

Ex Model 2020

clasele imutabile au toate membrele de tip final

metodele statice nu se suprascriu, doar se ascund

ABDEE

```
3. List<Diplomalicensita> list = new ArrayList();  
list.add(new Diplomalicensita("432", "Absolventi", 1990,  
"Facultate", "Specializare", 9.8);  
list.add(new Diplomalicensita("438", "A", 2001,  
"FMI", "Info", 9.4);  
list.add(new Diplomalicensita("434", "B", 2009,  
"Carol", "Med", 10);
```

```
list<Diplomalicensita> list1 =  
a) list.stream()
```

```
.filter(d -> d.getAn() >= 2000 && d.getAn() <= 2010)  
.sorted((d1, d2) -> d2.getMedie() - d1.getMedie())  
.toList();
```

```
for (Diplomalicensita d : list1)  
System.out.println(d);
```


b) List<String> res = list.stream()
 . filter(d -> d.getFacultate().equals
 ("Facultatea de informatică"))
 . filter(d -> ~~d~~ d.getAn() == 2018)
 . map(Diplomalicenta::getSpecializare)
 . distinct()
 . toList();

System.out.println(res);

c) List<String> res2 = list.stream()
 . filter(d -> d.getMedie() == 10.0)
 . map(~~As~~ Diplomalicenta::getAbsolut~~at~~)
 . toList();

System.out.println(res2);

long sau sunt de asta
 d) ~~and~~ res3 = list.stream()
 . filter(d -> d.getSpecializare().equals("Informatică"))

Informatică")

. count();

-2- System.out.println(res3);

4)

```

from
public class Main {
    public static void main(String[] args)

```

{

```

    Connection conn connection = DriverManager.getConnection(
        "jdbc:derby://localhost:1527/Diploma");

```

```

    Statement statement = conn connection.createStatement();

```

```

    Scanner scanner = new Scanner(System.in);

```

```

    int an_min = scanner.nextInt();

```

```

    int an_max = scanner.nextInt();

```

```

    String den_spec = scanner.next();

```

```

    ResultSet rs = statement.executeQuery(

```

```

        "Select * from Diplomaticenta #

```

```

        where an >= " + an_min + " and

```

```

        an <= " + an_max + " and

```

```

        list < Diplomaticenta > l = specializare like new ArrayList();

```

```

        while (rs.next()) {

```

```

            Diplomaticenta dipl = new Diplomaticenta();

```

```

            dipl.setAbsolvent(rs.getString("Absolvent"));

```

```

            etc pt restul comparator

```

```
if (!l.empty()) {  
    ObjectOutputStream s = new ObjectOutputStream(  
        new FileOutputStream("diploma.dat"));  
    s.writeObject(l);  
    s.close();  
}
```

```
else else {
```

```
    System.out.println("Error");  
}
```

```
}
```

```
}
```


(iii)

```
class FirNumarare extends Thread {  
    private final String numeFiror;  
    private List<String> s;  
    public List<String> getSC() {  
        return s;  
    }  
}
```

```
FileNumarare2 f2 = new FileNumarare("parere_2.txt");
```

```
f1.start();
```

```
f1.join();
```

```
f2.start();
```

```
f2.join();
```

```
List<String> l1 = f1.getS();
```

```
List<String> l2 = f2.getS();
```

```
for (String cur : l2)  
    l1.add(cur);
```

```
System.out.println(l.stream()
```

- filter(cur -> cur.length() >= 1)
- distinct()
- toList());

}

T
- 1. a)
2. b)
3. a)
4. b)
5. c)