**JAVA CODING CHALLENGE**

Coding Challenges: CareerHub, The Job Board

BY,

POOJA SRI B

Mail ID: [poojasribk@gmail.com](mailto:poojasribk@gmail.com)

**1. JobListingClass**

**File:** JobListingClass.java

This class represents a job listing in the CareerHub system. It contains details about a particular job posting, including the job ID, company ID, job title, description, location, salary, type, and the date it was posted. The constructor initializes these details when a new job listing is created, and the class provides getter and setter methods for each attribute to enable access and modification of these details.

**Key Attributes:**

* **jobID:** A unique identifier for the job.
* **companyID:** The unique identifier for the company posting the job.
* **jobTitle:** The title of the job being offered.
* **jobDescription:** A description of the job duties and responsibilities.
* **jobLocation:** The location of the job.
* **salary:** The salary being offered for the job.
* **jobType:** The type of job (e.g., full-time, part-time).
* **postedDate:** The date when the job was posted.
* **companyName:** The name of the company offering the job.

**Constructor:**

The constructor accepts all the attributes and initializes them when a new object of JobListingClass is created.

**MYCODING:**

package org.careerhub.entity;

import org.careerhub.dao;

import java.util.Date;

public class JobListingClass{

private int jobID;

private int companyID;

private String jobTitle;

private String jobDescription;

private String jobLocation;

private double salary;

private String jobType;

private Date postedDate;

private String companyName;

public JobListingClass(int jobID, int companyID, String jobTitle, String jobDescription, String jobLocation, double salary, String jobType, Date postedDate, String companyName) {

this.jobID = jobID;

this.companyID = companyID;

this.jobTitle = jobTitle;

this.jobDescription = jobDescription;

this.jobLocation = jobLocation;

this.salary = salary;

this.jobType = jobType;

this.postedDate = postedDate;

this.companyName = companyName;

}

public int getJobID() {

return jobID;

}

public void setJobID(int jobID) {

this.jobID = jobID;

}

public int getCompanyID() {

return companyID;

}

public void setCompanyID(int companyID) {

this.companyID = companyID;

}

public String getJobTitle() {

return jobTitle;

}

public void setJobTitle(String jobTitle) {

this.jobTitle = jobTitle;

}

public String getJobDescription() {

return jobDescription;

}

public void setJobDescription(String jobDescription) {

this.jobDescription = jobDescription;

}

public String getJobLocation() {

return jobLocation;

}

public void setJobLocation(String jobLocation) {

this.jobLocation = jobLocation;

}

public double getSalary() {

return salary;

}

public void setSalary(double salary) {

this.salary = salary;

}

public String getJobType() {

return jobType;

}

public void setJobType(String jobType) {

this.jobType = jobType;

}

public Date getPostedDate() {

return postedDate;

}

public void setPostedDate(Date postedDate) {

this.postedDate = postedDate;

}

public String getCompanyName() {

return companyName;

}

public void setCompanyName(String companyName) {

this.companyName = companyName;

}

}

**2. Company**

**File:** Company.java

This class represents a company in the Career hub system. Each company has a unique ID, a name, and a location. The constructor initializes these details, and getter and setter methods allow interaction with the class fields.

**Key Attributes:**

* **companyID:** A unique identifier for the company.
* **companyName:** The name of the company.
* **location:** The location where the company operates.

**MYCODE:**

**package** org.careerhub.entity;

**public** **class** Company {

**private** **int** companyID;

**private** String companyName;

**private** String location;

**public** Company(**int** companyID, String companyName, String location) {

**this**.companyID = companyID;

**this**.companyName = companyName;

**this**.location = location;

}

**public** **int** getCompanyID() {

**return** companyID;

}

**public** **void** setCompanyID(**int** companyID) {

**this**.companyID = companyID;

}

**public** String getCompanyName() {

**return** companyName;

}

**public** **void** setCompanyName(String companyName) {

**this**.companyName = companyName;

}

**public** String getLocation() {

**return** location;

}

**public** **void** setLocation(String location) {

**this**.location = location;

}

}

**3. Applicant**

**File:** Applicant.java

This class represents an applicant in the CareerHub system. It includes personal details like the applicant’s ID, first name, last name, email, phone number, and resume. These attributes are essential for storing the applicant's information in the system.

**Key Attributes:**

* **applicantID:** A unique identifier for the applicant.
* **firstName:** The applicant's first name.
* **lastName:** The applicant's last name.
* **email:** The applicant's email address.
* **phone:** The applicant’s phone number.
* **resume:** The applicant's resume, typically stored as a file or a link.

**MYCODE:**

**package** org.careerhub.entity;

**public** **class** Applicant {

**private** **int** applicantID;

**private** String firstName;

**private** String lastName;

**private** String email;

**private** String phone;

**private** String resume;

**public** Applicant(**int** applicantID, String firstName, String lastName, String email, String phone, String resume) {

**this**.applicantID = applicantID;

**this**.firstName = firstName;

**this**.lastName = lastName;

**this**.email = email;

**this**.phone = phone;

**this**.resume = resume;

}

**public** **int** getApplicantID() {

**return** applicantID;

}

**public** **void** setApplicantID(**int** applicantID) {

**this**.applicantID = applicantID;

}

**public** String getFirstName() {

**return** firstName;

}

**public** **void** setFirstName(String firstName) {

**this**.firstName = firstName;

}

**public** String getLastName() {

**return** lastName;

}

**public** **void** setLastName(String lastName) {

**this**.lastName = lastName;

}

**public** String getEmail() {

**return** email;

}

**public** **void** setEmail(String email) {

**this**.email = email;

}

**public** String getPhone() {

**return** phone;

}

**public** **void** setPhone(String phone) {

**this**.phone = phone;

}

**public** String getResume() {

**return** resume;

}

**public** **void** setResume(String resume) {

**this**.resume = resume;

}

}

**4. Jobapplications**

**File:** Jobapplications.java

This class represents an application submitted by an applicant for a specific job. It includes the application ID, job ID, applicant ID, application date, and a cover letter. The constructor initializes these attributes, and getter and setter methods allow access to and modification of the attributes.

**Key Attributes:**

* **applicationID:** A unique identifier for the application.
* **jobID:** The ID of the job for which the applicant has applied.
* **applicantID:** The ID of the applicant submitting the application.
* **applicationDate:** The date when the application was submitted, automatically set to the current date.
* **coverLetter:** The cover letter provided by the applicant as part of their application.

MYCODING:

**package** org.careerhub.entity;

**import** java.time.LocalDateTime;

**public** **class** Jobapplications {

**private** **int** applicationID;

**private** **int** jobID;

**private** **int** applicantID;

**private** LocalDateTime applicationDate;

**private** String coverLetter;

**public** Jobapplications(**int** applicationID, **int** jobID, **int** applicantID, String coverLetter) {

**this**.applicationID = applicationID;

**this**.jobID = jobID;

**this**.applicantID = applicantID;

**this**.applicationDate = LocalDateTime.*now*();

**this**.coverLetter = coverLetter;

}

**public** **int** getApplicationID() {

**return** applicationID;

}

**public** **void** setApplicationID(**int** applicationID) {

**this**.applicationID = applicationID;

}

**public** **int** getJobID() {

**return** jobID;

}

**public** **void** setJobID(**int** jobID) {

**this**.jobID = jobID;

}

**public** **int** getApplicantID() {

**return** applicantID;

}

**public** **void** setApplicantID(**int** applicantID) {

**this**.applicantID = applicantID;

}

**public** LocalDateTime getApplicationDate() {

**return** applicationDate;

}

**public** **void** setApplicationDate(LocalDateTime applicationDate) {

**this**.applicationDate = applicationDate;

}

**public** String getCoverLetter() {

**return** coverLetter;

}

**public** **void** setCoverLetter(String coverLetter) {

**this**.coverLetter = coverLetter;

}

}

**5. Main (Driver Class)**

**File:** Main.java

This is the driver class where the main execution of the CareerHub system starts. It serves as a test class that interacts with the DAO (Data Access Object) implementation. In this class, the following actions are performed:

* A new job listing is added to the system.
* All job listings are fetched and displayed.
* An existing job listing's salary is updated.
* A job listing is deleted from the system.

This class serves as the starting point to demonstrate how the data is being handled in the CareerHub system. It uses the JobListingDaoImplementation to interact with the database and perform CRUD operations on job listings.

MYCODING:

**package** org.main.entity;

**import** org.careerhub.dao.\*;

**import** java.util.List;

**import** org.careerhub.entity.\*;

**public** **class** Main {

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

JobListingDaoImplementation jobDAO = **new** JobListingDaoImplementation();

JobListingClass job = **new** JobListingClass(1, 101, "Software Engineer", "Develop and maintain software", "Bangalore", 100000, "Full-time", **new** java.util.Date(), "TechCorp");

jobDAO.addJobListing(job);

List<JobListingClass> jobList = jobDAO.getAllJobListings();

**for** (JobListingClass j : jobList) {

System.***out***.println("Job Title: " + j.getJobTitle() + ", Company: " + j.getCompanyName());

}

// Update

job.setSalary(110000);

jobDAO.updateJobListing(job);

// Delete

jobDAO.deleteJobListing(1);

}

}

**6. DBproperty**

**File:** DBproperty.java

This class handles the configuration and connection string for connecting to the database. It reads database properties (host, port, user, password, etc.) from a db.properties file and constructs a JDBC connection string. The getConnectionString() method is used to read the properties and return the connection URL for the MySQL database.

**Method:**

* **getConnectionString:** Reads the properties from the file and returns a MySQL connection string to be used for connecting to the database.

**MYCODING:**

**package** org.DBproperty.util;

**import** java.io.FileInputStream;

**import** java.util.Properties;

**public** **class** DBproperty {

**public** **static** String getConnectionString(String propertyFile) {

Properties props = **new** Properties();

String connectionString = "";

**try** {

props.load(**new** FileInputStream(propertyFile));

connectionString = "jdbc:mysql://" + props.getProperty("host") + ":" +

props.getProperty("port") + "/" + props.getProperty("database") +

"?user=" + props.getProperty("user") +

"&password=" + props.getProperty("password");

} **catch** (Exception e) {

e.printStackTrace();

}

**return** connectionString;

}

}

**7. JobListingDAO Interface**

**File:** JobListingDAO.java

This interface defines the basic operations that can be performed on job listings in the CareerHub system. It includes methods for adding, retrieving, updating, and deleting job listings. The implementation of these methods will interact with the database to perform the required operations.

**Methods:**

* **addJobListing:** Adds a new job listing to the database.
* **getAllJobListings:** Retrieves a list of all job listings.
* **getJobById:** Retrieves a job listing by its ID.
* **updateJobListing:** Updates an existing job listing.
* **deleteJobListing:** Deletes a job listing based on its ID.

**MYCODING:**

**package** org.careerhub.dao;

**import** org.careerhub.entity.JobListingClass;

**import** java.util.List;

**public** **interface** JobListingDAO {

**void** addJobListing(JobListingClass job);

List<JobListingClass> getAllJobListings();

JobListingClass getJobById(**int** jobId);

**void** updateJobListing(JobListingClass job);

**void** deleteJobListing(**int** jobId);

}

**8. JobListingDaoImplementation**

**File:** JobListingDaoImplementation.java

This class implements the JobListingDAO interface and provides the actual database interaction for job listings. It uses JDBC to connect to the database and execute SQL queries for adding, retrieving, updating, and deleting job listings.

**Methods:**

* **addJobListing:** Executes an INSERT SQL query to add a new job listing to the job\_listings table.
* **getAllJobListings:** Executes a SELECT SQL query to fetch all job listings from the job\_listings table and returns a list of JobListingClass objects.
* **getJobById:** Executes a SELECT SQL query to fetch a specific job listing by its ID.
* **updateJobListing:** Executes an UPDATE SQL query to modify the details of an existing job listing.
* **deleteJobListing:** Executes a DELETE SQL query to remove a job listing from the job\_listings table.

**Mycoding:**

**package** org.careerhub.dao;

**import** org.careerhub.entity.JobListingClass;

**import** org.DBproperty.util.DBproperty;

**import** java.sql.\*;

**import** java.util.ArrayList;

**import** java.util.List;

**public** **class** JobListingDaoImplementation **implements** JobListingDAO {

**private** Connection getConnection() **throws** SQLException {

String url = DBproperty.*getConnectionString*("db.properties");

**return** DriverManager.*getConnection*(url);

}

**public** **void** addJobListing(JobListingClass job) {

String query = "INSERT INTO job\_listings (jobID, companyID, jobTitle, jobDescription, jobLocation, salary, jobType, postedDate, companyName) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)";

**try** (Connection connection = getConnection(); PreparedStatement ps = connection.prepareStatement(query)) {

ps.setInt(1, job.getJobID());

ps.setInt(2, job.getCompanyID());

ps.setString(3, job.getJobTitle());

ps.setString(4, job.getJobDescription());

ps.setString(5, job.getJobLocation());

ps.setDouble(6, job.getSalary());

ps.setString(7, job.getJobType());

ps.setDate(8, **new** java.sql.Date(job.getPostedDate().getTime()));

ps.setString(9, job.getCompanyName());

ps.executeUpdate();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

**public** List<JobListingClass> getAllJobListings() {

List<JobListingClass> jobList = **new** ArrayList<>();

String query = "SELECT \* FROM job\_listings";

**try** (Connection connection = getConnection(); Statement stmt = connection.createStatement(); ResultSet rs = stmt.executeQuery(query)) {

**while** (rs.next()) {

JobListingClass job = **new** JobListingClass(

rs.getInt("jobID"),

rs.getInt("companyID"),

rs.getString("jobTitle"),

rs.getString("jobDescription"),

rs.getString("jobLocation"),

rs.getDouble("salary"),

rs.getString("jobType"),

rs.getDate("postedDate"),

rs.getString("companyName")

);

jobList.add(job);

}

} **catch** (SQLException e) {

e.printStackTrace();

}

**return** jobList;

}

**public** JobListingClass getJobById(**int** jobId) {

JobListingClass job = **null**;

String query = "SELECT \* FROM job\_listings WHERE jobID = ?";

**try** (Connection connection = getConnection(); PreparedStatement ps = connection.prepareStatement(query)) {

ps.setInt(1, jobId);

**try** (ResultSet rs = ps.executeQuery()) {

**if** (rs.next()) {

job = **new** JobListingClass(

rs.getInt("jobID"),

rs.getInt("companyID"),

rs.getString("jobTitle"),

rs.getString("jobDescription"),

rs.getString("jobLocation"),

rs.getDouble("salary"),

rs.getString("jobType"),

rs.getDate("postedDate"),

rs.getString("companyName")

);

}

}

} **catch** (SQLException e) {

e.printStackTrace();

}

**return** job;

}

**public** **void** updateJobListing(JobListingClass job) {

String query = "UPDATE job\_listings SET jobTitle = ?, jobDescription = ?, jobLocation = ?, salary = ?, jobType = ?, companyName = ? WHERE jobID = ?";

**try** (Connection connection = getConnection(); PreparedStatement ps = connection.prepareStatement(query)) {

ps.setString(1, job.getJobTitle());

ps.setString(2, job.getJobDescription());

ps.setString(3, job.getJobLocation());

ps.setDouble(4, job.getSalary());

ps.setString(5, job.getJobType());

ps.setString(6, job.getCompanyName());

ps.setInt(7, job.getJobID());

ps.executeUpdate();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

**public** **void** deleteJobListing(**int** jobId) {

String query = "DELETE FROM job\_listings WHERE jobID = ?";

**try** (Connection connection = getConnection(); PreparedStatement ps = connection.prepareStatement(query)) {

ps.setInt(1, jobId);

ps.executeUpdate();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

}

**9. Database Configuration (db.properties)**

**File:** db.properties

This file contains the database connection properties needed to connect to the MySQL database. It includes details such as:

* **host:** The database server's hostname.
* **port:** The port on which MySQL is running.
* **database:** The name of the database.
* **user:** The username to authenticate with the database.
* **password:** The password for the database user.

These properties are loaded by the DBproperty class to create a connection string for the application.

host=localhost

port=3306

database=CareerhubDB

user=root

password=Pooja@14