

- II. Se consideră definită o clasă `Automobil` având datele membre `marca`, `model`, `capacitate` și `pret`. Clasa încapsulează constructori, metode de tip `set/get` pentru toate datele membre, precum și metodele `toString()`, `equals()` și `hashCode()`. Creați o listă care să conțină cel puțin 3 obiecte de tip `Automobil` și, folosind stream-uri bazate pe lista creată și lambda expresii, rezolvați următoarele cerințe:
- afișați automobilele care costă cel puțin 5000€, în ordinea descrescătoare a prețurilor;
 - afișați mărcile distincte de automobile;
 - creați o listă formată din automobilele care au capacitatea cilindrică cuprinsă între 2000 și 3000 cm³;
 - afișați prețul maxim al unui automobil marca "Audi".

```
public class Subiect_II {
    public static void main(String[] args) {
        LinkedList<Automobil> la = new LinkedList<>();

        la.add(new Automobil("BMW", "X5", 2000.5, 5000));
        la.add(new Automobil("Mercedes", "E Class", 3500, 6000));
        la.add(new Automobil("Audi", "A5", 1500, 3000));

        System.out.println("Lista initiala:");
        la.stream().forEach(System.out::println);
        System.out.println();

        System.out.println("Cerinta a)");
        la.stream().filter(a -> a.getPret() >=
5000).sorted(comparing(Automobil::getPret).reversed()).forEach(System.out::pr
intln);
        System.out.println();

        System.out.println("Cerinta b)");

        la.stream().map(Automobil::getMarca).distinct().forEach(System.out::println);
        System.out.println();

        System.out.println("Cerinta c)");
        List<Automobil> ln = la.stream().filter(a -> a.getCapacitate() >=
2000).filter(a -> a.getCapacitate() <= 3000).collect(toList());
        ln.stream().forEach(System.out::println);
        System.out.println();

        System.out.println("Cerinta d)");
        System.out.println(la.stream().filter(a ->
a.getMarca().equals("Audi")).max(comparing(Automobil::getPret)));
        System.out.println();
    }
}
```

- III. Scrieți o clasă Java care să calculeze de câte ori apare un cuvânt dat într-un fișier text, folosind un fir de executare. Scrieți un program care citește de la tastatură un cuvânt și, utilizând clasa definită anterior, afișează numărul total al aparițiilor cuvântului respectiv în fișierele text *exemplu_1.txt*, *exemplu_2.txt* și *exemplu_3.txt*. Cuvintele din fișierele text de intrare sunt despărțite între ele prin spații și semnele de punctuație uzuale.

```
class FirNumarareCuvinteFisier extends Thread {
    private String numeFisier;
    private String cuvantCautat;
    private int nrAparitii;

    public FirNumarareCuvinteFisier(String numeFisier, String cuvantCautat) {
        this.numeFisier = numeFisier;
        this.cuvantCautat = cuvantCautat;
        this.nrAparitii = 0;
    }

    public int getNrAparitii() {
        return nrAparitii;
    }

    @Override
    public void run() {
        try {
            Scanner in = new Scanner(new File(numeFisier));

            while(in.hasNextLine()) {
                String linie = in.nextLine();
                String []cuvinte = linie.split("[ .,\\n]+");
                for(int i = 0; i < cuvinte.length; i++)
                    if(cuvinte[i].equals(cuvantCautat))
                        nrAparitii++;
            }

            in.close();
        }
        catch (FileNotFoundException ex) {
            System.out.println("Eroare deschidere fisier!");
        }
    }
}

public class Subiect_III {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);

        System.out.println("Cuvantul cautat: ");
        String cuv = in.next();

        FirNumarareCuvinteFisier f_1 = new
        FirNumarareCuvinteFisier("exemplu_1.txt" , cuv);
        FirNumarareCuvinteFisier f_2 = new
        FirNumarareCuvinteFisier("exemplu_2.txt" , cuv);
        FirNumarareCuvinteFisier f_3 = new
        FirNumarareCuvinteFisier("exemplu_3.txt" , cuv);
    }
}
```

```
f_1.start();
f_2.start();
f_3.start();

try {
    f_1.join();
    f_2.join();
    f_3.join();
}
catch (InterruptedException ex) {
    System.out.println("Eroare fire de executare!");
}

int t = f_1.getNrAparitii() + f_2.getNrAparitii() +
f_3.getNrAparitii();
System.out.println("Cuvantul " + cuv + " apare de " + t + " ori in
fisierele date!");
}
}
```