її особливості

Предметна область даного курсового проєкту охоплює розробку програмного забезпечення для обробки зчитаних із файлів даних про книги та запис їх результатів у вихідні файли. Потрібно створити програму, яка дозволить спростити процес каталогізації, сортування та зберігання інформації про книги.

Книжковий ринок сьогодні характеризується широким вибором літератури на будь-який смак. Для зручності купівлі та продажу книг необхідно мати добре організовані каталоги. Вони можуть містити різну інформацію про книги, а саме: автор, назва, жанр, рік видання, ціна абощо. Формування таких каталогів вручну – дуже довгий і трудомісткий процес, який, до того ж, може допускати багато помилок.

Однією з основних особливостей даної предметної області є необхідність роботи з текстовими файлами. Програма повинна зчитувати їх, аналізувати вміст, виконувати необхідні операції з даними та зберігати результати в інших текстових файлах.

Процес обробки даних включає кілька етапів. Спочатку програма зчитує дані з файлів і заносить їх у списки чи інші структури даних, які підходять для цього. Далі здійснюється сортування записів за різними критеріями: за автором чи ціною, наприклад. Це дозволяє організувати дані у зручному для користувача вигляді та забезпечити швидкий доступ до потрібної інформації. Зрештою, результати обробки записуються у вихідні файли і в подальшому можуть використовуватися для створення книжкових каталогів тощо.

1.2 Інструменти розробки та їх короткий опис

Для розробки програми оброблення, фільтрування та сортування даних про книги було обрано мову програмування Java.

					КП.ІПЗ-421.ІЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

Java протягом останнього десятиліття залишається однією з найбільш популярних мов програмування[1]. Особливо сильно вона користується попитом серед великих підприємств, адже їй притаманна масштабованість, яка дуже важлива для розвитку та розростання проектів, та надійна підтримка спільноти. Програми на Java можуть виконуватися на будь-якій існуючій операційній системі[2], де встановлена віртуальна машина Java (JVM). Завдяки цьому досягається незалежність Java програм від платформи виконання.

Java має велику кількість різноманітних бібліотек, які полегшують роботу розробникам. Для даного проекту особливо важливими є бібліотеки для роботи з файлами. Для цього можуть знадобитися класи File, FileReader, BufferedReader, FileWriter, PrintWriter або з пакету java.io[3], які дозволяють зчитувати дані з одних файлів та записувати оброблену інформацію в інші.

Java також має хороші засоби для обробки тексту. Клас String з пакету java.lang та його методи дозволяють зручно маніпулювати рядками: розділяти рядки, порівнювати, додавати один рядок у кінець іншого або зчитувати швидко витягувати необхідну інформацію з тексту та обробляти її в залежності від потреб користувача.

Важливою особливістю Java є те, що вона найкраще пристосована до об'єктно-орієнтованого програмування на відміну від, наприклад, C++. Це дозволяє зручно організовувати програму в сукупність класів, що дозволяє вносити зміни в окремі модулі, що потребують цього, а не у весь проект, та масштабувати програму, створюючи нові класи, які не так сильно впливатимуть на роботу інших класів, на відміну від того ж процедурного програмування.

Деякі мови програмування, такі як C++, змушують програмістів виділяти та звільняти пам'ять вручну. Це забезпечує надійний контроль над ресурсами пам'яті, однак може виявитися дещо незручним та важким інструментом, особливо для початківців, через що можуть виникати помилки та витік пам'яті. Віртуальна машина Java автоматично керує пам'яттю, через що розробники можуть не турбуватися про розподілення ресурсів пам'яті під час написання коду. Пам'ять виділяється в купі JVM щоразу при створенні об'єкта та

					КП.ІІЗ-421.ІЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

звільняється за допомогою збирача сміття (Garbage Collector)[4]. Це особливо важливо для програм, які працюють з великими обсягами даних, як-от програм для обробки інформації про книги.

Одним із важливих інструментів розробки, які використовуються разом з Java, є IDE IntelliJ IDEA від JetBrains. Це середовище розробки має зручний інтерфейс для написання, відладки та тестування коду, а також різноманітну кількість інструментів, які допомагають пришвидшити та покращити процес розробки. Для початківців це середовище розробки може здатися трохи заскладним через велику кількість інструментів, на відміну від Eclipse, однак варто розібратися у функціоналі IntelliJ IDEA – і відмовитися від цього середовища розробки буде неможливо.

Вибір Java, як мови програмування, для розробки програми обробки даних з файлів та занесення їх в інші файли на тему "Книги" обумовлений її стабільністю, великими функціональними можливостями, зокрема ефективними інструментами обробки файлів і текстових даних, які є особливо важливими для розроблюваної програми.

					КП.ІПЗ-421.ІЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

АНАЛІЗ МЕТОДІВ РОЗВ'ЯЗАННЯ ЗАДАЧІ ТА ЗАСОБІВ РОЗРОБКИ ПЗ. ВИМОГИ ДО ПЗ

2.1 Методи програмування

У цьому розділі будуть розглянуті три основні парадигми програмування: процедурне, функціональне та об'єктно-орієнтоване програмування. Після зважувань усіх «за» та «проти» буде обрано метод, який найкраще підходить для виконання поставленого завдання.

2.1.1 Процедурне програмування

Процедурне програмування (ПП) є однією з найстаріших парадигм програмування. Концепція ПП полягає у розділенні програми на підпрограми, які називають процедурами та функціями[5]. У різних мовах програмування терміни “процедури” та “функції” можуть бути різними. Зазвичай процедурою називають підпрограму, що не повертає результату, тоді як функція – підпрограма, що повертає результат, і може виконувати роль частини певного виразу. В іншому ж випадку ці два поняття вважають синонімічними, зокрема в C++. Програма складається з послідовного виконання підпрограм. Такий підхід є хорошим тільки для невеликих програм з чіткою лінійною логікою.

Головною перевагою ПП є його простота та ясність. Завдяки тому, що програми розбиваються на дрібні частини, які виконують конкретні задачі або підзадачі, код є легкозрозумілим і легко підтримуваним. Подібні до C мови програмування часто використовуються саме для процедурного програмування.

Окрім переваг ПП має й певні недоліки, особливо добре помітні, якщо розглядати великі складні системи. Основним таким недоліком є те, що ця парадигма програмування не підтримує інкапсуляцію даних, тобто дані та функції, що їх обробляють, не мають чіткої структури й не об'єднані в логічні групи. Це може призвести до проблем із підтримкою та масштабуванням коду,

					КП.ІПЗ-421.ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

оскільки доволі важко простежити взаємозв'язки між даними та функціями, а якщо це й вдається, то такі зв'язки сприймаються важко, навіть якщо по суті це мало бути щось просте.

Також ПП може мати дублювання коду, оскільки може виникнути ситуація, де, в залежності від контексту, функції не можуть бути повторно використані без їх копіювання. Це ускладнює організацію коду та збільшує ризик виникнення помилок.

2.1.2 Функціональне програмування

Функціональне програмування (ФП) – парадигма, в основі якої лежать математичні функції[6]. У функціональному програмуванні дані не піддаються модифікаціям, а функції не мають побічних ефектів і повертають один і той самий результат для заданих вхідних параметрів, що дає перевагу ФП, яка полягає в його здатності до полегшення багатопотоковості. ФП дає змогу розробляти надійніші та стійкіші прості в тестуванні та підтримці програми. У цій парадигмі увага спрямована на обрахунок значень та використання функцій як перших класів, тобто їх можна передавати як аргументи іншим функціям, повертати як результат виконання інших функцій та зберігати в змінних.

Функціональне програмування має доволі простий синтаксис, високий рівень абстракції та надає можливість паралельного виконання коду. Тому воно вважається надійним та гнучким інструментом розробки програмного забезпечення.

Відомими представниками функціонального програмування є такі мови, як: Scala, Lisp, Clojure, Haskell. ФП часто використовується й у вискорівневих функціях у JavaScript та Python.

Недоліком ФП вважається складність освоєння, особливо якщо програміст звик до процедурного чи об'єктно-орієнтованого програмування. Функціональне програмування також може бути менш ефективним порівняно з іншими парадигмами, якщо брати до уваги використання пам'яті та продуктивність.

					КП.ІПЗ-421.ІЗ	Арк. 11
Змн.	Арк.	№ докум.	Підпис	Дата		

2.1.3 Об'єктно-орієнтоване програмування

Об'єктно-орієнтоване програмування (ООП) – це парадигма, яка розглядає програму, як сукупність об'єктів, які взаємодіють одні з одними; філософія проектування комп'ютерного програмування, яка зосереджується навколо даних або об'єктів, а не функцій і логіки[7]. Кожен такий об'єкт є екземпляром класу, у якому визначено дані й методи обробки даних. ООП підтримує такі концепції, як інкапсуляція, наслідування, поліморфізм та абстракція.

Основною перевагою ООП є його здатність до моделювання реальних об'єктів і процесів за допомогою простих і зручних інструментів. Так як об'єкти класів можна об'єднати в ієрархічні структури, полегшується простеження їх поведінки та взаємодії, що призводить до того, що код стає зрозумілішим та легшим у підтримці.

Інкапсуляція – це розміщення об'єктів у класі разом із методами об'єкта. Цей процес виконує захисну функцію. У більшості випадків до даних класу дають доступ тільки методам усередині класу, у якому ті дані зберігаються. Для отримання тієї інформації організовується зовнішній інтерфейс взаємодії з даними класу: дається доступ тільки до деяких методів класу, до полів класу майже завжди звернутися напямую неможливо. Це не дозволяє іншим об'єктам і зовнішнім кодам отримувати доступ до об'єктів класу або змінювати їх. Інкапсуляція дозволяє приховувати внутрішні деталі реалізації об'єкта і покращує модифікованість та масштабованість коду.

Наслідування дозволяє створювати нові класи на основі вже наявних для повторного використання коду й спрощення його розширення. Завдяки цій концепції будь-який клас може бути батьком або дитиною іншого класу чи класів. Тобто дочірній клас отримує у спадок від батьківського класу (його ще називають суперкласом) поведінку (методи) та властивості (поля) з урахуванням деяких обмежень доступу.

Поліморфізм дозволяє використовувати об'єкти різних класів однаковим чином для забезпечення гнучкості та зручності коду. Ця концепція стосується

					КП.ІІЗ-421.ІІЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

здатності програми інтерпретувати та використовувати один і той самий метод до об'єктів різних класів. Зазвичай це можливо, якщо об'єкти мають різні параметри, але мають спільний батьківський клас.

Абстракція означає відокремлення або приховування реалізації класу від його інтерфейсу. Наприклад, стереосистема може вважатися об'єктом, а кнопки її керування – інтерфейсом. Щоб збільшити чи зменшити гучність, користувач може взаємодіяти з кнопками, але йому не відомо, яким чином дії, зроблені внаслідок натискання тих кнопок, виконані.

Розробники широко використовують ООП в роботі з Java, C++ та Python. Недоліком ООП можна вважати дещо складне проектування та можливість надмірного ускладнення, якщо класи та їх ієрархію спроектовано недостатньо ретельно.

З огляду на розробку програми оброблення даних про книги, об'єктно-орієнтоване програмування є найкращим підходом до розв'язання цієї задачі, завдяки своїй структурованості, гнучкості, повторному використанні коду та інкапсуляції.

2.2 Інструменти розробки

Підбір ефективних інструментів розробки програмного забезпечення є важливим для досягнення успіху для будь-якої команди розробників. Грамотно підібрані засоби розробки значно підвищують продуктивність та забезпечують ефективне управління проектами. Декілька найпоширеніших із них будуть розглядатися далі у розділі.

2.2.1 Sublime Text

Текстовий редактор Sublime Text[8] часто використовують для редагування вихідного коду. Цей редактор підтримує зручне й легке перемикання між проектами, що особливо корисно, коли розробник має

					КП.ІПЗ-421.ІЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

одночасно кілька проектів. Перевагою роботи з Sublime Text є також наявна у ньому велика кількість корисних комбінацій клавіш для пришвидшення роботи з кодом. Завдяки можливості декількох виділень, можна редагувати одночасно кілька рядків коду, що може стати в пригоді при роботі з великими файлами.

2.2.2 IntelliJ IDEA

Інтегроване середовище розробки IntelliJ IDEA містить усі необхідні для розробки програмного забезпечення інструменти. Основу складають, звісно, редактор коду, компілятор та відлагоджувач. IntelliJ IDEA автоматизує виконання багатьох завдань, що добряче полегшує процес розробки. Його інтегровані функції сприяють зручній навігації по коду та аналізу помилок, які супроводжують розробників упродовж створення проекту. Для розробки на Java IntelliJ IDEA є одним з найкращих виборів.

2.2.3 Git та GitHub

Система контролю версій з відкритим кодом Git використовується для програмних проектів, коли необхідно часто редагувати код, взаємодіючи з іншими розробниками та їх кодами, та зберігати всі внесені зміни на кожному з етапів. Git є хорошим інструментом для управління версіями та підтримки історії змін.

Інтернет-сховище GitHub – місце для збереження програмних проектів. Проекти, завантажені туди, можуть бути як загальнодоступними, так і приватними – для однієї людини чи групи осіб. Це зручна платформа для обміну проектами та пошуку однодумців.

2.2.4 Stack Overflow та YouTube

Stack Overflow – найпопулярніший форум серед програмістів. Якщо

					КП.ІПЗ-421.ІЗ	Арк. 14
Змн.	Арк.	№ докум.	Підпис	Дата		

розробник зіткнувся з питанням чи проблемою, вирішити яку так і не зміг, скоріш за все він зможе знайти розв'язку саме на цьому сайті. Ця платформа чудово підходить для обміну знаннями, навчання, пошуку рішень для найрізноманітніших проблем, пов'язаних із процесом розробки.

Відеохостинг YouTube чудово підійде для тих, кому важко вивчати мови програмування, технології тощо, тільки ознайомлюючись із текстовими матеріалами. На цій платформі розміщено дуже багато відеоуроків на різноманітні тематики, де кожен зможе знайти потрібну йому інформацію та засвоїти її в зручній для нього манері.

Серед вищезгаданих інструментів IntelliJ IDEA є найоптимальнішим вибором для розробки програми обробки даних про книги. Це середовище розробки автоматизує значну частину завдань, що значно спростить розробку. А в разі виникнення проблем з написанням коду можна також звернутися до сайту Stack Overflow, де або вже існує відповідь на потрібне питання, або можна отримати її, попросивши допомоги в інших розробників.

2.3 Вимоги до розроблюваного програмного забезпечення

Вимоги до програмного забезпечення:

- читання даних з текстових файлів (формат txt): текстові файли мають бути сформовані таким чином, що кожен рядок міститиме інформацію про книги в такому вигляді «Номер_книги Автор Назва_книги Ціна»;

- фільтрація книг за ціною: у результаті виконання коду програми мають створюватися або перезаписуватися три файли, де міститимуться дані про книги різних цінових категорій: дешеві книги, книги середнього цінового діапазону та дорогі книги;

- сортування даних у межах уже сформованих за ціною файлів: забезпечити два варіанти сортування даних: по автору -> назві книги -> ціні та по ціні -> автору -> назві;

- взаємодія з користувачем: надати інтерфейс взаємодії з користувачем для

					КП.ІПЗ-421.ІЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

забезпечення можливості вибору методу сортування з наявних;

- продуктивність: забезпечити швидку обробку даних при великих обсягах інформації.

Розроблюване програмне забезпечення має бути продуктивним та зручним у використанні, якщо врахувати переваги та недоліки вищезгаданих методів та інструментів. Це має дозволити ефективно обробляти дані про книги та керувати ними, здійснювати їх фільтрацію та сортування.

					КП.ІПЗ-421.ІЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ОПИС ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ ТА ДЕМОНСТРАЦІЯ РОБОТИ ПРОГРАМИ

2.1 Опис вирішення задачі

Для реалізації завдання курсового проєкту було створено такі три класи:

- Book – клас, який описує та зберігає інформацію про книжки;
- Controller – клас, який містить методи обробки даних про книжки: зчитування інформації з файлів, розподіл отриманих даних по відповідних вихідних файлах, сортування даних про книжки та взаємодія з користувачем, який вирішуватиме, за яким параметром потрібно проводити сортування;
- Dispatcher – клас, який містить метод main, з якого починається робота програми та викликається метод з класу Controller, який запускає процес обробки вхідних файлів.

3.1.1 Клас Book

Клас Book має три приватні поля для збереження зчитаної з файлів інформації: рядкові поля author і title та поле типу дійсне число price.

```
class Book implements Comparable<Book> {  
    private String author;  
    private String title;  
    private double price;  
}
```

Конструктор класу (метод Book) приймає три параметри з тими типами даних, які мають об'єкти класу Book, тому не можна створити екземпляр цього класу без указування автора, назви чи ціни книжки.

```
Book(String author, String title, double price) {  
    this.author = author;  
    this.title = title;  
    this.price = price;  
}
```

					КП.ІІЗ-421.ІЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

Методи геттери з відповідними назвами повертають значення полів класу для подальшого їх опрацювання. Методів сеттерів у цьому проекті не передбачено, і поля класу набувають певних значень тільки під час виклику конструктора, тому редагувати інформацію про книжки в ході виконання програми не можна. Якщо потрібно внести якісь зміни в записи, треба робити це у вхідних текстових файлах.

```
public String getAuthor() {
    return this.author;
}
public String getTitle() {
    return this.title;
}
public Double getPrice() {
    return this.price;
}
```

Метод toString() перевизначає однойменний метод батьківського класу Object та повертає рядкове представлення об'єкту у форматі «author title price» (розділювачем є знак табуляції, а не пробіл).

Метод compareTo() перевизначає однойменний метод інтерфейсу Comparable для здійснення можливості порівняння об'єктів та їх подальшого сортування. Цей метод приймає екземпляр класу Book, виконує порівняння об'єкта, який викликав даний метод з об'єктом, який передається, як параметр. Метод повертає значення 1, -1 та 0, якщо об'єкт більший, менший чи рівний об'єкту, який є параметром цього методу. При впровадженні якогось інтерфейсу в програму необхідно реалізувати всі його нереалізовані методи, інакше виникне помилка. У даному випадку потрібно було реалізувати тільки метод compareTo().

```
@Override
public String toString() {
    return this.author + "\t" + this.title + "\t" + this.price;
}
@Override
public int compareTo(Book book) {
```

```

        int result = Double.compare(this.price, book.price);
        if(result != 0) {
            return result;
        } else {
            result = this.author.compareTo(book.author);
            if(result != 0) {
                return result;
            } else {
                return this.title.compareTo(book.title);
            }
        }
    }
}

```

Клас Book описує основні властивості об'єкту книга, має конструктор для створення екземпляра класу та присвоєння йому зчитаних із файлу значень, геттери для отримання цих значень для їх подальшого опрацювання, методи toString() та compareTo() для рядкового представлення та порівняння об'єкта відповідно.

3.1.2 Клас Controller

Клас Controller реалізовує зчитування даних про книжки із вхідних файлів та подальше їх опрацювання. Він містить три публічні константні поля типу змінного масиву ArrayList, елементами якого є екземпляри класу Book. Вони якраз відповідатимуть за збереження інформації про книжки відповідних цінових категорій: дешевих, середньої ціни та дорогих. Ключове слово static пов'язує ці поля з класом, тому для використання цих констант не потрібно створювати об'єкт класу Controller, а маніпулювати ними безпосередньо через клас. Final відповідає за те, щоб масиви не могли міняти своїх значень після ініціалізації.

```

class Controller {
    public static final ArrayList<Book> CHEAP_BOOKS = new
    ArrayList<>();
}

```

					КП.ІІЗ-421.ІЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        public static final ArrayList<Book> MEDIUM_PRICE_BOOKS = new
ArrayList<>();

        public static final ArrayList<Book> EXPENSIVE_BOOKS = new
ArrayList<>();
    }

```

Метод `handleFiles()` реалізує алгоритм обробки даних. Цей метод, як і всі інші методи цього класу статичні, тому викликаються через звернення до класу, а не до об'єкту. Його параметрами є об'єкт типу `File`, який містить шлях до вихідних файлів, а також будь-яка кількість вхідних файлів (варіативний аргумент). Метод є публічним, тож його можна викликати з будь-якої точки програми. Він по черзі викликає методи, які слугують для визначення типу сортування, зчитування файлу, сортування одержаних даних та їх подальшого запису у відповідні файли. Самі файли у циклі `foreach` по черзі відправляються в метод `readFile()` для зчитування даних. У разі провалу будуть опрацьовані винятки й надійдуть відповідні повідомлення про це. Після завершення роботи методу нічого не повертається.

```

        public static void handleFiles(File outputFilesDir, File ...
files) {

            String typeSort = inputSortType();
            for(File file : files) {
                try {
                    readFile(file);
                } catch (FileNotFoundException fnfe) {
                    System.out.println("file for read with path " +
file.getPath() + " not exist!");
                } catch (IOException ioe) {
                    System.out.println("exception is thrown: " +
ioe.getMessage());
                }
            }
            sortBooks(typeSort);
            writeFiles(outputFilesDir);
        }

```

					КП.ІІЗ-421.ІІЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

Приватний метод `inputSortType()` не приймає жодних параметрів і реалізує інтерфейс взаємодії з користувачем. На початку виконання методу в консоль виводиться прохання обрати метод сортування (за автором або ціною). Після того, як користувач зробить свій крок, буде виведено повідомлення про те, що він обрав відповідний вид сортування. Варто зазначити, що, якщо користувач введе не «1» або «2», а що-небудь інше, буде призначено тип сортування за замовчуванням (за ціною). Після завершення роботи метод поверне рядкову змінну – тип сортування.

```
private static String inputSortType() {
    System.out.println("select file sort type:\n1 - author\n2 - price\nsomething other - default type\nEnter: ");
    Scanner input = new Scanner(System.in);
    String str = "";
    try {
        int type = input.nextInt();
        if(type == 1) {
            str = "author";
        } else if(type == 2) {
            str = "price";
        }
    } catch (Exception e) {
        str = "default";
    }
    System.out.println("you choose " + str + " type of sort");
    return str;
}
```

Метод `readFile()` приймає змінну типу `File` з методу `handleFiles()` для зчитування інформації про книги. Змінна `line` у ході виконання методу зберігає кожен рядок, зчитаний із файлу. `BufferedReader` забезпечує ефективне читання тексту з використанням буфера. Цикл `while` за одну ітерацію отримує рядок з файлу, ініціалізує список, який зберігатиме складові рядка, відправляє отриманий рядок та список у метод `splitLine()` для поділу рядка на частини, а тоді добуває зі списку ці значення (окрім ідентифікатора, адже він ніякої ролі не

					КП.ІПЗ-421.ІЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

виконує) і за допомогою них створює об'єкт класу Book та відправляє його в метод `separateBook()` для фільтрування книжок за категоріями. Цикл триватиме, поки рядків у файлі більше не залишиться. Після цього файл закриється й метод не поверне нічого.

```
private static void readFile(File file) throws IOException{
    String line = "";
    BufferedReader br = new BufferedReader(new
    FileReader(file));
    while ((line = br.readLine()) != null){
        ArrayList<String> list = new ArrayList<>();
        splitLine(line, list);
        separateBook(new Book(list.get(1), list.get(2),
        Double.parseDouble(list.get(3))));
    }
    br.close();
}
```

Метод `splitLine()` виконує поділ отриманого з файлу рядка на частини. Приймає рядок і список, у який зберігатиме результат поділу рядка. Розділювачем слугує знак табуляції, а не пробіл, як це зазвичай роблять, бо автор та назва книги можуть складатися з більш ніж одного слова, тому, якщо у файлі різні дані розділені не знаком табуляції, виникнуть проблеми. Рядковий масив `array` зберігає ті частинки після поділу. Потім цикл рухається по масиву `i`, якщо масив не порожній, додає до списку `list` отримані частинки. У результаті виконання методу не повертається нічого.

```
private static void splitLine(String line, ArrayList<String>
list) {
    String[] array = line.split("\t");
    for (String str: array) {
        if (!str.equals("")) {
            list.add(str);
        }
    }
}
```

					КП.ІІЗ-421.ІЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

Метод `separateBook` приймає екземпляр класу `Book` і виконує розподіл книжок за категоріями в залежності від їх ціни. Якщо ціна до 10 включно, книга вважається дешевою і додається до константного поля контролера `CHEAP_BOOKS`. Якщо ціна потрапляє в діапазон 10.01-19 – книга середньої цінової категорії і відправляється в `MEDIUM_PRICE_BOOKS`. Якщо ж ціна перевищує 19, книга дорога й за аналогією переходить до `EXPENSIVE_BOOKS`. Метод нічого не повертає.

```
private static void separateBook(Book book) {
    double price = book.getPrice();
    if(price < 10.01) {
        CHEAP_BOOKS.add(book);
    } else if(price > 19.00) {
        EXPENSIVE_BOOKS.add(book);
    } else {
        MEDIUM_PRICE_BOOKS.add(book);
    }
}
```

Метод `writeFiles` приймає шлях до директорії, яка містить або міститиме вихідні файли. Він намагається по черзі відкрити та опрацювати вихідні файли, якщо це не вдається, опрацьовує виняток `FileNotFoundException` та виводить відповідне повідомлення. Спочатку метод відкриває `cheapBooks.txt` для записування. Далі у циклі проходиться по елементах списку `CHEAP_BOOKS` і записує порядково у файл інформацію: спочатку індекс+1 (для зручності, бо нумерація в списках починається з 0), тоді елемент списку `CHEAP_BOOKS`, а точніше його рядкове представлення, реалізоване вже розглянутим методом `toString()`. Далі `pw.flush()` очищує буфер `PrintWriter`, а `pw.close()` закриває `PrintWriter` та вивільняє ресурси. Далі усе те саме відбувається з `mediumPriceBooks.txt` та `expensiveBooks.txt`. Важливо зазначити, що вихідні файли щоразу в результаті роботи програми перезаписуються, тому, якщо треба дописати у файл лише якісь нові дані, старі зітруться. У результаті виконання методу нічого не повертається.

```
private static void writeFiles(File outputFilesDir) {
```

					КП.ІІЗ-421.ІІЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        try {
            PrintWriter pw = new PrintWriter(new
File(outputFilesDir, "cheapBooks.txt"));
            for(int i = 0; i < CHEAP_BOOKS.size(); i++) {
                pw.println("" + (i+1) + "\t" + CHEAP_BOOKS.get(i));
            }
            pw.flush();
            pw.close();
            pw = new PrintWriter(new File(outputFilesDir,
"mediumPriceBooks.txt"));
            for(int i = 0; i < MEDIUM_PRICE_BOOKS.size(); i++) {
                pw.println("" + (i+1) + "\t" +
MEDIUM_PRICE_BOOKS.get(i));
            }
            pw.flush();
            pw.close();
            pw = new PrintWriter(new File(outputFilesDir,
"expensiveBooks.txt"));
            for(int i = 0; i < EXPENSIVE_BOOKS.size(); i++) {
                pw.println("" + (i+1) + "\t" +
EXPENSIVE_BOOKS.get(i));
            }
            pw.flush();
            pw.close();
        } catch (FileNotFoundException fnfe) {
            System.out.println("file for write not exist!");
        }
    }
}

```

Метод sortBooks() приймає ключ, за яким відбуватиметься сортування. Якщо ключ – "author", книги сортуються за автором, назвою, ціною. Якщо "price", сортування йде за ціною, автором, назвою. Якщо критерій не співпадає ні з "author", ні з "price", застосовується стандартне сортування, визначене у класі Book. У результаті виконання методу нічого не повертається.

					КП.ІІЗ-421.ІІЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		


```

private static void sortBooks (String key) {
    switch (key) {
        case "author": {
            Comparator<Book> authorSort = new Comparator<>() {
                @Override
                public int compare(Book b1, Book b2) {
                    int result =
b1.getAuthor().compareTo(b2.getAuthor());
                    if (result != 0) {
                        return result;
                    } else {
                        result = b1.compareTo(b2);
                        if (result != 0) {
                            return result;
                        } else {
                            return
b1.getTitle().compareTo(b2.getTitle());
                        }
                    }
                }
            };
            Collections.sort(CHEAP_BOOKS, authorSort);
            Collections.sort(MEDIUM_PRICE_BOOKS, authorSort);
            Collections.sort(EXPENSIVE_BOOKS, authorSort);
            break;
        }
        case "price": {
            Comparator<Book> priceSort = new Comparator<>() {
                @Override
                public int compare(Book b1, Book b2) {
                    int result =
b1.getPrice().compareTo(b2.getPrice());
                    if (result != 0) {
                        return result;
                    } else {

```

					КП.ІІІЗ-421.ІІЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        result =
b1.getAuthor().compareTo(b2.getAuthor());
        if (result != 0) {
            return result;
        } else {
            return b1.compareTo(b2);
        }
    }
}

};
Collections.sort(CHEAP_BOOKS, priceSort);
Collections.sort(MEDIUM_PRICE_BOOKS, priceSort);
Collections.sort(EXPENSIVE_BOOKS, priceSort);
break;
}
default:
    Collections.sort(CHEAP_BOOKS);
    Collections.sort(MEDIUM_PRICE_BOOKS);
    Collections.sort(EXPENSIVE_BOOKS);
    break;
}
}

```

Клас Controller реалізує основний функціонал програми, який відповідає за обробку файлів.

3.1.3 Клас Dispatcher

Клас Dispatcher не має полів і містить всього один метод main(), з якого починається робота програми. У цьому методі створюються директорії з вхідними та вихідними файлами. Далі вихідна директорія та вхідні файли (у даному випадку їх 3, але може бути будь-яка кількість) передаються до вже згаданого методу handleFiles() для подальшої їх обробки. Наприкінці в консоль

					КП.ІІІЗ-421.ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

виводиться повідомлення, що файли були прочитані та записані, а також вміст списків з даними, розділеними за категоріями.

```
public class Dispatcher {
    public static void main(String[] args) {
        File        inputFiles        =        new
File("C:\\Users\\User\\IdeaProjects\\courseProject\\src\\inputFiles");
        File        outputFiles        =        new
File("C:\\Users\\User\\IdeaProjects\\courseProject\\src\\outputFiles");
        inputFiles.mkdir();
        outputFiles.mkdir();

        File f1 = new File(inputFiles, "file1.txt");
        File f2 = new File(inputFiles, "file2.txt");
        File f3 = new File(inputFiles, "file3.txt");

        Controller.handleFiles(outputFiles, f1, f2, f3);
        System.out.println("files read and wrote");
        System.out.println(Controller.CHEAP_BOOKS);
        System.out.println(Controller.MEDIUM_PRICE_BOOKS);
        System.out.println(Controller.EXPENSIVE_BOOKS);
    }
}
```

Робота програми починається з методу main(), де ініціалізуються файли й директорії, після чого робота переходить до методу handleFiles(). Там через метод inputSortType() користувач вводить ключ сортування. Далі починається обробка файлів. Спочатку викликається метод readFile(), який у свою чергу викликає splitLine() та separateBook() для поділу рядків на окремі дані та їх фільтрування за категоріями. Потім у handleFiles() викликається метод sortBooks() з передаванням ключа сортування, де дані отримують порядок, заданий користувачем, а тоді у writeFiles() зчитані та відсортовані дані записуються по черзі в три вихідні файли. Після завершення обробки файлів відбувається повернення до методу main() і в консоль виводяться повідомлення

					КП.ІІЗ-421.ІЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

про завершення опрацювання файлів, а також його результат.

3.2 Демонстрація роботи програми

Для демонстрації роботи програми було створено 3 вхідні файли з даними про книжки. На рисунках 3.1-3.3 зображено їх вміст.

1	Taras Shevchenko	Kobzar	40	
2	Freida McFadden	Never lie	15.40	
3	David Baldacci	A calamity of souls	8.99	
4	Leigh Bardugo	The familiar	19	
5	Salaman Rushdie	Knife: After an attempted murder		17.35

Рисунок 3.1 – Вміст файлу file1.txt

1	Alyssa Cole	One of us knows	9.55	
2	Percival Everett	James	18	
3	Amy Tan	The backyard bird chronicles	6.80	
4	Holly Jackson	The Reapperance of Rachel Price	10	
5	Douglas Preston	Extinction	36.20	

Рисунок 3.2 – Вміст файлу file2.txt

1	Emily Henry	Funny story	4.99	
2	Sarah J. Maas	Throne of glass	13	
3	Lysa TerKeurst	Clear mind, peaceful heart		12.99
4	Sarah Adams	The rule book	10.55	
5	Leif Enger	I cheerfully refuse	22.60	
6	Sarah J. Maas	Throne of glass	11	
7	Arah J. Maas	Throne of glass	11	
7	Arah J. Maas	Throne of apples	11	

Рисунок 3.3 – Вміст файлу file3.txt

Результат виконання програми та вмісти вихідних файлів, якщо користувач обирає сортування за автором, зображено на рисунках 3.4-3.7.

```
"C:\Program Files\Java\jdk-17\bin\java.exe"
select file sort type:
1 - author
2 - price
something other - default type
Enter:
1
you choose author type of sort
files read and wrote
[Alyssa Cole    One of us knows 9.55, Amy Tan
[Arah J. Maas   Throne of apples    11.0, A
[Douglas Preston  Extinction  36.2, Leif

Process finished with exit code 0
```

Рисунок 3.4 – Результат сортування за автором

```
1 Alyssa Cole    One of us knows 9.55
2 Amy Tan        The backyard bird chronicles    6.8
3 David Baldacci    A calamity of souls    8.99
4 Emily Henry     Funny story    4.99
5 Holly Jackson   The Reappearance of Rachel Price 10.0
```

Рисунок 3.5 – Вміст cheapBooks.txt (за автором)

```
1 Douglas Preston    Extinction    36.2
2 Leif Enger        I cheerfully refuse    22.6
3 Taras Shevchenko   Kobzar    40.0
```

Рисунок 3.6 – Вміст expensiveBooks.txt (за автором)

```
1 Arah J. Maas   Throne of apples    11.0
2 Arah J. Maas   Throne of glass 11.0
3 Freida McFadden    Never lie    15.4
4 Leigh Bardugo   The familiar    19.0
5 Lysa TerKeurst    Clear mind, peaceful heart    12.99
6 Percival Everett    James    18.0
7 Salaman Rushdie    Knife: After an attempted murder    17.35
8 Sarah Adams     The rule book    10.55
9 Sarah J. Maas   Throne of glass 11.0
10 Sarah J. Maas   Throne of glass 13.0
```

Рисунок 3.7 – Вміст mediunPriceBooks.txt (за автором)

У файлі з книжками середньої цінової категорії чудово видно, що всі автори відсортовані в алфавітному порядку. У 1 та 2 рядках продемонстрований випадок, коли автори однакові й сортування продовжується за назвою книги. А в 9 та 10 рядках видно, що при рівності авторів та назв сортування відбувається за ціною в порядку зростання.

На рисунках 3.8-3.11 показано результати виконання програми та вихідні файли, якщо користувач обере сортування за ціною.

```
"C:\Program Files\Java\jdk-17\bin\java.exe"
select file sort type:
1 - author
2 - price
something other - default type
Enter:
2
you choose price type of sort
files read and wrote
[Emily Henry    Funny story 4.99, Amy Tan
[Sarah Adams    The rule book  10.55, Arah
[Leif Enger I cheerfully refuse 22.6, Dougla
```

Рисунок 3.8 – Результат сортування за ціною

```
1 Emily Henry    Funny story      4.99
2 Amy Tan        The backyard bird chronicles  6.8
3 David Baldacci A calamity of souls  8.99
4 Alyssa Cole    One of us knows 9.55
5 Holly Jackson The Reappearance of Rachel Price 10.0
```

Рисунок 3.9 – Вміст cheapBooks.txt (за ціною)

```
1 Leif Enger     I cheerfully refuse  22.6
2 Douglas Preston Extinction           36.2
3 Taras Shevchenko Kobzar  40.0
```

Рисунок 3.10 – Вміст expensiveBooks.txt (за ціною)

1	Sarah Adams	The rule book	10.55	
2	Arah J. Maas	Throne of apples	11.0	
3	Arah J. Maas	Throne of glass	11.0	
4	Sarah J. Maas	Throne of glass	11.0	
5	Lysa TerKeurst	Clear mind, peaceful heart	12.99	
6	Sarah J. Maas	Throne of glass	13.0	
7	Freida McFadden	Never lie	15.4	
8	Salaman Rushdie	Knife: After an attempted murder	17.35	
9	Percival Everett	James	18.0	
10	Leigh Bardugo	The familiar	19.0	

Рисунок 3.11 – Вміст mediunPriceBooks.txt (за ціною)

У 2-4 рядочках mediumPriceBooks продемонстровано випадки, коли ціна однакова і сортування за автором, і коли ціна й автор однакові й сортування з назвою. Як видно, програма чудово впоралася з поставленим завданням.

Важливою деталлю є те, що між номерами книжок, їх авторами, назвою та ціною стоять знаки табуляції для коректної роботи програми, тому не рекомендується використовувати інші розділювачі. Також не слід редагувати текстові файли в IDE: знаки табуляції проставляються неправильно. Замість цього варто використовувати будь-який текстовий редактор. Якщо не дотримуватись цих рекомендацій, виникне помилка, як на рисунку 3.12.

```
"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.1.4\lib\idea
select file sort type:
1 - author
2 - price
something other - default type
Enter:
1
you choose author type of sort
Exception in thread "main" java.lang.IndexOutOfBoundsException: Create breakpoint : Index 3 out of bounds for length 3 <3 internal lines>
    at java.base/java.util.Objects.checkIndex(Objects.java:359)
    at java.base/java.util.ArrayList.get(ArrayList.java:427)
    at Controller.readFile(Dispatcher.java:105)
    at Controller.handleFiles(Dispatcher.java:72)
    at Dispatcher.main(Dispatcher.java:18)
```

Рисунок 3.12 – Демонстрація некоректної роботи програми при неправильній організації вхідних файлів

Дана програма має простий і зрозумілий алгоритм зчитування, обробки та запису даних у файли та є ефективною для обробки невеликих та середніх обсягів даних.

ВИСНОВКИ

У результаті розробки програмного забезпечення для обробки даних з файлів на тему «Книги» вдалося створити програму для автоматизації процесів каталогізації та зберігання даних про книги.

У першому розділі роботи було детально розглянуто предметну область – книжковий ринок. Було описано інструменти розробки, зокрема середовища розробки.

У другому розділі проаналізовано різні парадигми програмування – процедурне, функціональне та об'єктно-орієнтоване. Обґрунтовано вибір об'єктно-орієнтованого програмування (ООП), як методу програмування, який найкраще підходить для вирішення поставленої задачі. Розглянуто інструменти розробки, такі як Sublime Text, IntelliJ IDEA, Git, GitHub, Stack Overflow та YouTube та обґрунтовано залучення деяких із них у процесі розробки програмного забезпечення для цього проєкту. Сформульовано вимоги до розроблюваного програмного забезпечення.

У третьому розділі докладно описано структуру програми та реалізацію її основних компонентів. Розглянуто класи Book, Controller, Dispatcher та їх взаємозв'язки. Було пред'явлено докази коректної роботи програми у вигляді скріншотів консолі та вхідних/вихідних файлів. Програма успішно зчитала дані з вхідних файлів, відсортувала книги згідно з указаним користувачем ключем та зберегла результати обробки у відповідні вихідні файли. Це свідчить про її ефективність та відповідність поставленим вимогам.

Результати виконаної роботи підтвердили, що автоматизація процесів обробки та каталогізації даних про книги є важливою складовою сучасного книжкового ринку. Розроблена програма забезпечує високу точність, швидкість та зручність роботи з даними. Виконання завдання курсового проєкту розширило знання та покращило навички у сфері програмування, розробки ПЗ та роботи з даними.

					КП.ІПЗ-421.ІЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Medium [електронний ресурс] – 10 Most Popular Programming Languages in 2024 – 2024 – Режим доступу: URL: <https://learnwithstataanalytics.medium.com/10-most-popular-programming-languages-in-2024-dcdada056833> (Дата звернення: 03.03.2024)
2. UA5.ORG [електронний ресурс] – Основи програмування мовою Java – 2024 – Режим доступу: URL: <https://ua5.org/java/370-osnovi-programuvannja-movoju-java.html> (Дата звернення: 10.03.2024)
3. Oracle [електронний ресурс] – Java™ Platform, Standard Edition 8 API Specification – 2024 – Режим доступу: URL: <https://docs.oracle.com/javase/8/docs/api/> (Дата звернення: 23.03.2024)
4. W3schoolsUA [електронний ресурс] – Збирач сміття – 2024 – Режим доступу: URL: <https://w3schoolsua.github.io/hyperskill/garbage-collectors.html#gsc.tab=0> (Дата звернення: 03.03.2024)
5. Дистанційна підтримка освіти школярів [електронний ресурс] – Поняття “підпрограма”, “функція” та “процедура” – 2024 – Режим доступу: URL: <https://disted.edu.vn.ua/courses/learn/12651> (Дата звернення: 29.03.2024)
6. Foxminded [електронний ресурс] – Функціональне програмування та приклади його використання – 2024 – Режим доступу: URL: <https://foxminded.ua/funktsionalne-prohramuvannia/> (Дата звернення: 08.04.2024)
- EPAM Campus [електронний ресурс] – Основні принципи ООП – 2024 – Режим доступу:
URL: <https://training.epam.ua/ua/blog/275> (Дата звернення: 17.04.2024)
8. ITpedia [електронний ресурс] – 10 Software Development Tools – 2024 – Режим доступу: URL: <https://uk.itpedia.nl/2018/04/25/10-software-development-tools/> (Дата звернення: 22.04.2024)