Informe de Practica 01

Tema: Composicion-Agregacion

Nota

Integrantes	Escuela	Asignatura
Roni Companocca Checco Franco Jesus Cahua Soto William Herderson Choquehuanca Berna	Escuela Profesional de Ingeniería de Sistemas	Semestre: II Código:

Practica	Tema	Duración
01	Composicion-Agregacion	02 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2023 - B	Del 10 de Enero 2024	Al 11 Enero 2024

1. TAREA

- SUCURSALES: La biblioteca dispone de varios locales a los que llama sucursales. Se dispone de uno o varios ejemplares de cada libro, que se encuentran distribuidas por las sucursales. Les interesaría saber por cada libro el número de ejemplares asignados a cada sucursal, y el identificador y nombre únicos de la sucursal junto a la dirección de la sucursal. Se deben hacer consultas por autores y almacenar los autores de cada libro. Puede ocurrir que hayan autores diferentes que se llamen igual. Se debe distinguir a dos autores con el mismo nombre por el libro del que son autores. No puede haber dos autores con el mismo nombre que hayan escrito el mismo libro (distinguiendo a los libros por su identificador único).

2. EQUIPOS, MATERIALES Y TEMAS UTILIZADOS

- Sistema Operativo Windows
- OpenJDK 64-Bits 17.0.7.
- Git 2.39.2.
- Cuenta en GitHub con el correo institucional.

3. URL DE REPOSITORIO GITHUB

- URL para el Repositorio GitHub.
- `https://github.com/RONI-COMPANOCCA-CHECCO`
- URL para la practica 01 en el Repositorio GitHub.
- `https://github.com/RONI-COMPANOCCA-CHECCO/PRACTICA01-FASE03`

4. EJERCICIO

- son los codigos que creamos y a los que modificamos para que nuestro programa compile adecuadamente.

4.1. Clase Branch.java la clase sucursal

```java
import java.util.HashMap;

public class Branch {
    private String id;
    private String name;
    private String address;
    private HashMap<Book, Integer> bookCopies;

    public Branch(String id, String name, String address) {
        this.id = id;
        this.name = name;
        this.address = address;
        this.bookCopies = new HashMap<>();
    }

    public String getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public String getAddress() {
        return address;
    }

    public void addBookCopy(Book book, int copies) {
        bookCopies.put(book, copies);
    }

    public int getBookCopies(Book book) {
        return bookCopies.getOrDefault(book, 0);
    }
    public void printBranchInfo() {
        System.out.println("Sucursal: " + getName());
        System.out.println("ID: " + getId());
        System.out.println("Direccin: " + getAddress());
```

```
        System.out.println("Libros en la sucursal:");

        for (Book book : bookCopies.keySet()) {
            int copies = bookCopies.get(book);
            System.out.println(" - Libro: " + book.getTitle() + ", Ejemplares: " +
                copies);
        }
    }
}
```

4.2. Clase Author.java

```
public class Author {
    private String name;
    private String code;

    public Author(String name) {
        this.name = name;
        this.code = code;
    }

    public String getName() {
        return name;
    }

    public String getCode() {
        return code;
    }
}
```

4.3. Clase Editorial.java

```
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class Editorial {
   private String name;
    private String webPage;
    private boolean status;
    private ArrayList<Article> articles;
    private ArrayList<Branch> branches;
    private ArrayList<Author> authors;
    private Map<Author, List<Book>> authorBooks;

   public Editorial(String name) {
        this.name = name;
        this.articles = new ArrayList<>();
        this.branches = new ArrayList<>();
        this.authors = new ArrayList<>();
        this.authorBooks = new HashMap<>();
```

```
 }

public String getName() {
   return name;
}
public void setName(String name) {
   this.name = name;
}
public String getWebPage() {
   return webPage;
}
public void setWebPage(String webPage) {
   this.webPage = webPage;
}
public boolean isStatus() {
   return status;
}
public void setStatus(boolean status) {
   this.status = status;
}

public boolean addArticle(Article article) {
     if (articles.contains(article)) {
         return false;
     } else {
         this.articles.add(article);

         if (article instanceof Book) {
             Author author = ((Book) article).getAuthor();
             if (!authors.contains(author)) {
                 authors.add(author);
             }

             // Almacenar el libro bajo el autor
             authorBooks.computeIfAbsent(author, k -> new ArrayList<>()).add((Book)
                 article);
         }

         return true;
     }
 }

public boolean addBranch(Branch branch) {
     if (branches.contains(branch)) {
         return false;
     } else {
         branches.add(branch);
         return true;
     }
 }

public boolean searchBooks(String code) {
   for(int i=0;i<articles.size();i++) {
      if(articles.get(i).getCode().equals(code)) {
         return true;
      }
```

```
    }
    return false;
  }
  public ArrayList<Article> getArticles() {
    return articles;
  }

  public void printArticles() {
    for (Article article : articles) {
         System.out.println("Editorial: " + this.getName());
         System.out.println("-------------------------");
         System.out.println("Cdigo: " + article.getCode());
         System.out.println("Ttulo: " + article.getTitle());
         System.out.println("Pginas: " + article.getPages());
         System.out.println("Autor: " + ((Book) article).getAuthor().getName());
         System.out.println("\n");

         /*if (article instanceof Book) {
             System.out.println("ISBN: " + ((Book) article).getISBN());
             System.out.println("Autor: " + ((Book) article).getAuthor().getName());
         } else if (article instanceof Thesis) {

         } else if (article instanceof Journal) {

         }*/

      }

  }
   public void printAuthorBooks() {
       System.out.println("\n----Libros por Autor:----");

       for (Author author : authorBooks.keySet()) {
           System.out.println("Autor: " + author.getName());
           List<Book> books = authorBooks.get(author);

           for (Book book : books) {
               System.out.println(" - Libro: " + book.getTitle());
           }
       }
   }

}
```

4.4. Clase Main.java

```
public class Main {
    public static void main(String[] args) {
        Editorial e1 = new Editorial("UNIVERSIDAD");
        Author author1 = new Author("Robert C. Martin");
        Author author2 = new Author("Marijn Haverbeke");
        Author author3 = new Author("Kyle Simpson");
        Book b1 = new Book("F23-678","Clean Code", author1, 358);
        Book b2 = new Book("F78-098","Eloquent JavaScript", author2, 250);
```

```
        b2.setISBN("978-612-00-0855-3");
        Book b3 = new Book("F67-908","You dont know JavaScript", author3, 410);
        Book b4 = new Book("K58-098","Clean Agile: Back to Basics", author1, 350);
        Book b5 = new Book("K68-098","Agile Software Development", author1, 950);
        Thesis t1 = new Thesis("THESIS-ISIS-01", "Modelo de Analisis del sentimiento en
            la Red Social Facebook usando Redes Neuronales", 118);
        Journal j1 = new Journal("JOURNAL-ISIS-01", "Revista de la Escuela Profesional
            de Ingenieria de Sistemas", 80);

        Branch branch1 = new Branch("BRANCH-001", "Sucursal A", "Direccion A");
        Branch branch2 = new Branch("BRANCH-002", "Sucursal B", "Direccion B");
        Branch branch3 = new Branch("BRANCH-003", "Sucursal C", "Direccion C");
        Branch branch4 = new Branch("BRANCH-004", "Sucursal D", "Direccion D");

        System.out.println((e1.addArticle(b1)) ? "Exito" : "Fracaso");
        System.out.println((e1.addArticle(b2)) ? "Exito" : "Fracaso");
        System.out.println((e1.addArticle(b3)) ? "Exito" : "Fracaso");
        System.out.println((e1.addArticle(b4)) ? "Exito" : "Fracaso");
        System.out.println((e1.addArticle(b5)) ? "Exito" : "Fracaso");
        e1.printArticles();

        branch1.addBookCopy(b1, 5);
        branch1.addBookCopy(b2, 3);
        branch1.addBookCopy(b3, 7);
        branch1.addBookCopy(b4, 12);
        branch2.addBookCopy(b1, 7);
        branch2.addBookCopy(b2, 2);
        branch2.addBookCopy(b3, 8);
        branch2.addBookCopy(b5, 18);
        branch3.addBookCopy(b1, 7);
        branch3.addBookCopy(b2, 2);
        branch3.addBookCopy(b3, 8);
        branch3.addBookCopy(b4, 17);
        branch4.addBookCopy(b1, 7);
        branch4.addBookCopy(b2, 2);
        branch4.addBookCopy(b3, 8);
        branch4.addBookCopy(b4, 13);
        branch4.addBookCopy(b5, 21);

        System.out.println("---INFORMACION DE LAS SUCURSALES---");
        System.out.println("--------------------");
        System.out.println("--------------------");
        branch1.printBranchInfo();
        System.out.println("\n");
        branch2.printBranchInfo();
        System.out.println("\n");
        branch3.printBranchInfo();
        System.out.println("\n");
        branch4.printBranchInfo();

        e1.printAuthorBooks();
    }
}
```

4.5. EJECUCION

```
Editorial: UNIVERSIDAD
------------------------
 Cdigo : F23-678
 Ttulo : Clean Code
 Pginas : 358
Autor: Robert C. Martin


Editorial: UNIVERSIDAD
------------------------
 Cdigo : F78-098
 Ttulo : Eloquent JavaScript
 Pginas : 250
Autor: Marijn Haverbeke


Editorial: UNIVERSIDAD
------------------------
 Cdigo : F67-908
 Ttulo : You dont know JavaScript
 Pginas : 410
Autor: Kyle Simpson


Editorial: UNIVERSIDAD
------------------------
 Cdigo : K58-098
 Ttulo : Clean Agile: Back to Basics
 Pginas : 350
Autor: Robert C. Martin


Editorial: UNIVERSIDAD
------------------------
 Cdigo : K68-098
 Ttulo : Agile Software Development
 Pginas : 950
Autor: Robert C. Martin


---INFORMACION DE LAS SUCURSALES---
--------------------
--------------------
Sucursal: Sucursal A
ID: BRANCH-001
Direcci?n: Direccion A
Libros en la sucursal:
 - Libro: Clean Code, Ejemplares: 5
 - Libro: Eloquent JavaScript, Ejemplares: 3
 - Libro: You dont know JavaScript, Ejemplares: 7
 - Libro: Clean Agile: Back to Basics, Ejemplares: 12


Sucursal: Sucursal B
```

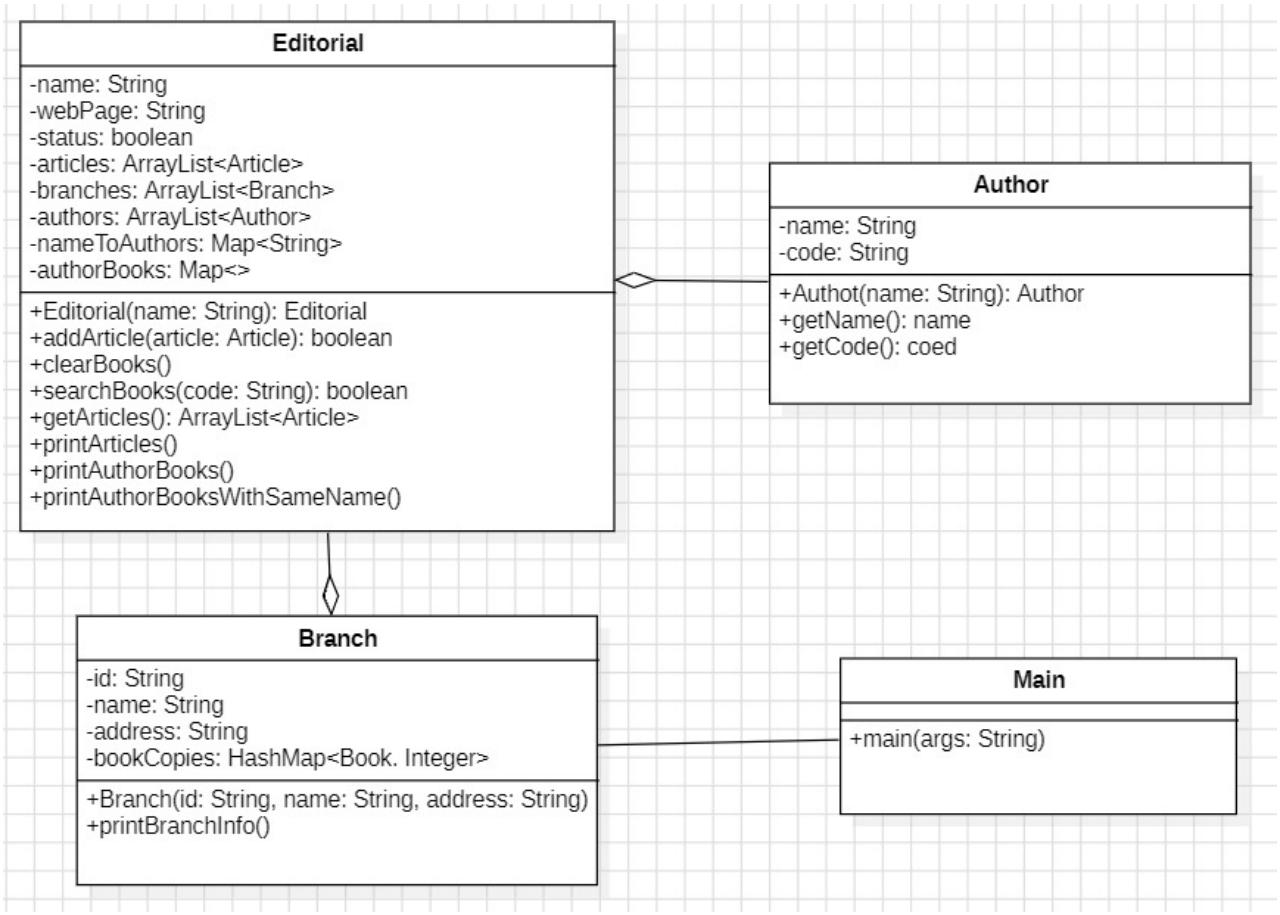
```
ID: BRANCH-002
Direcci?n: Direccion B
Libros en la sucursal:
 - Libro: Clean Code, Ejemplares: 7
 - Libro: Eloquent JavaScript, Ejemplares: 2
 - Libro: Agile Software Development, Ejemplares: 18
 - Libro: You dont know JavaScript, Ejemplares: 8


Sucursal: Sucursal C
ID: BRANCH-003
Direcci?n: Direccion C
Libros en la sucursal:
 - Libro: Clean Code, Ejemplares: 7
 - Libro: Eloquent JavaScript, Ejemplares: 2
 - Libro: You dont know JavaScript, Ejemplares: 8
 - Libro: Clean Agile: Back to Basics, Ejemplares: 17


Sucursal: Sucursal D
ID: BRANCH-004
Direcci?n: Direccion D
Libros en la sucursal:
 - Libro: Clean Code, Ejemplares: 7
 - Libro: Eloquent JavaScript, Ejemplares: 2
 - Libro: Agile Software Development, Ejemplares: 21
 - Libro: You dont know JavaScript, Ejemplares: 8
 - Libro: Clean Agile: Back to Basics, Ejemplares: 13

----Libros por Autor:----
Autor: Robert C. Martin
 - Libro: Clean Code
 - Libro: Clean Agile: Back to Basics
 - Libro: Agile Software Development
Autor: Marijn Haverbeke
 - Libro: Eloquent JavaScript
Autor: Kyle Simpson
 - Libro: You dont know JavaScript
```

- Diagrama UML



5. REFERENCIAS

- M. Aedo, "Fundamentos de Programación 2 - Tópicos de Programación Orientada a Objetos", Primera Edición, 2021, Editorial UNSA.
- `https://github.com/rescobedoq/programacion.git`
- J. Dean, Ïntroduction to programming with Java: A Problem Solving Approach", Third Edition, 2021, McGraw-Hill.
- C. T. Wu, .ᴬn Introduction to Object-Oriented Programming with Java", Fifth Edition, 2010, McGraw-Hill.
- P. Deitel, "Java How to Program", Eleventh Edition, 2017, Prentice Hall.