A diagram of a computer program

Description automatically generated

Ex2

A diagram of a diagram

Description automatically generated

import java.util.\*;

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 = new ArrayList<>();

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

super(titlu, numar);

this.rezumat = rezumat;

}

public String getRezumat() { return rezumat; }

public void setRezumat(String rezumat) { this.rezumat = rezumat; }

public void adaugaSectiune(Sectiune sectiune) { sectiuni.add(sectiune); }

public List<Sectiune> getSectiuni() { return sectiuni; }

}

class Parte extends ElementCarte {

private List<Capitol> capitole = new ArrayList<>();

public Parte(String titlu, int numar) {

super(titlu, numar);

}

public void adaugaCapitol(Capitol capitol) { capitole.add(capitol); }

public List<Capitol> getCapitole() { return capitole; }

}

class Carte {

private String editor;

private String dataPublicare;

private String ISBN;

private List<Parte> parti = new ArrayList<>();

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

this.editor = editor;

this.dataPublicare = dataPublicare;

this.ISBN = ISBN;

}

public void adaugaParte(Parte parte) { parti.add(parte); }

public List<Parte> getParti() { return parti; }

public String getEditor() { return editor; }

public void setEditor(String editor) { this.editor = editor; }

public String getDataPublicare() { return dataPublicare; }

public void setDataPublicare(String dataPublicare) { this.dataPublicare = dataPublicare; }

public String getISBN() { return ISBN; }

public void setISBN(String ISBN) { this.ISBN = ISBN; }

}

Ex 3

A diagram of a computer

Description automatically generated

import java.util.ArrayList;

import java.util.List;

class Persoana {

protected String nume;

protected int varsta;

protected String cnp;

public Persoana(String nume, int varsta, String cnp) {

this.nume = nume;

this.varsta = varsta;

this.cnp = cnp;

}

public String getNume() {

return nume;

}

}

class Parinte extends Persoana {

private List<Copil> copii;

public Parinte(String nume, int varsta, String cnp) {

super(nume, varsta, cnp);

this.copii = new ArrayList<>();

}

public void adaugaCopil(Copil copil) {

copii.add(copil);

copil.setParinte(this);

}

public void afiseazaCopii() {

System.out.print("Copiii lui " + nume + ": ");

if (copii.isEmpty()) {

System.out.println("Niciunul.");

} else {

for (Copil copil : copii) {

System.out.print(copil.getNume() + " ");

}

System.out.println();

}

}

}

class Copil extends Persoana {

private Parinte parinte;

public Copil(String nume, int varsta, String cnp) {

super(nume, varsta, cnp);

}

public void setParinte(Parinte parinte) {

this.parinte = parinte;

}

public void afiseazaParinte() {

if (parinte != null) {

System.out.println("Parintele lui " + nume + " este " + parinte.getNume());

} else {

System.out.println(nume + " nu are parinte specificat.");

}

}

}

public class Main {

public static void main(String[] args) {

Parinte ion = new Parinte("Ion", 40, "123456789");

Copil maria = new Copil("Maria", 10, "987654321");

ion.adaugaCopil(maria);

ion.afiseazaCopii();

maria.afiseazaParinte();

}

}

Ex4

A diagram of a company

Description automatically generated

import java.util.ArrayList;

import java.util.List;

class Angajat {

private String nume;

private String functie;

public Angajat(String nume, String functie) {

this.nume = nume;

this.functie = functie;

}

public String getNume() {

return nume;

}

public String getFunctie() {

return functie;

}

}

class Birou {

private String locatie;

private boolean sediuCentral;

public Birou(String locatie, boolean sediuCentral) {

this.locatie = locatie;

this.sediuCentral = sediuCentral;

}

public String getLocatie() {

return locatie;

}

public boolean isSediuCentral() {

return sediuCentral;

}

}

class Departament {

private String nume;

private Angajat manager;

private List<Angajat> angajati;

private List<Birou> birouri;

public Departament(String nume, Angajat manager) {

this.nume = nume;

this.manager = manager;

this.angajati = new ArrayList<>();

this.birouri = new ArrayList<>();

}

public void adaugaAngajat(Angajat angajat) {

angajati.add(angajat);

}

public void adaugaBirou(Birou birou) {

birouri.add(birou);

}

public void afiseazaStructura() {

System.out.println("Departament: " + nume);

System.out.println("Manager: " + manager.getNume());

System.out.print("Angajati: ");

for (Angajat angajat : angajati) {

System.out.print(angajat.getNume() + ", ");

}

System.out.println("\nBirouri:");

for (Birou birou : birouri) {

System.out.println("- " + birou.getLocatie() + (birou.isSediuCentral() ? " (Sediu Central)" : ""));

}

}

}

class Companie {

private String nume;

private List<Departament> departamente;

public Companie(String nume) {

this.nume = nume;

this.departamente = new ArrayList<>();

}

public void adaugaDepartament(Departament departament) {

departamente.add(departament);

}

public void afiseazaStructura() {

System.out.println("Compania: " + nume);

for (Departament d : departamente) {

d.afiseazaStructura();

System.out.println("-----------------------");

}

}

}

public class Main {

public static void main(String[] args) {

Angajat managerIT = new Angajat("Ion Popescu", "Manager IT");

Angajat angajat1 = new Angajat("Maria Ionescu", "Programator");

Angajat angajat2 = new Angajat("Vasile Georgescu", "Analist");

Departament it = new Departament("IT", managerIT);

it.adaugaAngajat(angajat1);

it.adaugaAngajat(angajat2);

Birou birou1 = new Birou("Bucuresti", true);

Birou birou2 = new Birou("Cluj", false);

it.adaugaBirou(birou1);

it.adaugaBirou(birou2);

Companie companie = new Companie("TechCorp");

companie.adaugaDepartament(it);

companie.afiseazaStructura();

}

}

A diagram of a computer

Description automatically generated

import java.util.List;

public class Rețea {

private String nume;

private List<Nod> listaNoduri;

public Rețea(String nume) {

this.nume = nume;

}

public String getNume() {

return nume;

}

public void adaugaNod(Nod nod) {

listaNoduri.add(nod);

}

}

public abstract class Nod {

private String nume;

private String tip;

private Nod urmatorNod;

public Nod(String nume, String tip) {

this.nume = nume;

this.tip = tip;

}

public String getNume() {

return nume;

}

public String getTip() {

return tip;

}

public Nod getUrmatorNod() {

return urmatorNod;

}

public void seteazaUrmatorNod(Nod urmator) {

this.urmatorNod = urmator;

}

public abstract void primestePachet(Pachet pachet);

}

public class StațieDeLucru extends Nod {

private String adresaIP;

public StațieDeLucru(String nume, String adresaIP) {

super(nume, "StațieDeLucru");

this.adresaIP = adresaIP;

}

@Override

public void primestePachet(Pachet pachet) {

}

}

public class Server extends Nod {

private int capacitateMemorie;

public Server(String nume, int capacitateMemorie) {

super(nume, "Server");

this.capacitateMemorie = capacitateMemorie;

}

@Override

public void primestePachet(Pachet pachet) {

}

}

public abstract class Imprimantă extends Nod {

public Imprimantă(String nume, String tip) {

super(nume, "Imprimantă");

}

public abstract void imprima(Pachet pachet);

}

public class ImprimantăASCII extends Imprimantă {

public ImprimantăASCII(String nume) {

super(nume, "ASCII");

}

@Override

public void imprima(Pachet pachet) {

}

@Override

public void primestePachet(Pachet pachet) {

}

}

public class ImprimantăPostScript extends Imprimantă {

public ImprimantăPostScript(String nume) {

super(nume, "PostScript");

}

@Override

public void imprima(Pachet pachet) {

}

@Override

public void primestePachet(Pachet pachet) {

}

}

public class Pachet {

private String origine;

private String destinatie;

private String continut;

public Pachet(String origine, String destinatie, String continut) {

this.origine = origine;

this.destinatie = destinatie;

this.continut = continut;

}

public String getOrigine() {

return origine;

}

public String getDestinatie() {

return destinatie;

}

public String getContinut() {

return continut;

}

}