**COLLECTION ASSIGNMENT**

**Sushma S - 46120864**

**Consider the below class structure:**

Diagram

Description automatically generated

Perform below operations with employee object. Create a menu driven approach to achieve the following:

1. Save Unique Employee

* 1. 2. Sort Employee

ByEmployeeId

ByFirstName

ByLastName

BySalary

ByAddress

ByDepartment

* 1. 3. Exit

Create appropriate Class structure below validation:

* + 1. a. Employee ID should be 5Digits and 2 chars (12345\_FS)
  1. b. Salary between 20000 to 5lak
  2. c. firstName, lastName should be alphabets.
  3. d. Join date must be today/future date.
  4. **Source-Code:**

**Employee.java:**

**package** com.cg.assigncollections;

**public** **class** Employee {

**private** String employeeId;

**private** String firstName;

**private** String lastName;

**private** **int** salary;

**private** String joinDate;

**private** Address address;

**private** Department department;

**public** Employee(String employeeId, String firstName, String lastName, **int** salary, String joinDate, Address address,

Department department) {

**super**();

**this**.employeeId = employeeId;

**this**.firstName = firstName;

**this**.lastName = lastName;

**this**.salary = salary;

**this**.joinDate = joinDate;

**this**.address = address;

**this**.department = department;

}

**public** String getEmployeeId() {

**return** employeeId;

}

**public** **void** setEmployeeId(String employeeId) {

**this**.employeeId = employeeId;

}

**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** **int** getSalary() {

**return** salary;

}

**public** **void** setSalary(**int** salary) {

**this**.salary = salary;

}

**public** String getJoinDate() {

**return** joinDate;

}

**public** **void** setJoinDate(String joinDate) {

**this**.joinDate = joinDate;

}

**public** Address getAddress() {

**return** address;

}

**public** **void** setAddress(Address address) {

**this**.address = address;

}

**public** Department getDepartment() {

**return** department;

}

**public** **void** setDepartment(Department department) {

**this**.department = department;

}

}

**Department.java:**

**package** com.cg.assigncollections;

**public** **class** Department {

**private** **int** departmentId;

**private** String departmentName;

**private** String location;

**public** Department(**int** departmentId, String departmentName, String location) {

**super**();

**this**.departmentId = departmentId;

**this**.departmentName = departmentName;

**this**.location = location;

}

**public** **int** getDepartmentId() {

**return** departmentId;

}

**public** **void** setDepartmentId(**int** departmentId) {

**this**.departmentId = departmentId;

}

**public** String getDepartmentName() {

**return** departmentName;

}

**public** **void** setDepartmentName(String departmentName) {

**this**.departmentName = departmentName;

}

**public** String getLocation() {

**return** location;

}

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

**this**.location = location;

}

}

**Address.java:**

**package** com.cg.assigncollections;

**public** **class** Address {

**private** **int** addressId;

**private** String addressLine1;

**private** String city;

**private** String state;

**public** Address(**int** addressId, String addressLine1, String city, String state) {

**super**();

**this**.addressId = addressId;

**this**.addressLine1 = addressLine1;

**this**.city = city;

**this**.state = state;

}

**public** **int** getAddressId() {

**return** addressId;

}

**public** **void** setAddressId(**int** addressId) {

**this**.addressId = addressId;

}

**public** String getAddressLine1() {

**return** addressLine1;

}

**public** **void** setAddressLine1(String addressLine1) {

**this**.addressLine1 = addressLine1;

}

**public** String getCity() {

**return** city;

}

**public** **void** setCity(String city) {

**this**.city = city;

}

**public** String getState() {

**return** state;

}

**public** **void** setState(String state) {

**this**.state = state;

}

}

**ByEmployeeId.java:**

**package** com.cg.assigncollections;

**import** java.util.Comparator;

**public** **class** ByEmployeeId **implements** Comparator<Employee> {

@Override

**public** **int** compare(Employee e1, Employee e2) {

**return** e1.getEmployeeId().compareTo(e2.getEmployeeId());

}

}

**ByFirstName.java:**

**package** com.cg.assigncollections;

**import** java.util.Comparator;

**public** **class** ByFirstName **implements** Comparator<Employee> {

@Override

**public** **int** compare(Employee e1, Employee e2) {

**return** e1.getFirstName().compareTo(e2.getFirstName());

}

}

**ByLastName.java:**

**package** com.cg.assigncollections;

**import** java.util.Comparator;

**public** **class** ByLastName **implements** Comparator<Employee> {

@Override

**public** **int** compare(Employee e1, Employee e2) {

**return** e1.getLastName().compareTo(e2.getLastName());

}

}

**BySalary.java:**

**package** com.cg.assigncollections;

**import** java.util.Comparator;

**public** **class** BySalary **implements** Comparator<Employee> {

@Override

**public** **int** compare(Employee e1, Employee e2) {

**if** (e1.getSalary() == e2.getSalary()) {

**return** 0;

} **else** **if** (e1.getSalary() > e2.getSalary()) {

**return** 1;

} **else** {

**return** -1;

}

}

}

**ByAddress.java:**

**package** com.cg.assigncollections;

**import** java.util.Comparator;

**public** **class** ByAddress **implements** Comparator<Employee> {

@Override

**public** **int** compare(Employee e1, Employee e2) {

**return** e1.getAddress().getCity().compareTo(e2.getAddress().getCity());

}

}

**ByDepartment.java:**

**package** com.cg.assigncollections;

**import** java.util.Comparator;

**public** **class** ByDepartment **implements** Comparator<Employee> {

@Override

**public** **int** compare(Employee e1, Employee e2) {

**return** e1.getDepartment().getDepartmentName().compareTo(e2.getDepartment().getDepartmentName());

}

}

**MainClass.java:**

**package** com.cg.assigncollections;

**import** java.time.\*;

**import** java.time.format.DateTimeFormatter;

**import** java.util.\*;

**import** java.util.regex.Matcher;

**import** java.util.regex.Pattern;

**public** **class** MainClass {

**public** **static** **boolean** isEmployeeIDValid(String employeeID) {

String ID\_REGEX = "[0-9]{5}" + "\_" + "[A-Z]{2}";

Pattern pattern = Pattern.*compile*(ID\_REGEX);

Matcher matcher = pattern.matcher(employeeID);

**return** matcher.matches();

}

**public** **static** **boolean** isSalaryValid(**int** employeeSalary) {

**if** (employeeSalary > 200000 || employeeSalary < 500000) {

**return** **true**;

} **else** {

**return** **false**;

}

}

**public** **static** **boolean** isFirstName(String employeeFirstName) {

String FIRST\_REGEX = "[a-zA-Z]\*";

Pattern pattern = Pattern.*compile*(FIRST\_REGEX);

Matcher matcher = pattern.matcher(employeeFirstName);

**return** matcher.matches();

}

**public** **static** **boolean** isLastName(String employeeLastName) {

String LAST\_REGEX = "[a-zA-Z]\*";

Pattern pattern = Pattern.*compile*(LAST\_REGEX);

Matcher matcher = pattern.matcher(employeeLastName);

**return** matcher.matches();

}

**public** **static** **boolean** isCorrectDate(String employeeDate) {

**final** DateTimeFormatter DATE\_PARSER = DateTimeFormatter.*ofPattern*("dd-MM-yyyy");

LocalDate today = LocalDate.*now*();

LocalDate myDate = LocalDate.*parse*(employeeDate, DATE\_PARSER);

**if** (myDate.isAfter(today) || myDate.isEqual(today)) {

**return** **true**;

} **else** {

**return** **false**;

}

}

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

**int** empSal = 0;

String empFname = **null**;

String empLname = **null**;

String empDoj = **null**;

**int** choice = 0;

String empId = **null**;

**int** empAddId = 0;

String empAddLine = **null**;

String empCity = **null**;

String empState = **null**;

**int** empDeptId = 0;

String empDeptName = **null**;

String empDeptLoc = **null**;

@SuppressWarnings("resource")

Scanner scan = **new** Scanner(System.***in***);

List<Employee> empList = **new** ArrayList<>();

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

System.***out***.println("1.Save Unique Employee");

System.***out***.println("2.Sort Employee");

System.***out***.println("3.Exit");

System.***out***.println("Enter the choice");

choice = scan.nextInt();

**switch** (choice) {

**case** 1:

System.***out***.println("Enter the number of Employees");

**int** num = scan.nextInt();

**for** (**int** i = 0; i < num; i++) {

System.***out***.println("Enter the employee Id(Employee ID should be 5 Digits and 2 Uppercase chars seperated by \_ ):");

scan.nextLine();

empId = scan.nextLine();

**if** (*isEmployeeIDValid*(empId)) {

System.***out***.println("Employee Id:" + empId);

} **else** {

System.***out***.println("Enter the employee Id as specified");

**break**;

}

System.***out***.println("Enter the salary(Salary between 20000 to 5lak):");

empSal = scan.nextInt();

**if** (*isSalaryValid*(empSal)) {

System.***out***.println("Salary:" + empSal);

} **else** {

System.***out***.println("Enter the salary as specified");

**break**;

}

System.***out***.println("Enter the First name(should be alphabets):");

scan.nextLine();

empFname = scan.nextLine();

**if** (*isFirstName*(empFname)) {

System.***out***.println("Employee First Name:" + empFname);

} **else** {

System.***out***.println("Enter the first name as specified");

**break**;

}

System.***out***.println("Enter the Last name(should be alphabets):");

empLname = scan.nextLine();

**if** (*isLastName*(empLname)) {

System.***out***.println("Employee Last Name:" + empLname);

} **else** {

System.***out***.println("Enter the last name as specified");

**break**;

}

System.***out***.println("Enter the date of joining(Join date must be today/future date.Format: dd-mm-yyyy):");

empDoj = scan.nextLine();

**if** (*isCorrectDate*(empDoj)) {

System.***out***.println("Employee Date of joining:" + empDoj);

System.***out***.println("Enter the Address id(number)");

empAddId = scan.nextInt();

System.***out***.println("Enter the Address line");

scan.nextLine();

empAddLine = scan.nextLine();

System.***out***.println("Enter the city");

empCity = scan.nextLine();

System.***out***.println("Enter the state");

empState = scan.nextLine();

System.***out***.println("Enter the Department Id(number)");

empDeptId = scan.nextInt();

System.***out***.println("Enter the Department name");

scan.nextLine();

empDeptName = scan.nextLine();

System.***out***.println("Enter the Department location");

empDeptLoc = scan.nextLine();

Address empAdr1 = **new** Address(empAddId, empAddLine, empCity, empState);

Department empDept1 = **new** Department(empDeptId, empDeptName, empDeptLoc);

empList.add(**new** Employee(empId, empFname, empLname, empSal, empDoj, empAdr1, empDept1));

} **else** {

System.***out***.println("Enter the date as specified");

}

}

**break**;

**case** 2:

System.***out***.println("a.ByEmployeeID");

System.***out***.println("b.ByFirstName");

System.***out***.println("c.ByLastName");

System.***out***.println("d.BySalary");

System.***out***.println("e.ByAddress");

System.***out***.println("f.ByDepartment");

System.***out***.println("Enter the choice");

scan.nextLine();

String input = scan.nextLine();

**switch** (input) {

**case** "a":

Collections.*sort*(empList, **new** ByEmployeeId());

**break**;

**case** "b":

Collections.*sort*(empList, **new** ByFirstName());

**break**;

**case** "c":

Collections.*sort*(empList, **new** ByLastName());

**break**;

**case** "d":

Collections.*sort*(empList, **new** BySalary());

**break**;

**case** "e":

Collections.*sort*(empList, **new** ByAddress());

**break**;

**case** "f":

Collections.*sort*(empList, **new** ByDepartment());

**break**;

}

**for** (Employee p : empList) {

System.***out***.println("-Employee ID- " + p.getEmployeeId() + "-First Name- " + p.getFirstName() + "-Last Name- " + p.getLastName() + "-Salary- " + p.getSalary() + "-Joining Date- "p.getJoinDate() + "-Address- " + p.getAddress().getAddressId()+ p.getAddress().getAddressLine1() + p.getAddress().getCity() + p.getAddress().getState()+ "-Department- " + p.getDepartment().getDepartmentId()+ p.getDepartment().getDepartmentName() + p.getDepartment().getLocation());

}

**break**;

**case** 3:

**return**;

}

}

}

}

**Output:**

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text

Description automatically generated

Text

Description automatically generated