A screenshot of a computer

AI-generated content may be incorrect.

1.Clasa Student are atribute private: nume, adresa, nrTelefon, email, IDstudent si mediaNotelor. Aceasta include metoda esteEligibilInscriere(), iar singurul getter public este getSeminariiAlese().

Clasa Inscriere contine atributul privat notePrimite, iar metodele getMediaLaZi() si getNotaFinala() sunt expuse public.

Clasa Seminar are atribute private: nume, IDseminar si taxe. Metodele addStudent() si dropStudent() sunt publice si permit gestionarea inscrierilor.

Clasa Profesor defineste atribute private pentru nume, adresa, nrTelefon, email si salar.

Intre Student si Inscriere, precum si intre Seminar si Student, exista o relatie de asociere unidirectionala.

Intre Seminar si Profesor exista o relatie de asociere bidirectionala.

Un Student poate avea mai multe inscrieri la diferite cursuri, iar un Seminar poate fi sustinut de unul sau mai multi profesori. De asemenea, un Seminar poate avea un numar variabil de studenti participanti, inclusiv niciunul.

2A diagram of a computer flow

AI-generated content may be incorrect.

import java.util.ArrayList;

import java.util.List;

abstract class ElementCarte {

private String titlu;

private int numar;

public ElementCarte(String titlu, int numar) {

this.titlu = titlu;

this.numar = numar;

}

public String getTitlu() {

return titlu;

}

public void setTitlu(String titlu) {

this.titlu = titlu;

}

public int getNumar() {

return numar;

}

public void setNumar(int numar) {

this.numar = numar;

}

}

class Sectiune extends ElementCarte {

public Sectiune(String titlu, int numar) {

super(titlu, numar);

}

}

class Capitol extends ElementCarte {

private String rezumat;

private List<Sectiune> sectiuni;

public Capitol(String titlu, int numar, String rezumat) {

super(titlu, numar);

this.rezumat = rezumat;

this.sectiuni = new ArrayList<>();

}

public String getRezumat() {

return rezumat;

}

public void setRezumat(String rezumat) {

this.rezumat = rezumat;

}

public void adaugaSectiune(Sectiune sectiune) {

sectiuni.add(sectiune);

}

}

class Parte extends ElementCarte {

private List<Capitol> capitole;

public Parte(String titlu, int numar) {

super(titlu, numar);

this.capitole = new ArrayList<>();

}

public void adaugaCapitol(Capitol capitol) {

capitole.add(capitol);

}

}

class Carte {

private String editor;

private String dataPublicare;

private String ISBN;

private List<Parte> parti;

public Carte(String editor, String dataPublicare, String ISBN) {

this.editor = editor;

this.dataPublicare = dataPublicare;

this.ISBN = ISBN;

this.parti = new ArrayList<>();

}

public void adaugaParte(Parte parte) {

parti.add(parte);

}

}

3. A screenshot of a computer

AI-generated content may be incorrect.

public class Persoana {  
 private String nume;  
 private List<Persoana> parinti;  
 private List<Persoana> copii;  
  
 public Persoana(String nume) {  
 this.nume = nume;  
 this.parinti = new ArrayList<>();  
 this.copii = new ArrayList<>();  
 }  
  
 public void adaugaParinte(Persoana parinte) {  
 if (!parinti.contains(parinte)) {  
 parinti.add(parinte);  
 parinte.adaugaCopil(this);  
 }  
 }  
  
 public void adaugaCopil(Persoana copil) {  
 if (!copii.contains(copil)) {  
 copii.add(copil);  
 copil.adaugaParinte(this);  
 }  
 }  
  
 public String getNume() {  
 return nume;  
 }  
  
 public List<Persoana> getParinti() {  
 return parinti;  
 }  
  
 public List<Persoana> getCopii() {  
 return copii;  
 }  
  
 @Override  
 public String toString() {  
 StringBuilder sb = new StringBuilder();  
 sb.append("Persoana: ").append(nume);  
 sb.append("\nPărinți: ");  
 for (Persoana p : parinti) {  
 sb.append(p.getNume()).append(" ");  
 }  
 sb.append("\nCopii: ");  
 for (Persoana c : copii) {  
 sb.append(c.getNume()).append(" ");  
 }  
 return sb.toString();  
 }  
  
}

public class Main {  
 public static void main(String[] args) {  
  
 Persoana ion = new Persoana("Ion");  
 Persoana maria = new Persoana("Maria");  
 Persoana vasile = new Persoana("Vasile");  
 Persoana elena = new Persoana("Elena");  
 Persoana gheorghe = new Persoana("Gheorghe");  
 Persoana ana = new Persoana("Ana");  
  
 ion.adaugaCopil(maria);  
 ion.adaugaCopil(vasile);  
 ion.adaugaCopil(elena);  
 ion.adaugaParinte(gheorghe);  
 ion.adaugaParinte(ana);  
  
 System.*out*.println(ion);  
  
  
 }  
}

4. A computer screen shot of a computer code

AI-generated content may be incorrect.

import java.util.\*;  
public class Angajat {  
  
 private String nume;  
 private String prenume;  
 private String dataNastere;  
 private String dataAngajare;  
  
 public Angajat(String nume, String prenume, String dataNastere, String dataAngajare) {  
 this.nume = nume;  
 this.prenume = prenume;  
 this.dataNastere = dataNastere;  
 this.dataAngajare = dataAngajare;  
 }  
  
 public String getNume() {  
 return nume;  
 }  
  
 public void setNume(String nume) {  
 this.nume = nume;  
 }  
  
 public String getPrenume() {  
 return prenume;  
 }  
  
 public void setPrenume(String prenume) {  
 this.prenume = prenume;  
 }  
  
 public String getDataNastere() {  
 return dataNastere;  
 }  
  
 public void setDataNastere(String dataNastere) {  
 this.dataNastere = dataNastere;  
 }  
  
 public String getDataAngajare() {  
 return dataAngajare;  
 }  
  
 public void setDataAngajare(String dataAngajare) {  
 this.dataAngajare = dataAngajare;  
 }  
  
 @Override  
 public String toString() {  
 return "Angajat{" +  
 "nume='" + nume + '\'' +  
 ", prenume='" + prenume + '\'' +  
 ", dataNastere='" + dataNastere + '\'' +  
 ", dataAngajare='" + dataAngajare + '\'' +  
 '}';  
 }  
  
}

import java.util.\*;  
public class Birou {  
 private Boolean esteSediulCentral;  
  
 public Birou(Boolean esteSediulCentral) {  
 this.esteSediulCentral = esteSediulCentral;  
 }  
  
 public Boolean getEsteSediulCentral() {  
 return esteSediulCentral;  
 }  
  
 public void setEsteSediulCentral(Boolean esteSediulCentral) {  
 this.esteSediulCentral = esteSediulCentral;  
 }  
  
 @Override  
 public String toString() {  
 return "Birou{" +  
 "esteSediulCentral=" + esteSediulCentral +  
 '}';  
 }  
  
}

import java.util.\*;  
  
public class Companie {  
  
 private String nume;  
 private Manager seful;  
 private String dataInfiintarii;  
 private List<Departament> departamente;  
  
 public Companie(String nume, Manager seful, String dataInfiintarii, List<Departament> departamente) {  
 this.nume = nume;  
 this.seful = seful;  
 this.dataInfiintarii = dataInfiintarii;  
 this.departamente = departamente;  
 }  
  
 public String getNume() {  
 return nume;  
 }  
  
 public void setNume(String nume) {  
 this.nume = nume;  
 }  
  
 public Manager getSeful() {  
 return seful;  
 }  
  
 public void setSeful(Manager seful) {  
 this.seful = seful;  
 }  
  
 public String getDataInfiintarii() {  
 return dataInfiintarii;  
 }  
  
 public void setDataInfiintarii(String dataInfiintarii) {  
 this.dataInfiintarii = dataInfiintarii;  
 }  
  
 public List<Departament> getDepartamente() {  
 return departamente;  
 }  
  
 public void setDepartamente(List<Departament> departamente) {  
 this.departamente = departamente;  
 }  
  
 @Override  
 public String toString() {  
 return "Companie{" +  
 "nume='" + nume + '\'' +  
 ", seful=" + seful +  
 ", dataInfiintarii='" + dataInfiintarii + '\'' +  
 ", departamente=" + departamente +  
 '}';  
 }  
}

import java.util.\*;  
public class Departament {  
  
 private String nume;  
 private Manager seful;  
 private List<Birou> birouri;  
 private List<Angajat> angajati;  
  
 public Departament(String nume, Manager seful, List<Birou> birouri, List<Angajat> angajati) {  
 this.nume = nume;  
 this.seful = seful;  
 this.birouri = birouri;  
 this.angajati = angajati;  
 }  
  
 public String getNume() {  
 return nume;  
 }  
  
 public void setNume(String nume) {  
 this.nume = nume;  
 }  
  
 public Manager getSeful() {  
 return seful;  
 }  
  
 public void setSeful(Manager seful) {  
 this.seful = seful;  
 }  
  
 public List<Birou> getBirouri() {  
 return birouri;  
 }  
  
 public void setBirouri(List<Birou> birouri) {  
 this.birouri = birouri;  
 }  
  
 public List<Angajat> getAngajati() {  
 return angajati;  
 }  
  
 public void setAngajati(List<Angajat> angajati) {  
 this.angajati = angajati;  
 }  
  
 @Override  
 public String toString() {  
 return "Departament{" +  
 "nume='" + nume + '\'' +  
 ", seful=" + seful +  
 ", birouri=" + birouri +  
 ", angajati=" + angajati +  
 '}';  
 }  
}

//TIP To <b>Run</b> code, press <shortcut actionId="Run"/> or  
// click the <icon src="AllIcons.Actions.Execute"/> icon in the gutter.  
import java.util.\*;  
  
public class Main {  
 public static void main(String[] args) {  
  
  
 Angajat angajat1 = new Angajat("Popescu", "Ion", "01.01.1990", "01.01.2010");  
 Angajat angajat2 = new Angajat("Ionescu", "Maria", "01.01.1995", "01.01.2015");  
 Angajat angajat3 = new Angajat("Georgescu", "Vasile", "01.01.1985", "01.01.2005");  
 Angajat angajat4 = new Angajat("Dumitrescu", "Andrei", "01.01.1992", "01.01.2012");  
 Angajat angajat5 = new Angajat("Constantinescu", "Mihai", "01.01.1993", "01.01.2013");  
 Angajat angajat6 = new Angajat("Stoica", "Cristina", "01.01.1994", "01.01.2014");  
 Angajat angajat7 = new Angajat("Dumitru", "Andreea", "01.01.1996", "01.01.2016");  
 Angajat angajat8 = new Angajat("Iancu", "Marian", "01.01.1997", "01.01.2017");  
 Angajat angajat9 = new Angajat("Munteanu", "Mihail", "01.01.1998", "01.01.2018");  
 Angajat angajat10 = new Angajat("Dinu", "Diana", "01.01.1999", "01.01.2019");  
  
 List<Angajat> echipaManager1 = new ArrayList<>();  
 echipaManager1.add(angajat4);  
 echipaManager1.add(angajat5);  
 echipaManager1.add(angajat6);  
  
 List<Angajat> echipaManager2 = new ArrayList<>();  
 echipaManager2.add(angajat7);  
 echipaManager2.add(angajat8);  
  
 List<Angajat> echipaManager3 = new ArrayList<>();  
 echipaManager3.add(angajat9);  
 echipaManager3.add(angajat10);  
  
  
  
 Manager manager1 = new Manager("Popescu", "Ion", "01.01.1990", "01.01.2010", echipaManager1, "01.01.2010");  
 Manager manager2 = new Manager("Ionescu", "Maria", "01.01.1995", "01.01.2015", echipaManager2, "01.01.2015");  
 Manager manager3 = new Manager("Georgescu", "Vasile", "01.01.1985", "01.01.2005", echipaManager3, "01.01.2005");  
  
  
 Birou birou1 = new Birou(true);  
 Birou birou2 = new Birou(false);  
 Birou birou3 = new Birou(false);  
  
  
 List<Birou> birouri1 = new ArrayList<>();  
 birouri1.add(birou1);  
 birouri1.add(birou2);  
  
 List<Birou> birouri2 = new ArrayList<>();  
 birouri2.add(birou3);  
  
 List<Angajat> angajati1 = new ArrayList<>();  
 angajati1.add(angajat1);  
 angajati1.add(angajat2);  
  
 List<Angajat> angajati2 = new ArrayList<>();  
 angajati2.add(angajat3);  
  
 Departament departament1 = new Departament("IT", manager1, birouri1, angajati1);  
 Departament departament2 = new Departament("HR", manager2, birouri2, angajati2);  
  
  
 List<Departament> departamente = new ArrayList<>();  
 departamente.add(departament1);  
 departamente.add(departament2);  
  
 Companie companie = new Companie("Companie1", manager3, "01.01.2000", departamente);  
  
 System.*out*.println(companie);  
 }  
}

import java.util.\*;  
  
public class Manager extends Angajat{  
 private List<Angajat> echipaManager;  
 private String dataDeCandEManager;  
  
 public Manager(String nume, String prenume, String dataNastere, String dataAngajare, List<Angajat> echipaManager, String dataDeCandEManager) {  
 super(nume, prenume, dataNastere, dataAngajare);  
 this.echipaManager = echipaManager;  
 this.dataDeCandEManager = dataDeCandEManager;  
 }  
  
 public List<Angajat> getEchipaManager() {  
 return echipaManager;  
 }  
  
 public void setEchipaManager(List<Angajat> echipaManager) {  
 this.echipaManager = echipaManager;  
 }  
  
 public String getDataDeCandEManager() {  
 return dataDeCandEManager;  
 }  
  
 public void setDataDeCandEManager(String dataDeCandEManager) {  
 this.dataDeCandEManager = dataDeCandEManager;  
 }  
  
 @Override  
 public String toString() {  
 return "Manager{" +  
 "nume: " + getNume() + ", " +  
 "prenume: " + getPrenume() + ", " +  
 "dataNastere: " + getDataNastere() + ", " +  
 "dataAngajare: " + getDataAngajare() + ", " +  
 "echipaManager: " + echipaManager + ", " +  
 "dataDeCandEManager: " + dataDeCandEManager +  
 '}';  
 }  
  
}

5. A diagram of a diagram

AI-generated content may be incorrect.

import java.util.List;

public class Retea {

private List<Pachet> pachete;

public List<Pachet> getPachete() {

return pachete;

}

public void setPachete(List<Pachet> pachete) {

this.pachete = pachete;

}

}

import java.util.List;

public class Pachet {

private List<Mesaj> mesaje;

public List<Mesaj> getMesaje() {

return mesaje;

}

public void setMesaje(List<Mesaj> mesaje) {

this.mesaje = mesaje;

}

}

public class Mesaj {

private String originea;

private String destinatia;

private String continutul;

public String getOriginea() {

return originea;

}

public void setOriginea(String originea) {

this.originea = originea;

}

public String getDestinatia() {

return destinatia;

}

public void setDestinatia(String destinatia) {

this.destinatia = destinatia;

}

public String getContinutul() {

return continutul;

}

public void setContinutul(String continutul) {

this.continutul = continutul;

}

}

public abstract class Nod {

private String nume;

private String referinta;

public String getNume() {

return nume;

}

public void setNume(String nume) {

this.nume = nume;

}

public String getReferinta() {

return referinta;

}

public void setReferinta(String referinta) {

this.referinta = referinta;

}

}

public abstract class Imprimanta extends Nod {

}

import java.util.List;

public class ImprimantaASCII extends Imprimanta {

private List<DocumentASCII> documente;

public List<DocumentASCII> getDocumente() {

return documente;

}

public void setDocumente(List<DocumentASCII> documente) {

this.documente = documente;

}

}

public class ImprimantaPostScript extends Imprimanta {

private List<DocumentPostScript> documente;

public List<DocumentPostScript> getDocumente() {

return documente;

}

public void setDocumente(List<DocumentPostScript> documente) {

this.documente = documente;

}

}

public abstract class Document {

}

public class DocumentASCII extends Document {

}

public class DocumentPostScript extends Document {

}

public class StatieDeLucru extends Nod {

}

public class Server extends Nod {

}