Dylan Carlson

Lab Assignment 101

08/31/18

Code:

**Employee:**

package lab101;

/\*\*

\* This class, Employee, is the parent class of Hourly and Salaried.

\* It contains the similar instance variables between Hourly and Salaried, and getters and setters for those instances.

\* It also contains an equals and toString method.

\*

\* @version 08/29/2018

\* @author Dylan Carlson

\*/

public class Employee {

private int id;

private String name;

public Employee (int id, String name) {

this.id = id;

this.name = name;

}

//Setters

/\*\*

\* setId is passed an integer that is set to id.

\* @param newId

\*/

public void setId( int newId ){

id = newId;

}

/\*\*

\* setName is passed a String that is set to name.

\* @param newName

\*/

public void setName( String newName ){

name = newName;

}

//Getters

/\*\*

\* getId returns id to the client.

\* @return

\*/

public int getId() {

return id;

}

/\*\*

\* getName returns name to the client.

\* @return

\*/

public String getName(){

return name;

}

/\*\*

\* toString returns all the information in Employee as a String.

\* @return

\*/

public String toString(){

return getClass().getName() + ":" + id + ":" + name;

}

/\*\*

\* The equals method tests an object to see if it is equal to the Employee Class.

\* If it is true it passes true, and false if false.

\*

\* @param o

\* @return

\*/

public boolean equals( Object o){

if ( !( o instanceof Employee ))

return false;

Employee e = (Employee) o;

return id == e.id

&& name.equals( e.name );

}

}

**Hourly:**

package lab101;

/\*\*

\* This class, Hourly, is a sub-class of Employee.

\* It contains the instance variables unique to Hourly, and getters and setters for those instances.

\* It also contains a unique equals and toString method.

\*

\* @version 08/29/2018

\* @author Dylan Carlson

\*/

public class Hourly extends Employee {

private String position;

private double hourlyRate;

public Hourly (int id, String name, String position, double hourlyRate){

super(id, name);

this.position = position;

setHourlyRate( hourlyRate );

}

//Setters

/\*\*

\* setPosition is passed a string that is set to position.

\* @param newPosition

\*/

public void setPosition(String newPosition){

position = newPosition;

}

/\*\*

\* setHourlyRate is passed a double that is set to hourlyRate.

\* @param newHourlyRate

\*/

public void setHourlyRate( double newHourlyRate){

hourlyRate = newHourlyRate;

}

//Getters

/\*\*

\* getPosition returns position to the client.

\* @return

\*/

public String getPosition(){

return position;

}

/\*\*

\* getHourlyRate returns hourlyRate to the client.

\* @return

\*/

public double getHourlyRate(){

return hourlyRate;

}

/\*\*

\* toString returns all the information in Hourly joined with Employee as a String.

\* @return

\*/

public String toString(){

return super.toString() + getClass().getName() + ":" + position + ":" + hourlyRate;

}

/\*\*

\* The equals method tests an object to see if it is equal to the Hourly Class.

\* If it is true it passes true, and false if false.

\* @param o

\* @return

\*/

public boolean equals( Object o){

if ( !( o instanceof Hourly ))

return false;

else

{

Hourly e = (Hourly) o;

return super.equals(o)

&& hourlyRate==e.hourlyRate

&&position.equals( e.position );

}

}

}

**Salaried:**

package lab101;

/\*\*

\* This class, Salaried, is a sub-class of Employee.

\* It contains the instance variables unique to Salaried, and getters and setters for those instances.

\* It also contains a unique equals and toString method.

\*

\* @version 08/29/2018

\* @author Dylan Carlson

\*/

public class Salaried extends Employee {

private String title;

private int salary;

public Salaried (int id, String name, String title, int salary){

super(id, name);

this.title = title;

this.salary = salary;

}

//Setters

/\*\*

\* setTitle is passed a string that is set to title.

\* @param newTitle

\*/

public void setTitle( String newTitle ){

title = newTitle;

}

/\*\*

\* setSalary is passed an integer that is set to salary.

\* @param newSalary

\*/

public void setSalary( int newSalary){

salary = newSalary;

}

//Getters

/\*\*

\* getTtitle returns title to the client.

\* @return

\*/

public String getTitle(){

return title;

}

/\*\*

\* getSalary returns salary to the client.

\* @return

\*/

public int getSalary(){

return salary;

}

/\*\*

\* toString returns all the information in Salaried joined with Employee as a String.

\* @return

\*/

public String toString(){

return super.toString() + getClass().getName() + ":" + title + ":" + salary;

}

/\*\*

\* The equals method tests an object to see if it is equal to the Salaried Class.

\* If it is true it passes true, and false if false.

\* @param o

\* @return

\*/

public boolean equals( Object o){

if ( !( o instanceof Salaried ))

return false;

else

{

Salaried e = (Salaried) o;

return super.equals(o)

&& salary == e.salary

&& title.equals( e.title );

}

}

}

**Client:**

package lab101;

import java.util.Scanner;

/\*\*

\*This class, Client, uses Employee, Hourly, and Salaried to make an array of

\*employees with different information and is able to store it, print it, and

\* add a 10% raise to all the inputed Employees.

\*

\* @version 08/29/2018

\* @author Dylan Carlson

\*/

public class Client {

public static void main(String[] args){

Employee employeeList[];

employeeList = new Employee[10];

Scanner scan = new Scanner (System.in);

System.out.println("Enter an integer value for how many Employees you wish to enter:");

int numberOfEmployees = scan.nextInt();

//Looping how ever many Employees the user chose (Less than or equal to 10).

while ( numberOfEmployees > 0){

numberOfEmployees--;

System.out.println("\nIs the employee an hourly or salaried employee? (Enter 0 for Hourly and 1 for Salaried)");

int employeeType = scan.nextInt();

//This If statement determines the type of employee based on a 1 or 0.

if (employeeType == 0){

System.out.println("Enter their ID: ");

int e1 = scan.nextInt();

System.out.println("Enter their name: ");

scan.nextLine();

String e2 = scan.nextLine();

System.out.println("Enter their position: ");

String e3 = scan.nextLine();

System.out.println("Enter their hourly rate: ");

double e4 = scan.nextDouble();

Hourly h = new Hourly(e1, e2, e3, e4);

employeeList[numberOfEmployees] = h;

}

else if (employeeType == 1){

System.out.println("Enter their ID: ");

int e1 = scan.nextInt();

System.out.println("Enter their name: ");

scan.nextLine();

String e2 = scan.nextLine();

System.out.println("Enter their title: ");

String e3 = scan.nextLine();

System.out.println("Enter their Salary: ");

int e4 = scan.nextInt();

Salaried s = new Salaried(e1, e2, e3, e4);

employeeList[numberOfEmployees] = s;

}

else

System.out.println("ERROR");

}

System.out.println("");

//Printing out nicely all of the information of Employees

for(int i=9; i>=0; i--){

if ( employeeList[i] instanceof Hourly ){

Hourly h1 = (Hourly) employeeList[i];

System.out.println("\nEmployee ID: " + h1.getId());

System.out.println("Employee Name: " + h1.getName());

System.out.println("Employee Position : " + h1.getPosition());

System.out.println("Employee Hourly Rate: " + h1.getHourlyRate());

}

else if( employeeList[i] instanceof Salaried ){

Salaried s1 = (Salaried) employeeList[i];

System.out.println("\nEmployee ID: " + s1.getId());

System.out.println("Employee Name: " + s1.getName());

System.out.println("Employee Title: " + s1.getTitle());

System.out.println("Employee Salary: " + s1.getSalary());

}

else {

System.out.println("\nnull");

}

}

//Giving the Employees a 10% Raise

for(int i=9; i>=0; i--){

if ( employeeList[i] instanceof Hourly ){

Hourly h2 = (Hourly) employeeList[i];

h2.setHourlyRate(h2.getHourlyRate() \* 1.1);

}

else if( employeeList[i] instanceof Salaried ){

Salaried s2 = (Salaried) employeeList[i];

s2.setSalary( (int)(s2.getSalary() \* 1.1) );

}

}

System.out.println("\n Now Printing Employees with a 10% Raise: \n");

//Now printing all Employee Information with a 10% Raise

for(int i=9; i>=0; i--){

if ( employeeList[i] instanceof Hourly ){

Hourly h1 = (Hourly) employeeList[i];

System.out.println("\nEmployee ID: " + h1.getId());

System.out.println("Employee Name: " + h1.getName());

System.out.println("Employee Position : " + h1.getPosition());

System.out.println("Employee Hourly Rate: " + h1.getHourlyRate());

}

else if( employeeList[i] instanceof Salaried ){

Salaried s1 = (Salaried) employeeList[i];

System.out.println("\nEmployee ID: " + s1.getId());

System.out.println("Employee Name: " + s1.getName());

System.out.println("Employee Title: " + s1.getTitle());

System.out.println("Employee Salary: " + s1.getSalary());

}

}

//Testing equals method

//These are test examples of an Hourly, Salaried and Employee that will be tested if they are equal.

Hourly h1 = new Hourly(7, "Joe", "boss", 20.0);

Salaried s1 = new Salaried(8, "Dylan", "manager", 90000);

Employee e1 = new Employee(9, "Boy");

//Testing the equals method from Hourly

System.out.println("\nIs a Salaried Employee equal to an Hourly Employee: " + h1.equals(s1));

System.out.println("Is the Hourly constructor I created equal to itself: " + h1.equals(h1));

//Testing the equals method from Salaried

System.out.println("Is an Hourly Employee equal to a Salaried Employee: " + s1.equals(h1));

System.out.println("Is the Salaried constructor I created equal to itself: " + s1.equals(s1));

//Testing the equals method from Employee

System.out.println("Is a Salaried Employee equal to an Employee: " + e1.equals(s1));

System.out.println("Is the Employee constructor I created equal to itself: " + e1.equals(e1));

}

}

**OUTPUT OF PROGRAM:**

Enter an integer value for how many Employees you wish to enter:

7

Is the employee an hourly or salaried employee? (Enter 0 for Hourly and 1 for Salaried)

0

Enter their ID:

1

Enter their name:

bob

Enter their position:

boss

Enter their hourly rate:

20.0

Is the employee an hourly or salaried employee? (Enter 0 for Hourly and 1 for Salaried)

0

Enter their ID:

2

Enter their name:

rob

Enter their position:

coworker

Enter their hourly rate:

25.0

Is the employee an hourly or salaried employee? (Enter 0 for Hourly and 1 for Salaried)

1

Enter their ID:

3

Enter their name:

ron

Enter their title:

manager

Enter their Salary:

90000

Is the employee an hourly or salaried employee? (Enter 0 for Hourly and 1 for Salaried)

1

Enter their ID:

4

Enter their name:

joe

Enter their title:

leader

Enter their Salary:

100000

Is the employee an hourly or salaried employee? (Enter 0 for Hourly and 1 for Salaried)

0

Enter their ID:

5

Enter their name:

billy

Enter their position:

assistant

Enter their hourly rate:

15.5

Is the employee an hourly or salaried employee? (Enter 0 for Hourly and 1 for Salaried)

1

Enter their ID:

6

Enter their name:

nate

Enter their title:

ceo

Enter their Salary:

120000

Is the employee an hourly or salaried employee? (Enter 0 for Hourly and 1 for Salaried)

0

Enter their ID:

7

Enter their name:

john

Enter their position:

janitor

Enter their hourly rate:

12.0

null

null

null

Employee ID: 1

Employee Name: bob

Employee Position : boss

Employee Hourly Rate: 20.0

Employee ID: 2

Employee Name: rob

Employee Position : coworker

Employee Hourly Rate: 25.0

Employee ID: 3

Employee Name: ron

Employee Title: manager

Employee Salary: 90000

Employee ID: 4

Employee Name: joe

Employee Title: leader

Employee Salary: 100000

Employee ID: 5

Employee Name: billy

Employee Position : assistant

Employee Hourly Rate: 15.5

Employee ID: 6

Employee Name: nate

Employee Title: ceo

Employee Salary: 120000

Employee ID: 7

Employee Name: john

Employee Position : janitor

Employee Hourly Rate: 12.0

Now Printing Employees with a 10% Raise:

Employee ID: 1

Employee Name: bob

Employee Position : boss

Employee Hourly Rate: 22.0

Employee ID: 2

Employee Name: rob

Employee Position : coworker

Employee Hourly Rate: 27.500000000000004

Employee ID: 3

Employee Name: ron

Employee Title: manager

Employee Salary: 99000

Employee ID: 4

Employee Name: joe

Employee Title: leader

Employee Salary: 110000

Employee ID: 5

Employee Name: billy

Employee Position : assistant

Employee Hourly Rate: 17.05

Employee ID: 6

Employee Name: nate

Employee Title: ceo

Employee Salary: 132000

Employee ID: 7

Employee Name: john

Employee Position : janitor

Employee Hourly Rate: 13.200000000000001

Is a Salaried Employee equal to an Hourly Employee: false

Is the Hourly constructor I created equal to itself: true

Is an Hourly Employee equal to a Salaried Employee: false

Is the Salaried constructor I created equal to itself: true

Is a Salaried Employee equal to an Employee: false

Is the Employee constructor I created equal to itself: true

BUILD SUCCESSFUL (total time: 3 minutes 41 seconds)