```java
/*
    2 Сложное (по желанию): Допустим дан List<Account>.
    Класс Account определен так же как и в уроке 15:
        private String iban;
        private double balance;
        private Person owner;
        private MyDate openDate;

    Ваша задача реализовать следующий функционал:
        получить List<Account> всех счетов с балансом больше заданного числа
        получить List<Account> всех счетов заданного владельца
*/

import java.util.ArrayList;
import java.util.List;

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

        List<Account> accounts = List.of(
                new Account("DE0001", 1000.50, new Person("Jack", 20), new MyDate(10, 5, 2
                new Account("DE0002", 8732.55, new Person("John", 28), new MyDate(1, 3, 20
                new Account("DE0003", 7640.00, new Person("Bob", 23), new MyDate(19, 5, 20
                new Account("DE0004", 12001.00, new Person("Bob", 23), new MyDate(11, 2, 2
                new Account("DE0005", 123.00, new Person("Bob", 23), new MyDate(19, 5, 201
                new Account("DE0006", 3800.01, new Person("Tom", 10), new MyDate(2, 5, 202
                new Account("DE0007", 100.50, new Person("Alice", 16), new MyDate(6, 5, 20
                new Account("DE0008", 300012.00, new Person("Nick", 32), new MyDate(7, 5,
                );

        printAccounts(accounts);
        System.out.println("--------------------------");
        printAccounts(getAccountsWithMoreTHanGivenLimit(accounts, 1500));
        System.out.println("-----------------------");
        printAccounts(getAccountsByOwner(accounts, new Person("Bob", 23)));
        System.out.println("----------getAccountsByYear-----------");
```

```
37              printAccounts(getAccountsByOpenYear(accounts, 2024));
38              System.out.println("---------------getAccountsByOwnerAge----------------------");
39              printAccounts(getAccountsByOwnerAge(accounts, 20));
40          }
41          public  static void printAccounts(List<Account> accounts){
42              for (Account account: accounts){
43                  System.out.println(account);
44              }
45          }
46
47          public static List<Account> getAccountsWithMoreTHanGivenLimit (List<Account> list, dou
48              List<Account> result = new ArrayList<>();
49                                                          // Перебор, итерирование
50              for (Account account: list){
51                                                          // отбор элементов, условие, фильт
52                  if (account.getBalance()>limitBalance){
53                      result.add(account);                // действие
54                  }
55              }
56              return result;
57          }
58
59          public static List<Account> getAccountsByOwner (List<Account> list, Person owner){
60              List<Account> result = new ArrayList<>();
61              for (Account account: list){
62                  if (account.getOwner().equals(owner)){
63                      result.add(account);
64                  }
65              }
66              return result;
67          }
68
69          public static List<Account> getAccountsByOpenYear (List<Account> list, int year){
70              List<Account> result = new ArrayList<>();
71              for (Account account: list){
72                  if (account.getOpenDate().getYear() == year){
73                      result.add(account);
74                  }
75              }
76              return result;
77          }
78
79          public static List<Account> getAccountsByOwnerAge(List<Account> list, int age){
80              List<Account> result = new ArrayList<>();
81              for (Account account: list){
82                  if (account.getOwner().getAge() < age){
83                      result.add(account);
84                  }
85              }
86              return result;
87          }
88      }
```