**Code :**

**package** com.cognizant;

**import** java.io.BufferedReader;

**import** java.io.IOException;

**import** java.io.InputStreamReader;

**import** java.time.LocalDate;

**import** com.cognizant.model.Book;

**import** com.cognizant.model.Subject;

**import** com.cognizant.repository.EntityDao;

**import** com.cognizant.repository.EntityDaoImpl;

/\*\*

\* **@author** Saugata Ray

\*

\*/

**public** **class** Application {

**private** EntityDao entityDao;

Application() {

**this**.entityDao = **new** EntityDaoImpl();

**while** (**true**) {

System.***out***.println("\n");

System.***out***.println("enter '1' to add a new subject");

System.***out***.println("enter '2' to add a new book");

System.***out***.println("enter '3' to delete a subject");

System.***out***.println("enter '4' to delete a book");

System.***out***.println("enter '5' to search a subject");

System.***out***.println("enter '6' to search a book");

System.***out***.println("enter '7' to show all");

System.***out***.println("enter '8' to exit");

String input = gatValFromConsole("Enter your choise ");

select(Integer.*parseInt*(input));

}

}

**public** **static** **void** main(String[] args) {

**new** Application();

}

**public** **void** select(**int** i) {

**switch**(i) {

**case** 1: entityDao.addSubject(addSubject());

**break**;

**case** 2: entityDao.addBook(addBook());

**break**;

**case** 3: entityDao.deleteSubject(deleteSubject());

**break**;

**case** 4: entityDao.deleteBook(deleteBook());

**break**;

**case** 5: Subject subject=entityDao.searchSubject(searchSubject());

System.***out***.println("Result :"+subject);

**break**;

**case** 6: Book book=entityDao.searchBook(searchBook());

System.***out***.println("Result :"+book);

**break**;

**case** 7: showAll();

**break**;

**case** 8: exit();

**break**;

}

}

**private** **void** showAll() {

System.***out***.println("\n");

System.***out***.println("<------Total Subject List------->");

entityDao.showAllSubjects().stream().forEach(System.***out***::println);

System.***out***.println("\n");

System.***out***.println("<------Total Book List------->");

entityDao.showAllBooks().stream().forEach(System.***out***::println);

}

**private** Subject addSubject() {

Subject subject = **new** Subject();;

subject.setSubjectId(Long.*parseLong*(gatValFromConsole("enter subject id")));

subject.setSubTitle(gatValFromConsole("enter subject title"));

subject.setDurationInHours(Integer.*parseInt*(gatValFromConsole("enter duration (In Hours)")));

System.***out***.println(subject.toString());

**return** subject;

}

**private** **long** deleteSubject() {

**long** subjectId=Long.*parseLong*(gatValFromConsole("enter the subject id to be deleted"));

**return** subjectId;

}

**private** **long** searchSubject() {

**long** subjectId=Long.*parseLong*(gatValFromConsole("enter the subject id for search"));

**return** subjectId;

}

**private** Book addBook() {

Book book = **new** Book();

book.setBookId(Long.*parseLong*(gatValFromConsole("enter book id")));

book.setTitle(gatValFromConsole("enter book title"));

book.setPrice(Integer.*parseInt*(gatValFromConsole("enter book price")));

book.setVolume(Integer.*parseInt*(gatValFromConsole("enter book volume")));

book.setPublishDate(LocalDate.*parse*(gatValFromConsole("enter book publish date(yyyy-mm-dd) ")));

System.***out***.println(book.toString());

**return** book;

}

**private** **long** deleteBook() {

**long** bookId=Long.*parseLong*(gatValFromConsole("enter the book id to be deleted"));

**return** bookId;

}

**private** **long** searchBook() {

**long** bookId=Long.*parseLong*(gatValFromConsole("enter the book id for search "));

**return** bookId;

}

**private** String gatValFromConsole(String log) {

System.***out***.println(log);

String inputString = "";

**try** {

BufferedReader bufferRead = **new** BufferedReader(**new** InputStreamReader(System.***in***));

inputString = bufferRead.readLine();

} **catch** (IOException ex) {

ex.printStackTrace();

}

**return** inputString;

}

**private** **void** exit() {

System.*exit*(0);

}

}

**hibernate.cfg.xml**

<?xml version=*'1.0'* encoding=*'utf-8'*?>

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD//EN"

"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<property name=*"hibernate.connection.driver\_class"*>com.mysql.jdbc.Driver</property>

<property name=*"hibernate.connection.url"*>jdbc:mysql://localhost:3306/assignment</property>

<property name=*"hibernate.connection.username"*>root</property>

<property name=*"hibernate.connection.password"*>pass@word1</property>

<property name=*"hibernate.dialect"*>org.hibernate.dialect.MySQL5InnoDBDialect</property>

<property name=*"hibernate.hbm2ddl.auto"*>update</property>

<property name=*"show\_sql"*>false</property>

<mapping class=*"com.cognizant.model.Book"*/>

<mapping class=*"com.cognizant.model.Subject"*/>

</session-factory>

</hibernate-configuration>

**Book.java**

/\*\*

\*

\*/

package com.cognizant.model;

import java.io.Serializable;

import java.time.LocalDate;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

import javax.persistence.Transient;

/\*\*

\* @author Saugata Ray

\*

\*/

@Entity

@Table(name="book")

public class Book implements Serializable{

@Transient

private static final long serialVersionUID = 3667779253735136971L;

@Id

//@GeneratedValue(strategy=GenerationType.AUTO)

@Column(name="BOOKID")

private long bookId;

@Column(name="TITLE")

private String title;

@Column(name="PRICE")

private double price;

@Column(name="VOLUME")

private Integer volume;

@Column(name="PUBLISHDATE")

private LocalDate publishDate;

public long getBookId() {

return bookId;

}

public void setBookId(long bookId) {

this.bookId = bookId;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public double getPrice() {

return price;

}

public void setPrice(double price) {

this.price = price;

}

public Integer getVolume() {

return volume;

}

public void setVolume(Integer volume) {

this.volume = volume;

}

public LocalDate getPublishDate() {

return publishDate;

}

public void setPublishDate(LocalDate publishDate) {

this.publishDate = publishDate;

}

@Override

public String toString() {

return "Book [bookId=" + bookId + ", title=" + title + ", price=" + price + ", volume=" + volume

+ ", publishDate=" + publishDate + "]";

}

}

**Subject.java**

**package** com.cognizant.model;

**import** java.io.Serializable;

**import** java.util.HashSet;

**import** java.util.Set;

**import** javax.persistence.Column;

**import** javax.persistence.ElementCollection;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.JoinColumn;

**import** javax.persistence.JoinTable;

**import** javax.persistence.Table;

**import** javax.persistence.Transient;

/\*\*

\* **@author** Saugata Ray

\*

\*/

@Entity

@Table(name="subject")

**public** **class** Subject **implements** Serializable{

@Transient

**private** **static** **final** **long** ***serialVersionId*** = 1L;

@Id

//@GeneratedValue(strategy=GenerationType.AUTO)

@Column(name="SUBJECTID")

**private** **long** subjectId;

@Column(name="SUBTITLE")

**private** String subTitle;

@Column(name="DURATIONINHOURS")

**private** **int** durationInHours;

@ElementCollection

@JoinTable(name="SUBJECT\_BOOK",

joinColumns=@JoinColumn(name="SUBJECT\_ID")

,inverseJoinColumns=@JoinColumn(name="BOOK\_ID"))

**private** Set<Book> references=**new** HashSet<Book>();

**public** **long** getSubjectId() {

**return** subjectId;

}

**public** **void** setSubjectId(**long** subjectId) {

**this**.subjectId = subjectId;

}

**public** String getSubTitle() {

**return** subTitle;

}

**public** **void** setSubTitle(String subTitle) {

**this**.subTitle = subTitle;

}

**public** **int** getDurationInHours() {

**return** durationInHours;

}

**public** **void** setDurationInHours(**int** durationInHours) {

**this**.durationInHours = durationInHours;

}

**public** Set<Book> getReferences() {

**return** references;

}

**public** **void** setReferences(Set<Book> references) {

**this**.references = references;

}

@Override

**public** String toString() {

**return** "Subject [subjectId=" + subjectId + ", subTitle=" + subTitle + ", durationInHours=" + durationInHours

+ "]";

}

}

**EntityDao.java**

**package** com.cognizant.repository;

**import** java.util.List;

**import** com.cognizant.model.Book;

**import** com.cognizant.model.Subject;

**public** **interface** EntityDao {

**public** **void** addSubject(Subject subject);

**public** **void** addBook(Book book);

**public** **void** deleteSubject(**long** subjectId);

**public** Subject searchSubject(**long** subjectId);

**public** **void** deleteBook(**long** bookId);

**public** Book searchBook(**long** bookId);

**public** List<Subject> showAllSubjects();

**public** List<Book> showAllBooks();

}

**EntityDaoImpl.java**

**package** com.cognizant.repository;

**import** java.util.List;

**import** org.hibernate.Criteria;

**import** org.hibernate.HibernateException;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.cfg.Configuration;

**import** com.cognizant.model.Book;

**import** com.cognizant.model.Subject;

**public** **class** EntityDaoImpl **implements** EntityDao {

**private** SessionFactory sessionFactory;

**public** EntityDaoImpl() {

setSessionFactory(**new** Configuration().configure().buildSessionFactory());

}

**public** SessionFactory getSessionFactory() {

**return** sessionFactory;

}

**public** **void** setSessionFactory(SessionFactory sessionFactory) {

**this**.sessionFactory = sessionFactory;

}

@Override

**public** **void** addSubject(Subject subject) {

Session session = **null**;

**try** {

session = sessionFactory.openSession();

session.beginTransaction();

session.save(subject);

session.getTransaction().commit();

} **catch** (HibernateException e) {

e.printStackTrace();

} **finally** {

closeSession(session);

}

}

@Override

**public** **void** addBook(Book book) {

Session session = **null**;

**try** {

session = sessionFactory.openSession();

session.beginTransaction();

session.save(book);

session.getTransaction().commit();

} **catch** (HibernateException e) {

e.printStackTrace();

} **finally** {

closeSession(session);

}

}

@Override

**public** **void** deleteSubject(**long** subjectId) {

Session session = **null**;

**try** {

session = sessionFactory.openSession();

session.beginTransaction();

Subject subject = session.load(Subject.**class**, subjectId);

session.delete(subject);

session.getTransaction().commit();

} **catch** (HibernateException e) {

e.printStackTrace();

} **finally** {

closeSession(session);

}

}

@Override

**public** Subject searchSubject(**long** subjectId) {

Session session = **null**;

Subject subject = **null**;

**try** {

session = sessionFactory.openSession();

subject = session.get(Subject.**class**, subjectId);

} **catch** (HibernateException e) {

e.printStackTrace();

} **finally** {

closeSession(session);

}

**return** subject;

}

@Override

**public** **void** deleteBook(**long** bookId) {

Session session = **null**;

**try** {

session = sessionFactory.openSession();

session.beginTransaction();

Book book = session.load(Book.**class**, bookId);

session.delete(book);

session.getTransaction().commit();

} **catch** (HibernateException e) {

e.printStackTrace();

} **finally** {

closeSession(session);

}

}

@Override

**public** Book searchBook(**long** bookId) {

Session session = **null**;

Book book = **null**;

**try** {

session = sessionFactory.openSession();

book = session.get(Book.**class**, bookId);

} **catch** (HibernateException e) {

e.printStackTrace();

} **finally** {

closeSession(session);

}

**return** book;

}

@Override

**public** List<Subject> showAllSubjects() {

Session session = **null**;

List<Subject> subjectList = **null**;

**try** {

session = sessionFactory.openSession();

Criteria subjectCriteria = session.~~createCriteria~~(Subject.**class**);

subjectList = subjectCriteria.list();

} **catch** (HibernateException e) {

e.printStackTrace();

} **finally** {

closeSession(session);

}

**return** subjectList;

}

@Override

**public** List<Book> showAllBooks() {

Session session = **null**;

List<Book> bookList = **null**;

**try** {

session = sessionFactory.openSession();

Criteria bookCriteria = session.~~createCriteria~~(Book.**class**);

bookList = bookCriteria.list();

} **catch** (HibernateException e) {

e.printStackTrace();

} **finally** {

closeSession(session);

}

**return** bookList;

}

**public** **void** closeSession(Session session) {

// session.flush();

session.close();

}

}













































