|  |
| --- |
| **Lab06: Association and Aggregation** |

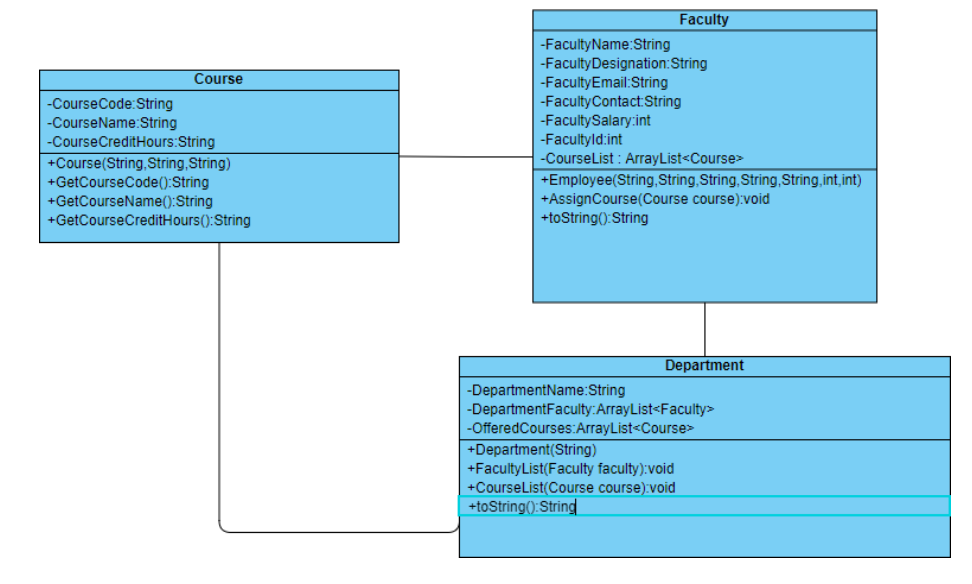
Designing and implementing Java programs that deal with:

|  |
| --- |
| 1. Association 2. Aggregation |

|  |
| --- |
| **Exercises** |

Exercise 1 a (Department)

Consider a department scenario. Create a java program based on the given UML diagram of department based scenario and implement the relation between the classes as shown in the diagram



**Source Code:**

**Faculty:**

package department;

import java.util.ArrayList;

public class Faculty {

private String facultyId;

private String facultyName;

private String designation;

private String contact;

private double salary;

private String email;

private ArrayList<Course> courseList;

public Faculty(String facultyId, String facultyName, String designation, String contact, double salary, String email) {

this.facultyId = facultyId;

this.facultyName = facultyName;

this.designation= designation;

this.contact = contact;

this.salary = salary;

this.email = email;

this.courseList = new ArrayList<>();

}

public void addCourse(Course c) {

courseList.add(c);

}

@Override

public String toString() {

StringBuilder sb = new StringBuilder();

sb.append("Faculty ID: ").append(facultyId).append(", Faculty Name: ").append(facultyName).append("\n");

sb.append("Designation: ").append(designation).append(", Contact: ").append(contact) .append(", Salary: ").append(salary).append(", Email: ").append(email).append("\n");

sb.append("Courses Taught:\n");

for (Course co : courseList) {

sb.append(" ").append(co).append("\n");

}

return sb.toString();

}

}

**Course:**

package department;

public class Course {

private String courseId;

private String courseName;

private int creditHours;

public Course(String courseId, String courseName, int creditHours) {

this.courseId = courseId;

this.courseName = courseName;

this.creditHours = creditHours;

}

@Override

public String toString() {

return "Course ID: " + courseId + ", Course Name: " + courseName + ", Credit Hours: "+ creditHours;

}

}

**Department:**

package department;

import java.util.ArrayList;

public class Department {

private String departmentId;

private String departmentName;

private ArrayList<Course> courses = new ArrayList<>();

private ArrayList<Faculty> faculties = new ArrayList<>();

public Department(String departmentId, String departmentName) {

this.departmentId = departmentId;

this.departmentName = departmentName;

}

public void addCourse(Course c) {

courses.add(c);

}

public void addFaculty(Faculty f) {

faculties.add(f);

}

@Override

public String toString() {

StringBuilder sb = new StringBuilder();

sb.append("Department ID: ").append(departmentId)

.append(", Department Name: ").append(departmentName).append("\n");

sb.append("Courses:\n");

for (Course co : courses) {

sb.append(" ").append(co).append("\n");

}

sb.append("Faculties:\n");

for (Faculty fa : faculties) {

sb.append(" ").append(fa).append("\n");

}

return sb.toString();

}

}

Exercise 1 b (Department)

Create Driver class named as Association\_aggregation\_1. Create proper Objects of all classes as follows • 2 Objects of Course class • 3 Objects of Faculty class • 3 Objects of Department Class And properly display all the information

**Source Code:**

**Association\_aggregation\_1:**

package department;

public class Association\_aggregation\_1 {

public static void main(String[] args) {

Course course1 = new Course("B101", "Digital Design", 9);

Course course2 = new Course("C102", "Physics", 3);

Course course3 = new Course("M113", "Calculus", 2);

Faculty faculty1 = new Faculty("F101", "Dr. Mirza","Professor", "03001117890", 80000.00, "mirza@work.com");

faculty1.addCourse(course1);

faculty1.addCourse(course2);

Faculty faculty2 = new Faculty("F118", "Dr. Waqas", "Assistant Professor", "03021217790", 45000.00, "waqas@work.com");

faculty2.addCourse(course3);

Faculty faculty3 = new Faculty("F116", "Dr. Hanan", "Lecturer", "03021217790", 90000.00, "hanan@gmail.com"); Department dept1 = new Department("K121", "OOP");

dept1.addCourse(course1);

dept1.addCourse(course2);

dept1.addCourse(course3);

dept1.addFaculty(faculty1);

dept1.addFaculty(faculty2);

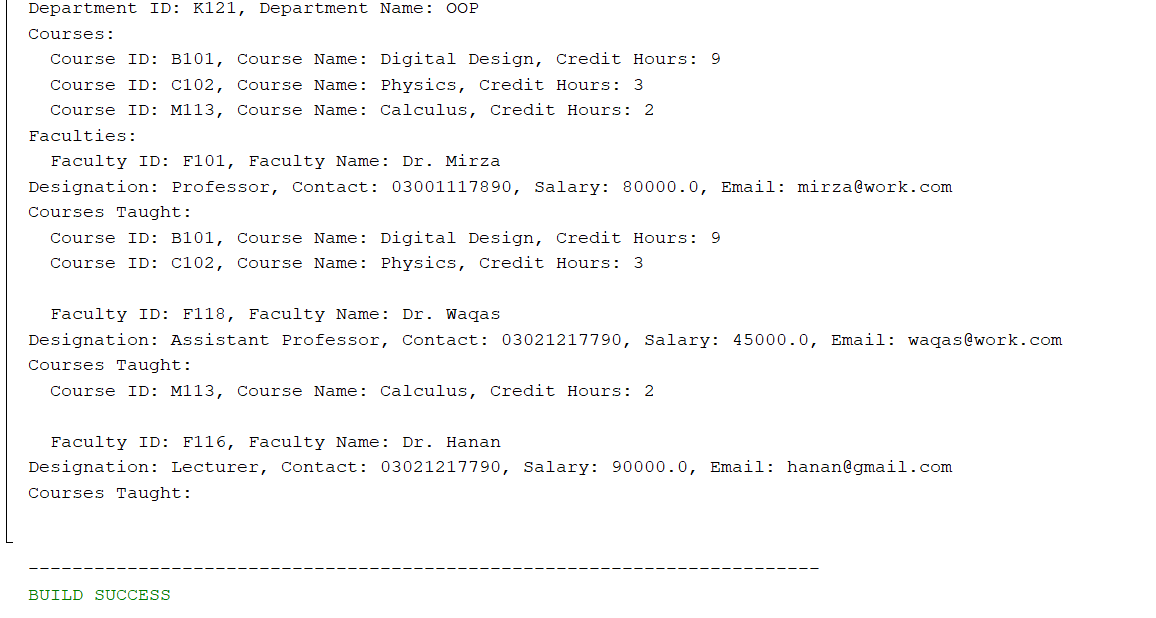
dept1.addFaculty(faculty3);

System.out.println(dept1);

}

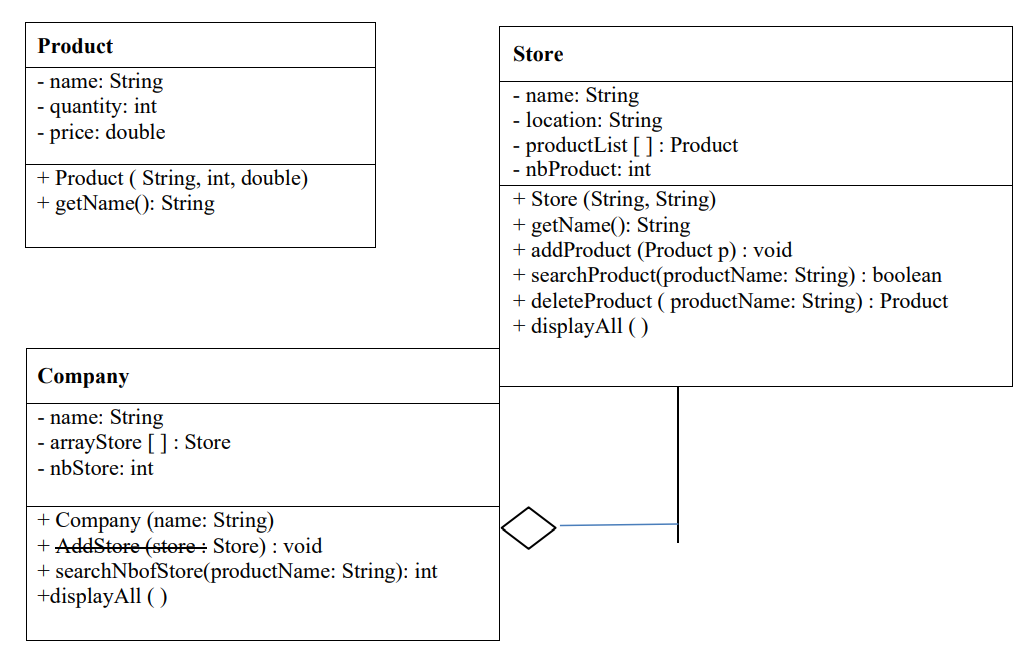
}

**Output:**

****

Exercise 2 a (TestCompany)

A company manages many stores. Each Store contains many Products. The UML diagram of this company system is represented as follow



Class Store:

Attribute: name, location, productList, nbProduct Constructor: Store (name: String, location: String): Method: addProduct() that adds a new product.

Maximum 100 products can be added.

* searchProduct() that accepts the name of product and return True if exist, False otherwise.
* deleteProduct() that accepts the name of product that has to be deleted and returns the deleted object.
* displayAll() prints the name of products available in store. Class Company: Attribute: name, arrayStore, nbStore Constructor: Company (name: string):
* Method: addStore() that adds a new Store.

Maximum 10 stores can be added. searchNbofStore() that accepts the name of product and returens the number of stores containing the product. displayAll() prints the name of stores belongs to company.

**Source Code:**

**Company:**

package testcompany;

import java.util.ArrayList;

public class Company {

private String name;

private ArrayList<Store> arrayStore;

private int nbStore;

public Company(String name) {

this.name = name;

this.arrayStore = new ArrayList<>();

this.nbStore = 0;

}

public void addStore(Store store) {

if (nbStore < 10) {

arrayStore.add(store);

nbStore++;

}

else {

System.out.println("Cannot add more stores. Maximum limit reached.");

}

}

public int searchNbofStore(String productName) {

int count = 0;

for (Store st : arrayStore) {

if (st.searchProduct(productName)) {

count++;

}

}

return count;

}

public void displayAll() {

System.out.println("Stores in a company " + name + ":");

for (Store st : arrayStore) {

System.out.println(" " + st);

}

}

@Override

public String toString() {

return "Company Name: " + name + ", Number of Stores: " + nbStore;

}

}

**Store:**

package testcompany;

import java.util.ArrayList;

public class Store {

private String name;

private String location;

private ArrayList<Product> productList;

private int nbProduct;

public Store(String name, String location) {

this.name = name;

this.location = location;

this.productList = new ArrayList<>();

this.nbProduct = 0;

}

public void addProduct(Product p) {

if (nbProduct < 100) {

productList.add(p);

nbProduct++;

}

else {

System.out.println("Cannot add more products. Maximum limit reached.");

}

}

public boolean searchProduct(String productName) {

for (Product pr : productList) {

if (pr.getName().equalsIgnoreCase(productName)) {

return true;

}

}

return false;

}

public Product deleteProduct(String productName) {

for (Product pr : productList) {

if (pr.getName().equalsIgnoreCase(productName)) {

productList.remove(pr);

nbProduct--;

return pr;

}

}

return null;

}

public void displayAll() {

System.out.println("Products available in the store " + name + ":");

for (Product pr : productList) {

System.out.println(" " + pr);

}

}

@Override

public String toString() {

return "Store Name: " + name + ", Location: " + location + ", Number of Products: " + nbProduct;

}

}

**Product:**

package testcompany;

public class Product {

private String name;

private int quantity;

private double price;

public Product(String name, int quantity, double price) {

this.name = name;

this.quantity = quantity;

this.price = price;

}

public String getName() {

return name;

}

@Override

public String toString() {

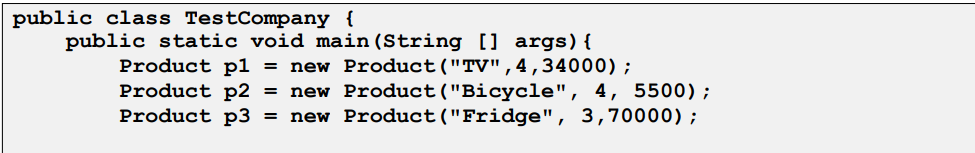
return "Product Name: " + name + ", Quantity: " + quantity + ", Price: " + price;

}

}

Exercise 2 b (TestCompany)

Implement Product, Store and Company classes and use the following class to test.



A screenshot of a computer code

Description automatically generated

**Source Code:**

**TestClass:**

package testcompany;

public class TestClass {

public static void main(String[] args) {

Product p1 = new Product("TV", 4, 34000);

Product p2 = new Product("Bicycle", 4, 5500);

Product p3 = new Product("Fridge", 3, 70000);

Store s1 = new Store("Maven", "Karachi");

Store s2 = new Store("Hyper Mart", "Karachi");

s1.addProduct(p1);

s1.addProduct(p2);

s1.addProduct(p3);

s1.displayAll();

Product tempProduct = s1.deleteProduct("Bicycle");

if (tempProduct != null)

System.out.println("Product " + tempProduct.getName() + " is deleted");

else

System.out.println("There is no product to delete");

s1.displayAll();

s2.addProduct(p1);

s2.addProduct(p2);

s2.addProduct(p3);

s2.displayAll();

Company c1 = new Company("Unilever");

c1.addStore(s1);

c1.addStore(s2);

c1.displayAll();

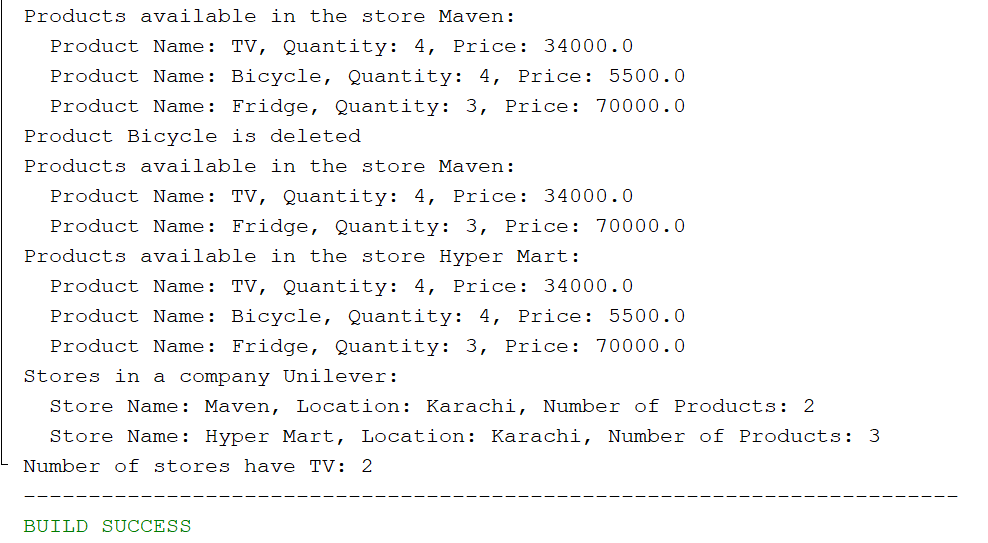
int n = c1.searchNbofStore("TV");

System.out.println("Number of stores have TV: " + n);

}

}

**Output:**



Exercise 3 (Library Management System)

You're tasked with developing a Library Management System in Java. This system should facilitate the management of libraries and their multiple branches. Each library can contain various books, and each branch can maintain its own collection of books. So, Create Java classes to represent the entities involved: Library, LibraryBranch, and Book. Implement methods within these classes to enable functionality such as adding books to a library, removing books, displaying all books in a library branch, etc. Ensure that the relationships between these entities are properly modeled without explicitly mentioning the technical terms for those relationships

**Source Code:**

**TestClass:**

package librarymanagementsystem;

public class TestClass {

public static void main(String[] args) {

Library library = new Library("Central Library");

LibraryBranch branch1 = new LibraryBranch("North Karachi");

LibraryBranch branch2 = new LibraryBranch("Karachi Central");

library.addBranch(branch1);

library.addBranch(branch2);

Book book1 = new Book("To Kill a Mockingbird");

Book book2 = new Book("Once an incident");

Book book3 = new Book("The Great Gatsby");

branch1.addBook(book1);

branch1.addBook(book2);

branch2.addBook(book3);

library.displayAllBranches();

System.out.println();

branch1.displayBooks();

System.out.println();

branch2.displayBooks();

boolean removed = branch1.removeBook("To kill a Mockingbird");

if (removed) {

System.out.println("Successfully removed 'To Kill a Mockingbird' from North Karachi branch.");

}

else {

System.out.println("'To Kill a Mockingbird' not found in North Karachi branch.");

}

System.out.println();

branch1.displayBooks();

}

}

**Library :**

package librarymanagementsystem;

import java.util.ArrayList;

public class Library {

private String libraryName;

private ArrayList<LibraryBranch> branches;

public Library(String libraryName) {

this.libraryName = libraryName;

this.branches = new ArrayList<>();

}

public void addBranch(LibraryBranch branch) {

branches.add(branch);

}

public void displayAllBranches() {

System.out.println("Branches in library " + libraryName + ":");

for (LibraryBranch br : branches) {

System.out.println("- " + br.getBranchName());

}

}

}

**Library branch:**

package librarymanagementsystem;

import java.util.ArrayList;

public class LibraryBranch {

private String branchName;

private ArrayList<Book> books;

public LibraryBranch(String branchName) {

this.branchName = branchName;

this.books = new ArrayList<>();

}

public void addBook(Book book) {

books.add(book);

}

public boolean removeBook(String title) {

for (Book b : books) {

if (b.getTitle().equalsIgnoreCase(title)) {

books.remove(b);

return true;

}

}

return false;

}

public void displayBooks() {

System.out.println("Books in branch " + branchName + ":");

for (Book b : books) {

System.out.println("- " + b.getTitle());

}

}

public String getBranchName() {

return branchName;

}

}

**Book:**

package librarymanagementsystem;

public class Book {

private String title;

public Book(String title) {

this.title = title;

}

public String getTitle() {

return title;

}

}

**Output:**

