

Міністерство освіти і науки України
Харківський національний університет ім. В. Н. Каразіна
Факультет комп'ютерних наук

Курсова робота

з дисципліни «Об'єктно-орієнтоване програмування»

Тема: «Розробка прототипу програмної системи управління виданнями у
бібліотеці(модифікація 1.5)»

Оцінка _____ / _____

Члени комісії:

_____ Поклонський Є.В.

_____ Нагорний К.А.

_____ Богучарський С.І.

Виконав:

студент 2 курсу, групи КС- 21

Спеціальності:

122 «Комп'ютерні науки»

Сумятін Ярослав Олександрович

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

ЕОМ - електронно-обчислювальна машина.

ПК (РС) – персональний комп'ютер.

UML (Unified Modeling Language) — уніфікована мова об'єктно-орієнтованого моделювання.

ООП - Об'єктно - орієнтоване програмування.

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1 КОНЦЕПЦІЯ.....	5
РОЗДІЛ 2 АНАЛІЗ	6
2.1 Склад системи	6
2.2 Використовувані файли	6
2.3 Основні вимоги програмної системи	6
РОЗДІЛ 3 ПРОЕКТУВАННЯ	7
3.1 Опис предметної області	7
3.2 Класи предметної області	7
3.3 UML-діаграма класів	8
3.4 Опис використовуваних класів	8
РОЗДІЛ 4 ТЕСТУВАННЯ	10
РОЗДІЛ 5 ІНСТРУКЦІЯ ДЛЯ КОРИСТУВАЧА.....	12
ВИСНОВКИ	15
ДОДАТОК А ФОРМАТ ФАЙЛІВ.....	16
ДОДАТОК Б ФРАГМЕНТИ ВИХІДНОГО КОДУ	17

ВСТУП

Курсова робота присвячена об'єктно-орієнтованому програмуванню та одній з основних його мов - Java.

Мова Java поставляється з досить об'ємною бібліотекою класів. Бібліотеки класів Java значно спрощують розробку додатків, надаючи в розпорядження програміста потужні засоби вирішення поширених завдань.

Програми Java зазвичай транслюються в спеціальний байт-код, тому вони можуть працювати на будь-якій комп'ютерній архітектурі за допомогою віртуальної Java-машини. Дата офіційного випуску - 23 травня 1995 року. На 2019 рік Java - одна з найпопулярніших мов програмування. Мета даної курсової роботи полягає у створенні прототипу програмної системи на базі основних принципів об'єктно-орієнтованого дизайну та сучасних практик програмування із використанням об'єктно-орієнтованої мови програмування Java.

В результаті курсової роботи буде розроблено прототип програмної системи «Бібліотека», яка призначена для управління реєстром книжкових видань у бібліотеці.

Для реалізації поставленої мети необхідно ознайомитися з основними принципами Java , потім написати програмний код для програмної системи згідно з отриманим завданням.

РОЗДІЛ 1 КОНЦЕПЦІЯ

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.

Основні можливості системи:

1. Додати видання до реєстру;
2. Видалити видання із реєстру;
3. Отримати список усіх видань, що містяться у реєстрі;
4. Отримати список усіх видань які містять задане слово;
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);
7. Отримати реєстр із сховища (текстового файлу).

Основні принципи та конструкції, що мають бути реалізовані:

1. Агрегація. Бібліотека – сутність, яка містить колекцію видань.
2. Наслідування. Книга та Журнал повинні походити від базової абстракції Видання;
3. Поліморфізм. Кожне видання має поліморфний метод `getType`, який використовується під час вибірки та відображення видання за типом (п.4, п.5 основних можливостей системи).
4. Жанри/спрямованість видання є фіксованою множиною та має належати до перерахування (Enumeration): наукове видання, розважальне видання, фантастика, містика, детектив, драма.
5. Маніпуляція строками, робота із бібліотекою `org.apache.StringUtils`.
6. Асоціативна сутність або відображення `<Користувач, Видання>` (Map), для реалізації видачі книги користувачеві бібліотеки.

РОЗДІЛ 2 АНАЛІЗ

2.1 Склад системи

У комплект поставки входять файли *Library.jar* та *library.csv*.

Library.jar – Java-архів. Основне завдання цього архіву - зберігати файли з класами.

library.csv – текстовий файл, який виступає в ролі реєстру видань.

2.2 Використовувані файли

Програма використовує текстовий файл “*library.csv*” як реєстр для видань. У цьому файлі зберігається інформація про видання різних типів. Типи видань, що зберігаються: Книга (*Book*) та Журнал (*Journal*).

Інформація з файлу завантажується в програму під час її запуску. Інформація завантажується у файл під час виходу з програми через її меню.

Приклад форматування цього файлу надано у додатку А.

2.3 Основні вимоги програмної системи

Вимоги: 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. Перед завершенням роботи системи, реєстр має бути збережений у постійному сховищі, у форматі CSV. Описати відповідну структуру класів розробленого програмного рішення в нотації UML у термінах діаграми класів.

РОЗДІЛ 3 ПРОЕКТУВАННЯ

3.1 Опис предметної області

Основні сутності предметної області:

1. Бібліотека. Має назву, адресу, містить реєстр видань;
2. Книга – неперіодичне видання, яке має назву, одного або декількох авторів, кількість сторінок та жанр (спрямованість). У книги є видавець, який видає книгу у відповідному році.
3. Журнал – періодичне видання, яке має назву, авторів статей та кількість статей та загальну кількість сторінок та спрямованість. У журналу є видавець, який видає журнал із відповідною датою.
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.

3.2 Класи предметної області

Програма має такі класи:

- *Main*
- *Library*
- *bookEdition*
- *Book*
- *Journal*
- *Genre*
- *Client*
- *VectorForBookEdition*
- *VectorForString*

3.3 UML-діаграма класів

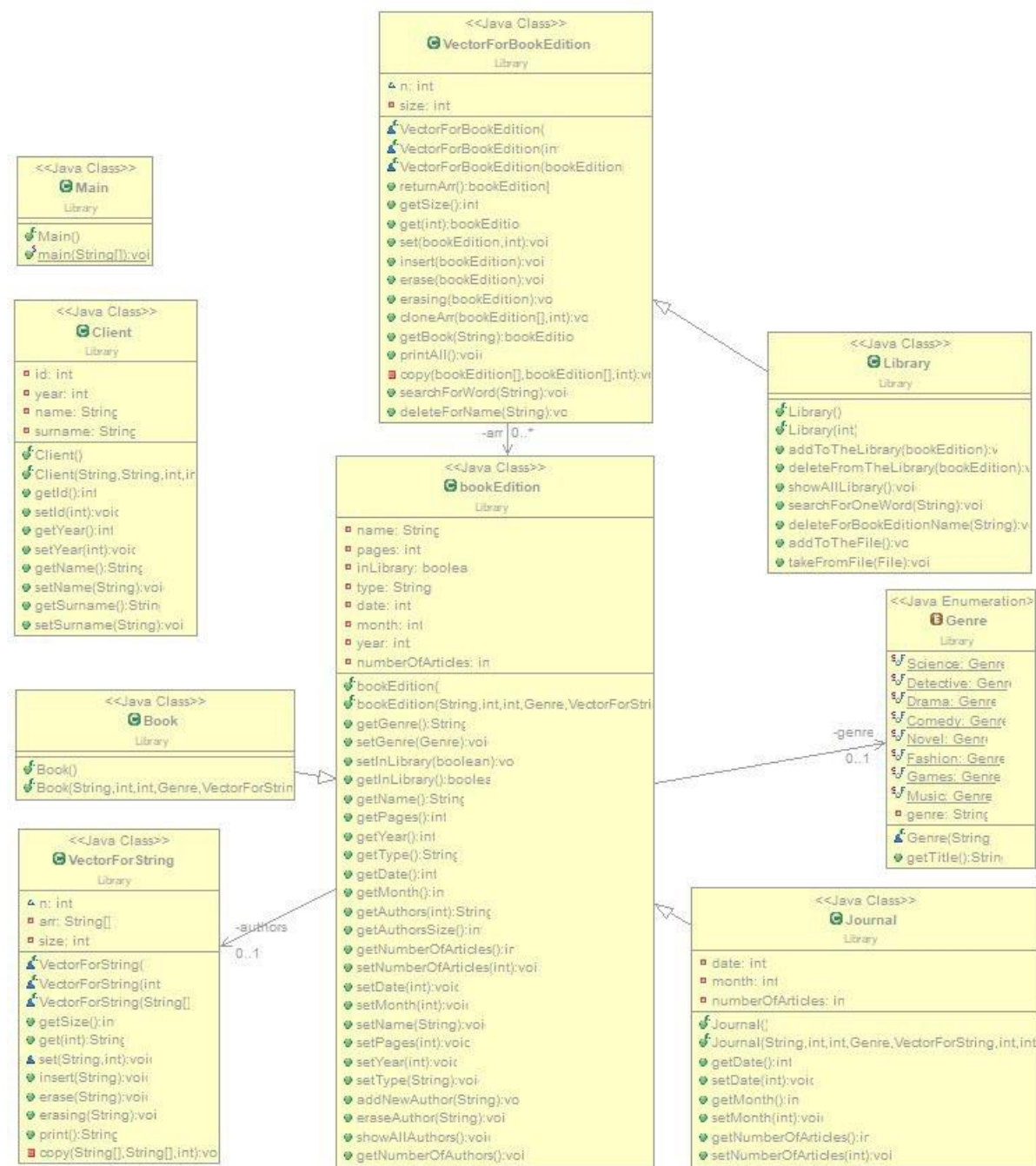


Рисунок 3.1 – UML-діаграма класів

3.4 Опис використовуваних класів

Main є класом, де розташовано головне меню. Користувач може додати нову книгу до бібліотеки, додати новий журнал до бібліотеки, видалити видання з бібліотеки, вивести усі видання, здійснювати пошук по слову, видати книгу клієнту.

Library є класом, який виконує функцію виклику методів для вирішення задач, які користувач потребує у класі *Main*.

Клас *bookEdition* містить основні характеристики, властиві для книжкового видання.

Клас *Book* є спадкоємцем класу *bookEdition*. Цей клас приймає характеристики, які властиві книзі.

Клас *Journal*, так само як і *Book*, є спадкоємцем класу *bookEdition* та приймає характеристики, які властиві журналу.

Перерахування *Genre* – допомагає здійснювати вибір жанру/напрямку нашого видання, при цьому жанр/напрямок є фіксованою множиною.

Клас *Client* відповідає за клієнтів бібліотеки, які можуть брати книги, або замінювати їх на інші.

Клас *VectorForBookEdition* є масивом для видань та має ще інший функціонал.

Клас *VectorForString* є масивом для авторів видань.

У додатку Б надано приклади деяких кодів з цих класів.

РОЗДІЛ 4 ТЕСТУВАННЯ

Класи *Book* та *Journal* – нащадки класу *bookEdition*, тому їх тестування проводилося під час тестування суперкласу.

Тестувалися такі методи класу:

getAuthorSize() – цей метод повертає кількість авторів видання,

addNewAuthor(String newAuthor) – додавання нового автору під час створення видання,

eraseAuthor(String excessAuthor) – видалення автору під час створення видання(на випадок, якщо під час введення було допущено помилку),

showAllAuthors() – виведення всіх введеніх авторів під час створення видання.

Клас *Library* наслідує клас *VectorForBookEdition*, який є допоміжним для зберігання масиву видань. Тому перевірялися як методи класу *Library*, так і наслідувані від *VectorForBookEdition*:

addToTheLibrary(bookEdition bookEdition) – додання створеного видання до бібліотеки,

showAllLibrary() – вивод усіх видань на консоль,

searchForOneWord(String word) – пошук видання за словом у його назві,

deleteForBookEditionName(String name) – видалення видання за його назвою,

addToTheFile() – запис даних у реєстр,

takeFromFile() – запис даних з реєстру.

Після цього основні тести класів було завершено і почалося тестування всієї програми. Меню програми у консолі виглядало так:

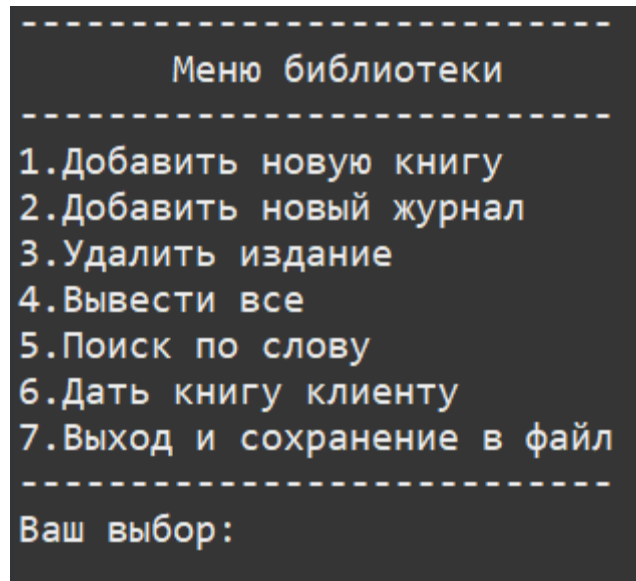


Рисунок 4.1 – Меню програми

Основні можливості кожного пункту меню:

1. Додає нове видання типу Книга (*Book*) до реєстру.
2. Додає нове видання типу Журнал (*Journal*) до реєстру.
3. Видаляє видання за введеною назвою.
4. Виводить у консоль список усіх видать.
5. Шукає видання за введеним словом з його назви.
6. Передає видання (*bookEdition*) клієнту (*Client*) за номером бібліотечного квитка клієнта.
7. Вихід з програми та збереження даних у реєстр.

РОЗДІЛ 5 ІНСТРУКЦІЯ ДЛЯ КОРИСТУВАЧА

Користувач працює з прототипом програмної системи управління виданнями у бібліотеці. Під час запуску програми дані з реєстру(якщо вони є) завантажуються з текстового файлу. Користувач має можливості:

1. Додати нове видання.

```
1
Добавление новой книги:
Введите название книги:
BookExample
|
1.Добавить автора
2.Удалить автора
3.Показать всех авторов
4.Продолжить создание книги
Ваш выбор:
```

Рисунок 5.1 – Приклад додавання нового видання типу Книга

2. Видалити видання.

```
3
Удаление издания:
Введите название издания:
GG WP
|Успешно удалено!
```

Рисунок 5.2 – Приклад видалення видання

3. Знайти видання за словом у його назві.

```

5
Поиск по слову:
Введите слово по которому хотите осуществить поиск:
War
|-----
2 Book

Название: War Never Changes
Жанр: Detective
Автор(авторы):
1.Vladimir Quok
Количество страниц: 123
Год издания книги : 1998
-----

```

Рисунок 5.3 – Приклад пошуку видання за словом у його назві

4. Вивести всі видання на консоль.

```

-----
2 Book

Название: War Never Changes
Жанр: Detective
Автор(авторы):
1.Vladimir Quok
Количество страниц: 123
Год издания книги : 1998
-----

-----
3 Journal

Название: TOP Electro 1998
Направление: Music
Автор(авторы):
1.Roman "Loqiemean" Hudyakov      2.Denis "Gryaz" Astapov
Количество статей: 15
Количество страниц: 123
Дата издания журнала: 15.12.2000
-----

```

Рисунок 5.4 – Приклад виведення видань

5. Видати видання клієнту бібліотеки за номером його бібліотечного квитка (якщо книгу не повернено до бібліотеки, її не можливо видалити з реєстру)

```
6
Введите название книги:
Book 4 Life
Введите номер библиотечного билета:
2
Book Book 4 Life выдана клиенту: Denis Astapov 1989 2
```

Рисунок 5.4 – Приклад видачі книги клієнту

Після завершення програми всі дані зберігаються до реєстру.

Клієнти надані у вигляді масиву об'єктів класу *Client*.

ВИСНОВКИ

В процесі написання даної роботи були з'ясовані такі властивості мови Java:

- Мова програмування об'єктно-орієнтований, оснащений багатою бібліотекою класів і в той же час досить простий для освоєння
- Цикл розробки додатків скорочений за рахунок того, що система побудована на основі інтерпретатора
- Додаток виходить автоматично стерпним між безліччю платформ і операційних систем
- За рахунок вбудованої системи збірки сміття програміст звільняється від необхідності явного управління пам'яттю
- Додаток легко супроводжується і модифікується, тому що модулі можуть бути завантажені з мережі
- В додатки вбудована система безпеки, що не допускає незаконного доступу і проникнення вірусів

В процесі написання даної курсової роботи була так само створена програма, яка добре ілюструє багато гідності даної технології і показує її перспективність. Текст програми знаходиться в додатку А.

Таким чином, розроблений прототип програмної системи «Бібліотека» дає можливість користувачу маніпулювати реєстром видань за допомогою інтерфейсу. Перед завершенням роботи програмної системи реєстр зберігається у постійному сховищі. Все це доводить, що розроблений прототип відповідає поставленому завданню.

ДОДАТОК А

ФОРМАТ ФАЙЛІВ

Програма працює з файлом “*library.csv*”, який є реєстром для видань різного типу: Книга (*Book*) та Журнал (*Journal*). Вони відображаються у файлі по-різному через те, що класи цих видань мають різні поля. Книги зберігаються в файл у вигляді: Тип видання, Назва книги, Жанр, Кількість сторінок, Рік видання, Автори. Журнали, на відміну від книг, мають ще кількість статей та дату і місяць видання, тому зберігаються у наступному вигляді: Тип видання, Назва журналу, Спрямованість, Кількість статей, Кількість сторінок, Дата видання, Автори.

CSV (від англ. Comma-Separated Values - значення, розділені комами) - текстовий формат, призначений для представлення табличних даних. Рядок таблиці відповідає рядку тексту, яка містить одне або кілька полів, розділених комами.

ДОДАТОК Б

ФРАГМЕНТИ ВИХІДНОГО КОДУ

Конструктор Видання (*bookEdition*)

```
public bookEdition(String name, int year, int pages, Genre
genre, VectorForString authors) {
    this.name = name;
    this.pages = pages;
    this.genre = genre;
    this.year = year;
    this.authors = authors;
    this.inLibrary = true; //змінна яка вказує на те,
    //що книгу було/не було віддано клієнту
    //стандартне значення true
}
```

Перерахування Жанр (*Genre*)

```
public enum Genre{

    Science("Science"),
    Detective("Detective"),
    Drama("Drama"),
    Comedy("Comedy"),
    Novel("Novel"),
    Fashion("Fashion"),
    Games("Games"),
    Music("Music");

    private String genre;

    Genre(String genre){
        this.genre = genre;
    }

    public String getTitle() {
        return genre;
    }
}
```

Метод зберігання інформації до реєстру

```

public void addToTheFile() throws IOException {
    if(getSize()==0) {
        System.out.println();
    }
    else {
        File file=new File("library.csv");
        if(!file.exists()) {
            file.createNewFile();
        }

        FileOutputStream writer = new FileOutputStream(file);
        writer.close();

        bookEdition [] arr2= new
bookEdition[returnArr().length];

        cloneArr(arr2,arr2.length);

        try(RandomAccessFile fos = new RandomAccessFile(file,
"rw")){
            for (int i = 0; i < getSize() ; i++) {
                if (arr2[i].getType().equals("Book")) {
                    fos.writeBytes(arr2[i].getType() + ", " +
+ arr2[i].getName() + ", " +
                                arr2[i].getGenre() + ", " +
arr2[i].getPages() + ", " + arr2[i].getYear() +
                                ", ");
                    for(int j = 0; j <
arr2[i].getAuthorsSize(); j++) {
fos.writeBytes(arr2[i].getAuthors(j)+", ");
                    }
                    fos.writeBytes("\n");
                }
                else {
                    fos.writeBytes(arr2[i].getType() + ", " +
+ arr2[i].getName() + ", " +
                                arr2[i].getGenre() + ", " +
arr2[i].getNumberOfArticles() + ", "
                                + arr2[i].getPages() + ",
" + arr2[i].getDate() + "." + arr2[i].getMonth() + "."
                                + arr2[i].getYear()
+ ", ");
                    for(int j = 0; j <
arr2[i].getAuthorsSize(); j++) {
fos.writeBytes(arr2[i].getAuthors(j)+", ");
                    }
                    fos.writeBytes("\n");
                }
            }
        }
    }
}

```

```

        }
    }
}
catch(IOException ex) { }
}
}

```

Метод запису інформації з реєстру

```

public void takeFromFile(File file) {
    Scanner reader = null;

    try {
        reader = new Scanner(file);
    }
    catch (FileNotFoundException e) {
        System.out.println("Файл не знайдено.");
    }
    while (reader.hasNextLine()){
        String line = reader.nextLine();
        String[] data = line.split(", ");
        BookEdition be;
        if(data[0].equals("Book")) {

            be = new Book();
            insert(be);

            be.setName(data[1]);
            be.setGenre(Genre.valueOf(data[2]));
            be.setPages(Integer.parseInt(data[3]));
            be.setYear(Integer.parseInt(data[4]));
            for(int i = 5; i < data.length; i++) {
                be.addNewAuthor(data[i]);
            }

        }
        else {

            be = new Journal();
            insert(be);

            be.setName(data[1]);
            be.setGenre(Genre.valueOf(data[2]));

            be.setNumberOfArticles(Integer.parseInt(data[3]));
            be.setPages(Integer.parseInt(data[4]));

            String[] date = data[5].split("\\.");
            be.setDate(Integer.parseInt(date[0]));
            be.setMonth(Integer.parseInt(date[1]));
            be.setYear(Integer.parseInt(date[2]));
        }
    }
}

```

```
        for(int i = 6; i < data.length; i++) {  
            be.addNewAuthor(data[i]);  
        }  
    }  
}
```

Конструктор класу *Client*

```
public Client(String name, String surname, int year, int id) {  
    this.id = id;  
    this.name = name;  
    this.surname = surname;  
    this.year = year;  
}
```