

Assignment 3

Addison Douglas



November 12, 2022

Everett Community College

axdouglas@students.everettcc.edu

# Description

The assignment was to create an employee management system. Using an employee class that interacts with a a caller class. The employee class allows the creation of multiple employees in the system, and the caller class creates a interface for the user to interact with to create edit and remove employees.

# The Code (if more than one question, then specify the question and paste the code of every question)

/\*\*

\*

\*/

package cs141.axdouglas;

import java.util.ArrayList;

import java.util.Scanner;

/\*\*

\* @author axdouglas

\*

\*/

/\*\*

\* Name:Addison Douglas

\* Section: 2

\* Program Name: Caller

\* Description: The caller class presesnts the user with 6 options, using a switch statmnet going off the user input,

\* it calls a diffrent methods depedning on what the user chooses. The switch statnmnet is also in a while loop

\* so the user can contuine selecting options untill they are done. When they can then select exit to leave the

\* while loop

\*

\*/

public class Caller {

static ArrayList<Employee> allEmployee = new ArrayList<Employee>();

static ArrayList<Employee> microsoftEmployee = new ArrayList<Employee>();

static ArrayList<Employee> googleEmployee = new ArrayList<Employee>();

/\*\*

\* @param args

\*/

public static void main(String[] args) {

// TODO Auto-generated method stub

Scanner userInput = new Scanner(System.in);

int userNumber = 0;

boolean bool = true;

while(bool) {

System.out.println("Menu");

System.out.println("1. Add Employee");

System.out.println("2. Remove Employee");

System.out.println("3. Update Employee Information");

System.out.println("4. Print number of employees");

System.out.println("5. Print Employee Information");

System.out.println("6. Exit");

userNumber = userInput.nextInt();

switch(userNumber) {

case 1: addEmployee();

break;

case 2: removeEmployee();

break;

case 3: UpdateEmployeeInformation();

break;

case 4: printNumberOfEmployees();

break;

case 5: printEmployeeInformation();

break;

case 6: bool = false;

break;

default: System.out.println("error");

}

}

// userInput.close();

}

private static void addEmployee() {

boolean test = true;

int organaztionNumber;

Scanner userInput = new Scanner(System.in);

Scanner userInput2 = new Scanner(System.in);

String organaztion = "";

System.out.print("Enter employee name: ");

String name = userInput.nextLine();

System.out.print("Enter employee gender: ");

String gender = userInput.nextLine();

System.out.print("Enter employee job title: ");

String jobTitle = userInput.nextLine();

System.out.print("Enter employee birthday, format mm/dd/yy ");

String birthday = userInput.nextLine();

while(test) {

System.out.print("Please choose an organaztion, type 1 for Google 2 for Mircosoft: ");

organaztionNumber = userInput2.nextInt();

if(organaztionNumber == 1){

organaztion = "Google";

break;

}

else if(organaztionNumber == 2) {

organaztion = "Microsoft";

break;

}

}

Employee test1 = new Employee(name, gender, jobTitle, organaztion, birthday);

allEmployee.add(test1);

if(organaztion.equals("Google")) {

googleEmployee.add(test1);

}

else {

microsoftEmployee.add(test1);

}

// userInput.close();

// userInput2.close();

}

private static void removeEmployee() {

Scanner userInput = new Scanner(System.in);

int employeeNumber = 0;

System.out.println("Which Employee would you like to remove");

for(int i = 0; i < allEmployee.size(); i ++) {

System.out.println("[" + i + "]" + allEmployee.get(i).getName());

}

while (true) {

employeeNumber = userInput.nextInt();

if(employeeNumber > allEmployee.size()-1 || employeeNumber < 0 ) {

System.out.println("Invalid");

}

else {

break;

}

}

Employee.totalEmployee -= 1;

if(allEmployee.get(employeeNumber).getOrganazation().equals("Google")) {

Employee.googleCount -= 1;

}

else {

Employee.microsoftCount -= 1;

}

allEmployee.remove(employeeNumber);

// TODO Make it work to remove from total employee count

// userInput.close();

}

private static void UpdateEmployeeInformation() {

// TODO Auto-generated method stub

Scanner userInput = new Scanner(System.in);

int employeeNumber = 0;

boolean bool = true;

System.out.println("Which Employess info would you like to update?");

for(int i = 0; i < allEmployee.size(); i ++) {

System.out.println("[" + i + "]" + allEmployee.get(i).getName());

}

while (true) {

employeeNumber = userInput.nextInt();

if(employeeNumber > allEmployee.size()-1 || employeeNumber < 0 ) {

System.out.println("Invalid");

}

else {

break;

}

}

while(bool) {

Scanner userInputTwo = new Scanner(System.in);

int userNumber = 0;

System.out.println("What would you like to update");

System.out.println("1. Name");

System.out.println("2. job Title");

System.out.println("3. Workplace");

userNumber = userInput.nextInt();

switch(userNumber) {

case 1:

System.out.print("Enter new name: ");

allEmployee.get(employeeNumber).setName(userInputTwo.nextLine());

bool = false;

break;

case 2:

System.out.print("Enter new Job Title: ");

allEmployee.get(employeeNumber).setJobTitle(userInputTwo.nextLine());

bool = false;

break;

case 3:

System.out.print("Enter new work place: ");

if(allEmployee.get(employeeNumber).getOrganazation().equals("Google")) {

Employee.googleCount -= 1;

}

else {

Employee.microsoftCount -= 1;

}

String newWorkPlace = userInputTwo.nextLine();

allEmployee.get(employeeNumber).setOrganazation(newWorkPlace);

if(newWorkPlace.equals("Google")) {

Employee.googleCount += 1;

}

else {

Employee.microsoftCount += 1;

}

bool = false;

break;

default:

bool = true;

break;

}

}

}

private static void printNumberOfEmployees() {

System.out.println("Total employee count: " + Employee.getTotalEmployee() + " ");

System.out.print("Total google employee count: " + Employee.getGoogleCount()+ "\n");

System.out.print("Total microsoft employee count: " + Employee.getMicrosoftCount()+ " \n");

}

private static void printEmployeeInformation() {

Scanner userInput = new Scanner(System.in);

System.out.println("What would you like to print");

System.out.println("1. Information of single Employee");

System.out.println("2. All employee information");

System.out.println("3. Google Employee information");

System.out.println("4. Microsoft Employee information");

int userNumber = userInput.nextInt();

switch(userNumber) {

case 1:

System.out.println("Whos information do you want to print");

int employeeNumber = 0;

for(int i = 0; i < allEmployee.size(); i ++) {

System.out.println("[" + i + "]" + allEmployee.get(i).getName());

}

while (true) {

employeeNumber = userInput.nextInt();

if(employeeNumber > allEmployee.size()-1 || employeeNumber < 0 ) {

System.out.println("Invalid");

}

else {

break;

}

}

System.out.println(allEmployee.get(employeeNumber).toString());

break;

case 2: System.out.println(allEmployee.toString());

break;

case 3: System.out.println(googleEmployee.toString());

break;

case 4: System.out.println(microsoftEmployee.toString());

break;

}

// userInput.close();

}

}

/\*\*

\*

\*/

package cs141.axdouglas;

/\*\*

\* **@author** axdouglas

\*

\*/

/\*\*

\* Name:Addison Douglas

\* Section: 1

\* Program Name: Employee

\* Description: This is the employee class, in the construcor it takes in 5 varibles, one for name, gender, jobtitle, organization,

\* and birthday of the employee, the id is then randomly generted, and totalEmployee is increased, depending

\* on what organizatin the employee is in either microsoftCount or googleCount is inreased, beyond that there are setters

\* and getters for the varibles and a toString method to format the output of the employee class.

\*

\*/

public class Employee {

private int id;

private String name;

private String gender;

private String jobTitle;

private String Organazation;

private String birthday;

static int *microsoftCount*;

static int *googleCount*;

static int *totalEmployee*;

public Employee(String name, String gender, String jobTitle, String organazation, String birthday) {

*totalEmployee*++;

this.name = name;

this.gender = gender;

this.jobTitle = jobTitle;

this.Organazation = organazation;

this.birthday = birthday;

if(organazation.equalsIgnoreCase("Microsoft")) {

*microsoftCount*++;

}

else {

*googleCount*++;

}

id = (int) (Math.*random*() \* 1000000);

}

public String toString() {

return ("ID: " + id + "\n" + "Name: " + name + "\n" + "Gender: " + gender + "\n" + "Job title: " + jobTitle + "\n" + "Organiztion: " + Organazation + "\n" + "Birthday: " + birthday + "\n");

}

/\*\*

\* **@return** the totalEmployee

\*/

public static int getTotalEmployee() {

return *totalEmployee*;

}

/\*\*

\* **@param** totalEmployee the totalEmployee to set

\*/

public static void setTotalEmployee(int totalEmployee) {

Employee.*totalEmployee* = totalEmployee;

}

/\*\*

\* **@return** the name

\*/

/\*\*

\* **@return** the microsoftCount

\*/

public static int getMicrosoftCount() {

return *microsoftCount*;

}

/\*\*

\* **@return** the googleCount

\*/

public static int getGoogleCount() {

return *googleCount*;

}

public String getName() {

return name;

}

/\*\*

\* **@param** name the name to set

\*/

public void setName(String name) {

this.name = name;

}

/\*\*

\* **@return** the gender

\*/

public String getGender() {

return gender;

}

/\*\*

\* **@param** gender the gender to set

\*/

public void setGender(String gender) {

this.gender = gender;

}

/\*\*

\* **@return** the jobTitle

\*/

public String getJobTitle() {

return jobTitle;

}

/\*\*

\* **@param** jobTitle the jobTitle to set

\*/

public void setJobTitle(String jobTitle) {

this.jobTitle = jobTitle;

}

/\*\*

\* **@return** the organazation

\*/

public String getOrganazation() {

return Organazation;

}

/\*\*

\* **@param** organazation the organazation to set

\*/

public void setOrganazation(String organazation) {

Organazation = organazation;

}

}

# The output

Screenshots of your runs.

3 screenshots of 3 scenarios you tried in your program.

Text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

# Text Description automatically generated

# Comments/Notes (Extra Credit)

The comments or the notes section is if you wanted extra credit. This could be struggles you have come over while doing your program, or additions you wanted to highlight so I notice while grading…