

### **Список использованных источников:**

1. Планировки типовых квартир. URL: <https://accesimobil.md/ru/planirovki-kvartir> (дата обращения: 13.11.2019).
2. Травин В.И. Капитальный ремонт и реконструкция жилых и общественных зданий: Учебное пособие для архитектурных и строительных спец. вузов. – Ростов-на-Дону: Изд-во «Феникс», 2002. – 256 с.
3. Обследование и реконструкция жилых зданий: Учеб. пособ. для студ. высш. уч. завед. / Н.В. Прядко. – Макеевка, 2006. – 157 с.
4. Устройство мансард. URL: <http://www.karkas-dom.ru> (дата обращения: 13.11.2019).
5. Мансарда. Планировка и устройство мансарды в доме. URL: <http://www.builderclub.com> (дата обращения: 10.11.2019).
6. Лощенко А. Реновация жилых домов в массовой застройке 1957–1970 гг. через надстройку верхних этажей Журнал Стройкомплекс Среднего Урала № 216, 03 март 2018. – С. 13-16.

**Тулуб В.О.**

*магістр;*

**Катасва Є.Ю.**

*кандидат технічних наук, доцент,*

*Черкаський державний технологічний університет*

## **РОЗРОБКА РЕКОМЕНДАЦІЙНОЇ АВТОМАТИЗОВАНОЇ СИСТЕМИ ПІДБОРУ ЛІТЕРАТУРИ З ВИКОРИСТАННЯМ КОЕФІЦІЄНТА ПОДІБНОСТІ ЖАККАРА**

Рекомендаційна система – це інструмент, що дозволяє передбачити що саме сподобається користувачеві найбільше зі списку наданих предметів. Системи надання рекомендацій – це досить хороша сучасна альтернатива пошуку з використанням запитів, оскільки вони надають можливість знайти продукти чи медіа-контент, який користувач міг би пропустити або не знайти за своїм запитом. Це робить такі системи незамінною частиною таких сервісів, як Facebook, YouTube, Amazon та багатьох інших.

У більшості випадків, системи надання рекомендацій працюють одним із двох способів. В першому випадку вони орієнтуються на атрибути предметів, котрі подобаються користувачам, з метою їх аналізу і визначення що ще може сподобатися користувачеві. В іншому ж випадку, система покладається на те, що подобається або не подобається іншим користувачам для подальшого визначення індексу подібності між користувачами та надання схожих рекомендацій користувачам із однієї групи.

Також є цілком можливим поєднання цих двох способів для побудови більш потужних рекомендаційних систем. Щоправда, найбільшу увагу при вирішенні проблем надання рекомендацій потрібно звертати на предметну область і уже відповідно до неї вибрати підходящий алгоритм.

У якості предмету дослідження було вирішено розглянути алгоритми пошуку рекомендацій літератури. В наш час, коли швидко змінюються вимоги до працівників, особливо у сфері інформаційних технологій, дуже важливо підтримувати актуальність своїх знань у сфері професійної діяльності. Тут одразу виникає складність: як підібрати щось підходяще для себе високого рівня і при цьому вивчити дійсно актуальний предмет. В цьому вам може допомогти система рекомендацій, основна задача якої, зберегти ваш час, підібравши вам літературу для розгляду.

В основу алгоритму було вирішено закласти інформацію про те, як інші користувачі системи оцінили певну книгу і на основі цього робити передбачення, чи варто рекомендувати її користувачеві, який її ще не читав.

Для обраної предметної області можна змодельовати користувача як його ідентифікатор та множини книг які йому сподобалися і не сподобалися. У даному випадку доцільно використати другий спосіб, де рекомендації надаються на основі визначення подібності користувачів. Для цього можна застосувати коефіцієнт подібності Жаккара для визначення подібності користувачів.

В ході проведення дослідження також розглядалися методи колаборативної фільтрації [3], матриці декомпозиції рекомендацій, кластеризація [4, с. 289] книг та користувачів, методи машинного навчання та інші.

Для реалізації було обрано мову програмування Java, оскільки за її допомогою можна легко написати програму в об'єктно-орієнтованому

стилі, яка виконає всі необхідні обчислення. Теоретично, програма може бути реалізована на будь-якій сучасній мові програмування, яка має можливість для роботи з базами даних, але було обрано Java, як досить надійну мову, яка мінімізує кількість випадкових помилок розробника за рахунок досить жорсткого синтаксису, перевірок на етапі компіляції та інших засобів.

Алгоритм побудований на наступних принципах, програма знаходить книги, які можуть сподобатись користувачеві на основі схожості його смаків з іншими користувачами.

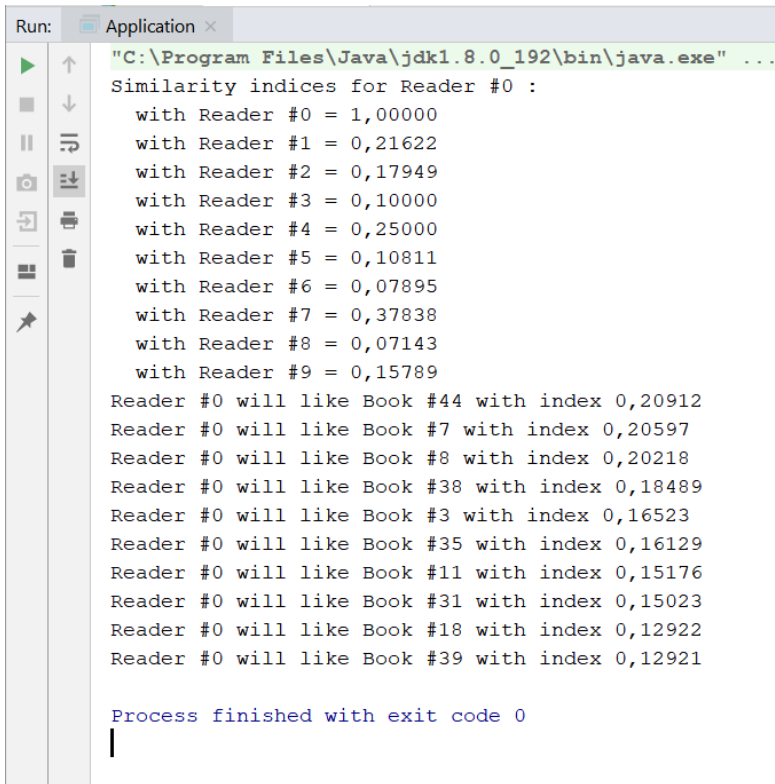
Класи `User`, `Book` та `UserRate` відображають основні сутності програми: користувача, книгу та оцінку книги певним користувачем відповідно. У даному алгоритмі застосовується оцінка книги подобається / не подобається. Клас `Storage` зберігає у собі користувачів, книги, та їх оцінки і надає інтерфейс для отримання рекомендацій. Клас `Application` містить точку входу в додаток і створений виключно для тестування і демонстрації можливостей даного додатку.

При старті програми генеруються тестові дані (10 користувачів, 50 книг та 250 оцінок). Далі для демонстрації і перевірки виводиться коефіцієнт подібності користувачів до першого з них. Потім виводяться рекомендації книг для першого користувача. Вони відсортовані за спаданням коефіцієнта (у прикладі виводяться лише 10 книг, але можна задати будь-яке інше значення).

Варто зазначити, що незначна розбіжність у індексах схожості користувачів зумовлена використанням випадкового набору даних, оскільки в такому випадку оцінка книги користувачем ні від чого не залежить (має псевдовипадкове значення). Для того щоб поліпшити ситуацію, для книги створене додаткове поле – якість. Від його значення залежить чи сподобається книга користувачеві, звичайно воно потрібне тільки для тестування і при використанні алгоритму на реальних даних використання такого підходу є недоцільним, оскільки «якість» книги – досить суб'єктивне поняття.

Було розглянуто вже існуючі розв'язки проблеми надання рекомендацій та розроблено програмне рішення на основі одного з можливих математичних розв'язків даної задачі. Розширення даної програми до повноцінного Web-додатку дозволить створити систему яка дійсно вирішуватиме проблему надання рекомендацій користувачам у зручному вигляді. Також необхідно вдосконалити алгоритм, використовуючи дані, наявні для даної предметної області: авторів,

жанри, тематику, тощо. Для цього можна використати алгоритми кластеризації, методи колаборативної фільтрації, матриці декомпозиції рекомендацій та інші і використовувати усереднений результат для отримання найбільш точних рекомендацій.



```
Run: Application x
"C:\Program Files\Java\jdk1.8.0_192\bin\java.exe" ...
Similarity indices for Reader #0 :
  with Reader #0 = 1,00000
  with Reader #1 = 0,21622
  with Reader #2 = 0,17949
  with Reader #3 = 0,10000
  with Reader #4 = 0,25000
  with Reader #5 = 0,10811
  with Reader #6 = 0,07895
  with Reader #7 = 0,37838
  with Reader #8 = 0,07143
  with Reader #9 = 0,15789
Reader #0 will like Book #44 with index 0,20912
Reader #0 will like Book #7 with index 0,20597
Reader #0 will like Book #8 with index 0,20218
Reader #0 will like Book #38 with index 0,18489
Reader #0 will like Book #3 with index 0,16523
Reader #0 will like Book #35 with index 0,16129
Reader #0 will like Book #11 with index 0,15176
Reader #0 will like Book #31 with index 0,15023
Reader #0 will like Book #18 with index 0,12922
Reader #0 will like Book #39 with index 0,12921

Process finished with exit code 0
|
```

**Рис. 1. Результати роботи програми**

### **Список використаних джерел:**

1. Leskovec J. Mining of Massive Datasets / J. Leskovec, A. Rajaraman, J. Ullman.. – 2014. – 326 с.
2. Ridwan M. Predicting Likes: Inside A Simple Recommendation Engine's Algorithms [Електронний ресурс] / Mahmud Ridwan. – 2015. – Режим доступу до ресурсу: <https://www.toptal.com/algorithms/predicting-likes-inside-a-simple-recommendation-engine>

3. Патент США US8949899B2. Collaborative recommendation system / James H. Errico, M. Ibrahim Sezan, George R. Borden, Gary A. Feather, Mick G. Grover; заявл. 13.06.2005, опубл. 14.12.2006.

4. MacQueen, J.B. (1967). Some Methods for classification and Analysis of Multivariate Observations. University of California Press, pp. 281–297.

**Хохленкова Д.В.**

*студентка,*

*Харківський національний економічний університет  
імені Семена Кузнеця*

## **РОЗРОБКА ЗАБЕЗПЕЧУВАЛЬНОЇ ТЕХНОЛОГІЇ СТВОРЕННЯ ДИЗАЙНУ ІНТЕРФЕЙСУ НАВЧАЛЬНОГО ДОДАТКУ З КАЛІГРАФІЇ**

Технологія, як певний процес, присутній в будь-якій предметній області. Існує три типи технологій: предметна, забезпечувальна та функціональна. Предметна технологія являє собою послідовність технологічних кроків щодо досягнення цілі проектування, при цьому на поточному етапі інструментальні засоби реалізації цих кроків не розглядаються. Забезпечувальні – технології обробки інформації, які можуть використовуватися як інструментарій у різних предметних областях для вирішення різних завдань предметної технології. Функціональна технологія є такою модифікацією забезпечувальної технології, при якій реалізується будь-яка з предметних технологій. Таким чином, забезпечувальна технологія є етапом вибору інструментарію, який є необхідним для реалізації предметної технології.

При створенні дизайну інтерфейсу навчального додатку з каліграфії розробка забезпечувальної технології є необхідним процесом, бо саме на цьому етапі має бути обране програмне забезпечення для реалізації поставлених задач.

Перед тим, як аналізувати необхідне для реалізації навчального додатку програмне забезпечення, треба сформулювати основні вимоги до його інтерфейсу:

- простий, лаконічний інтерфейс;
- легка навігація;