/\*\*

\*

\* @author Samuel Swedberg

\* @version 9/2/2022

\*

\* A client that demonstrates polymorphism.

\*

\*/

public class Client {

public static void main(String[] args) {

// Creates an array to hold hourly and salaried employees

Employee[] employeeList = new Employee[10];

// Create an instance of Salaried and Hourly employee

Salaried salaried1 = new Salaried(0, "Al", "Manager", 60000);

Hourly hourly1 = new Hourly(1, "Kelly", "Hostess", 25.75);

Salaried salaried2 = new Salaried(2, "Peggy", "CEO", 120000);

Hourly hourly2 = new Hourly(3, "Bud", "Busboy", 15.00);

Hourly hourly3 = new Hourly(4, "Marcy", "Server", 10.00);

Hourly hourly4 = new Hourly(5, "Jefferson", "Cook", 35.00);

// Connects instance to employeeList array

employeeList[0] = salaried1;

employeeList[1] = hourly1;

employeeList[2] = salaried2;

employeeList[3] = hourly2;

employeeList[4] = hourly3;

employeeList[5] = hourly4;

// Prints employeeList array

for(int i=0; i<employeeList.length; i++)

{

if(employeeList[i]==null){ System.out.println("null");} // Prints "null" if there is no entry

else { System.out.println("Employee" + i + " " + employeeList[i].toString()); }

}

int i;

// Gives employees 25% raise

for(i=0; i<employeeList.length; i++)

{

if(employeeList[i] instanceof Salaried) // 25% raise for Salaried workers

{

Salaried salary = ( Salaried ) employeeList[i];

salary.setSalary((int)(salary.getSalary()\*1.25));

}

else if(employeeList[i] instanceof Hourly) // 25% raise for Hourly workers

{

Hourly rate = ( Hourly ) employeeList[i];

rate.setHourlyRate(rate.getHourlyRate()\*1.25);

}

}

// Prints employeeList array after 25% raise

for(i=0; i<employeeList.length; i++)

{

if(employeeList[i]==null){ continue; } // Skips null entries

else { System.out.println("Employee" + i + " rasied to " + employeeList[i].toString()); }

}

Salaried empSalaried = ( Salaried ) employeeList[0];

// Salaried equals() returns true

System.out.println("Is empSalaried == to employee 0?: " + empSalaried.equals(employeeList[0]));

// Salaried equals() returns false

System.out.println("Is empSalaried == to employee 1?: " + empSalaried.equals(employeeList[1]));

Hourly empHourly = ( Hourly ) employeeList[1];

// Hourly equals() returns false

System.out.println("Is empHourly == to employee 0?: " + empHourly.equals(employeeList[0]));

// Hourly equals() returns true

System.out.println("Is empHourly == to employee 0?: " + empHourly.equals(employeeList[1]));

}

}

/\*\*

\*

\* @author Samuel Swedberg

\* @version 9/2/2022

\*

\* The Employee class holds information about an employee

\*/

public class Employee {

private static int employeeCount = 0;

private int id;

private String name; // the employees name

/\*\*

\* Constructor for the Employee class

\* @param id

\* @param name

\*/

public Employee( int id, String name ) {

this.id = id;

this.name = name;

employeeCount++;

}

/\*\*

\*

\* @return id

\*/

public int getId( ) { return id; }

/\*\*

\*

\* @param id updates id instance variable

\*/

public void setId( int id ) { this.id = id; }

/\*\*

\*

\* @return name

\*/

public String getName() { return name; }

/\*\*

\*

\* @param name updates name instance variable

\*/

public void setName( String name ) { this.name = name; }

/\*\*

\*

\* @return number of time Employee constructor called

\*/

public int getEmployeeCount() { return employeeCount; }

/\*\*

\*

\* @return contents of Employee instance

\*/

public String toString(){

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

}

/\*\*

\*

\* @param o object to be compared

\* @return true if objects equal

\*/

public boolean equals( Object o )

{

if ( !( o instanceof Employee ) )

return false;

Employee e = ( Employee ) o;

return id == e.id

&& name.equals( e.name );

}

}

/\*\*

\*

\* @author Samuel Swedberg

\* @version 9/2/2022

\*

\* The Salaried class contains information related to a salaried worker

\*/

public class Salaried extends Employee {

private String title;

private int salary;

/\*\*

\*

\* @param id

\* @param name

\* @param title

\* @param salary

\*/

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

{

super( id, name );

this.title = title;

this.salary = salary;

}

/\*\*

\*

\* @return title

\*/

public String getTitle( ) { return title; }

/\*\*

\*

\* @param title updates title

\*/

public void setTitle( String title ) { this.title = title; }

/\*\*

\*

\* @return salary

\*/

public int getSalary( ) { return salary; }

/\*\*

\*

\* @param salary updates salary

\*/

public void setSalary( int salary ) { this.salary = salary; }

/\*\*

\*

\* @return contents of instance

\*/

public String toString()

{

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

}

/\*\*

\*

\* @param o

\* @return true if equal, false otherwise

\*/

public boolean equals( Object o )

{

if ( !( o instanceof Salaried ) )

return false;

Salaried s = ( Salaried ) o;

return super.equals( s )

&& title.equals( s.title )

&& salary == s.salary;

}

}

/\*\*

\*

\* @author Samuel Swedberg

\* @version 9/2/2022

\*

\* The Hourly class contains information related to an hourly worker

\*/

public class Hourly extends Employee {

private String position;

private double hourly;

/\*\*

\*

\* @param id

\* @param name

\* @param position

\* @param hourly

\*/

public Hourly( int id, String name, String position, double hourly )

{

super( id, name );

this.position = position;

this.hourly = hourly;

}

/\*\*

\*

\* @return position

\*/

public String getPosition( ) { return position; }

/\*\*

\*

\* @param position updates position

\*/

public void setPosition( String position ) { this.position = position; }

/\*\*

\*

\* @return hourly

\*/

public double getHourlyRate( ) { return hourly; }

/\*\*

\*

\* @param hourly updates hourly

\*/

public void setHourlyRate( double hourly ) { this.hourly = hourly; }

/\*\*

\*

\* @return contents of instance

\*/

public String toString()

{

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

}

/\*\*

\*

\* @param o

\* @return true if equal, false otherwise

\*/

public boolean equals( Object o )

{

if ( !( o instanceof Hourly ) )

return false;

Hourly h = ( Hourly ) o;

return super.equals( h )

&& position.equals( h.position )

&& hourly == h.hourly;

}

}

run:

Employee0 Salaried@0:Al:6:Salaried@Manager:60000

Employee1 Hourly@1:Kelly:6:Hourly@Hostess:25.75

Employee2 Salaried@2:Peggy:6:Salaried@CEO:120000

Employee3 Hourly@3:Bud:6:Hourly@Busboy:15.0

Employee4 Hourly@4:Marcy:6:Hourly@Server:10.0

Employee5 Hourly@5:Jefferson:6:Hourly@Cook:35.0

null

null

null

null

Employee0 rasied to Salaried@0:Al:6:Salaried@Manager:75000

Employee1 rasied to Hourly@1:Kelly:6:Hourly@Hostess:32.1875

Employee2 rasied to Salaried@2:Peggy:6:Salaried@CEO:150000

Employee3 rasied to Hourly@3:Bud:6:Hourly@Busboy:18.75

Employee4 rasied to Hourly@4:Marcy:6:Hourly@Server:12.5

Employee5 rasied to Hourly@5:Jefferson:6:Hourly@Cook:43.75

Is empSalaried == to employee 0?: true

Is empSalaried == to employee 1?: false

Is empHourly == to employee 0?: false

Is empHourly == to employee 0?: true

BUILD SUCCESSFUL (total time: 0 seconds)