# Praktiskais darbs Nr. 5.

# Temats: Mantošana. Metožu pardefinēšana.

Uzdevumi:

1) Izveidot klašu hierarhiju, realizējot kaskades veida mantošanu (1 pt).

Bāzes klasē Person jāizveido metodes:

2. Konstruktors ar parametriem, kas ir paredzēts objekta atrubūtu aizpildīšanai ar sākumvērtībām (1 pt).

3. Metode showPersonalDetails(), kas izprinte objekta atribūtu vērtības (1 pt). Atvasinātajā klasē Employee jāizveido metodes:

4.Konstruktors ar parametriem, kas ir paredzēts objekta atrubūtu aizpildīšanai ar sākumvērtībām un kas izsauc bāzes klases konstruktoru, lai nodotu tam argumentus (1 pt).

5.Kopijas konstruktors, kas ļauj kopēt objektus (1 pt).

6. Pārdefinēta metode showPersonalDetails(), kas izprinte objekta atribūtu vērtības (1 pt). Atvasinātajā klasē Manager jāizveido metodes:

7.Konstruktors ar parametriem, kas ir paredzēts objekta atrubūtu aizpildīšanai ar sākumvērtībām un kas izsauc bāzes klases konstruktoru, lai nodotu tam argumentus (1 pt).

8. Pārdefinēta metode showPersonalDetails(), kas izprinte objekta atribūtu vērtības (1 pt). Programmāi jāizveido klases Manager objekti un jādemonstrē darbs ar klasēm: lietotājs izveido masīvu, kurā tiek saglabāta informācija par menedžeriem, masīva aizpildīšanai jābūt ar ciklu (lietotājs ievada informāciju par 5 menedzēriem), viens menedžeri strādā divās nodaļās (jāizmanto kopijas konstruktors, lai kopētu informāciju). (2 pt).

Kods:

Klase “Person”

package JavaPackages;

public class Person {

    private String personName;

    private String address;

    public void setPersonName(String personName){

        this.personName = personName;

    }

    public void setAddress(String address){

        this.address = address;

    }

    public String getPersonName(){

        if(this.personName != null)

            return this.personName;

        else

            return null;

    }

    public String getAddress(){

        if(this.address != null)

            return this.address;

        else

            return null;

    }

    public Person(String personName, String address){

        this.setPersonName(personName);

        this.setAddress(address);

    }

    public void showPersonalDetails(){

        System.out.println(this.getPersonName() +

            " lives on " + this.getAddress());

    }

}

Klase “Employee”

Klase izmanto statisko mainīgu “nextId”, lai automātiski piešķirtu ID “Employee” klases objektiem un to atvasināto klašu objektiem.

package JavaPackages;

import JavaPackages.Person;

public class Employee extends Person{

    private String employeeID;

    private double salary;

    private static int nextId = 1;

    public void setSalary(double salary){

       this.salary = salary;

    }

    private void setEmployeeID(){

        this.employeeID = Integer.toString(nextId++);

    }

    private void copyEmployeeID(String employeeID){

        this.employeeID = employeeID;

    }

    public double getSalary(){

        return this.salary;

    }

    public String getEmployeeID(){

        if(this.employeeID != null)

            return this.employeeID;

        else

            return null;

    }

    public Employee(String personName, String address, double salary){

        super(personName, address);

        this.setSalary(salary);

        this.setEmployeeID();

    }

    public Employee(Employee employee){

        super(employee.getPersonName(), employee.getAddress());

        this.setSalary(employee.getSalary());

        this.copyEmployeeID(employee.getEmployeeID());

    }

    @Override

    public void showPersonDetails(){

        System.out.println(this.getPersonName() + " with " +

            this.getEmployeeID() + " ID has " +

            Double.toString(this.getSalary()) +

            " salary, lives on " + this.getAddress());

    }

}

Klase “Manager”

package JavaPackages;

import JavaPackages.Employee;

public class Manager extends Employee{

    private int departmentID;

    public void setDepartmentID(int departmentID){

        this.departmentID = departmentID;

    }

    public int getDepartmentID(){

        return this.departmentID;

    }

    public Manager(String personName, String address, double salary, int departmentID){

        super(personName, address, salary);

        this.setDepartmentID(departmentID);

    }

    public Manager(Manager manager){

        super((Employee) manager);

        this.setDepartmentID(manager.getDepartmentID());

    }

    @Override

    public void showPersonDetails(){

        System.out.println("Manager " + this.getPersonName() + " with "

            + this.getEmployeeID() + " ID " + " from " +

            this.getDepartmentID() + " department has " +

            Double.toString(this.getSalary()) +

            " salary, lives on " + this.getAddress());

    }

}

Fails “prakt5.java” (main)

import JavaPackages.Manager;

import java.util.ArrayList;

public class prakt5{

    public static void main(String[] args){

        ArrayList<Manager> managerList = new ArrayList<Manager>();

        for(int i = 1; i <= 4; i++){

            System.out.print("Type in " + Integer.toString(i) +

                " manager's name: ");

            String inputName = System.console().readLine();

            System.out.print("Type in " + Integer.toString(i) +

                " manager's address: ");

            String inputAddress = System.console().readLine();

            boolean isDouble = false;

            double resultSalary = 1.0D;

            do{

                System.out.print("Type in " + Integer.toString(i) +

                    " manager's salary: ");

                String inputSalary = System.console().readLine();

                try{

                    resultSalary = Double.parseDouble(inputSalary);

                    isDouble = true;

                }

                catch(Exception e){

                    isDouble = false;

                }

            }while(isDouble == false);

            boolean isInteger = false;

            int resultDepartmentID = 1;

            do{

                System.out.print("Type in " + Integer.toString(i) +

                    " manager's department ID: ");

                String inputDepartmentID = System.console().readLine();

                try{

                    resultDepartmentID = Integer.parseInt(inputDepartmentID);

                    isInteger = true;

                }

                catch(Exception e){

                    isInteger = false;

                }

            }while(isInteger == false);

            managerList.add(new Manager(inputName, inputAddress,

                resultSalary, resultDepartmentID));

        }

        int newDepartmentID = 15;

        boolean isInteger = false;

        do{

            System.out.print("Type in " + Integer.toString(5) +

                " manager's department ID: ");

            String inputDepartmentID = System.console().readLine();

            try{

                newDepartmentID = Integer.parseInt(inputDepartmentID);

                isInteger = true;

            }

            catch(Exception e){

                isInteger = false;

            }

        }while(isInteger == false);

        Manager managerToCopy = new Manager(managerList.get(3));

        managerToCopy.setDepartmentID(newDepartmentID);

        managerList.add(new Manager(managerToCopy));

        for(JavaPackages.Manager manager : managerList)

           manager.showPersonDetails();

    }

}

Rezultāts:

Text

Description automatically generated