

 Evgen959 / Advanced\_Backend Public[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#)[Advanced\\_Backend](#) / [Lesen 047](#) / [code](#) / [hw\\_na\\_les047\\_1](#) / [src](#) / [Main.java](#) 

Evgen959 newLes47

c6a189a · 3 minutes ago



37 lines (31 loc) · 1.67 KB

Code

Blame

Raw



```
1  /* 3. Дан список Person, написать метод который проверит есть ли в этом списке заданный Person
2
3  Дан список Person, написать метод который проверит есть ли в этом списке Person с заданным email*/
4
5
6  import java.util.List;
7  import java.util.Optional;
8
9  ✓ public class Main {
10  ✓   public static void main(String[] args) {
11       List<Person> people = List.of(
12           new Person("Jack", "jack@mail.com", 10),
13           new Person("Ann", "ann@mail.com",15),
14           new Person("Lena", "lena@mail.com",8),
15           new Person("Bil", "bil@mail.com",21),
16           new Person("John", "john@mail.com",13)
17       );
18       System.out.println(find(people, "lena@mail.com"));
19       // true
20       System.out.println(people);
21       // [Person{name=Jack, email='jack@mail.com', age=10}, Person{name=Ann, email='ann@mail.com',
22       // age=15}, Person{name=Lena, email='lena@mail.com', age=8}, Person{name=Bil,
23       // email='bil@mail.com', age=21}, Person{name=John, email='john@mail.com', age=13}]
24   }
25
26   public static boolean find(List<Person> list, Person findPerson){
27       return list.indexOf(findPerson) >= 0;
28   }
29
30  ✓   public static boolean find(List<Person> list, String email){
31       Optional<Person> find = list.stream()
32           .filter(p -> p != null && p.getEmail().equals(email))
33           .findAny();      // находим любой Person
34       return find.isPresent(); // boolean метод проверяет есть ли
35                           // в списке кокой либо Person
36   }
37   }
```