

Evgen959 / Advanced\_Backend Public

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#)[Advanced\\_Backend](#) / [Lesen 045](#) / [code](#) / [hw\\_na\\_les045\\_2](#) / [src](#) / [Main.java](#)

...

Evgen959 newLes45

eaa127e · 1 hour ago ↻

49 lines (42 loc) · 2.16 KB

Code Blame

Raw Copy Download Edit View

```
1  /* 2.
2  Дан список групп Cohort (String title, List<Student> students).
3  У каждого студента задан список контактных email-ов:
4  Student{String name, List<String> emails}
5  Руководство поставило задачу организовать рассылку email-ов.
6  Необходимо написать метод, который возвращает список всех емэйлов без повторов.*/
7
8
9  import java.util.ArrayList;
10 import java.util.Collections;
11 import java.util.List;
12 import java.util.Set;
13 import java.util.stream.Collectors;
14
15 public class Main {
16     public static void main(String[] args) {
17         List<Cohort> cohorts = List.of(new Cohort("cohort_1", List.of(
18             new Student("Jack", List.of("jack@mail.com", "jack12@gmx.de")),
19             new Student("Nick", List.of("nick@gmx.de")),
20             new Student("Ann", List.of("nick@gmx.de", "ann@gmx.de")),
21             new Student("Stan", List.of("stan@mail.com", "stan@gmx.de", "stan@Hotmail.com"))
22         )),
23             new Cohort("cohort_2", List.of(
24                 new Student("Ron", List.of("ron@mail.com")),
25                 new Student("Rick", Collections.emptyList()),
26                 new Student("Rick", null),
27                 new Student("Roman", List.of("roman@gmx.de", "roman@gmail.de"))
28             )),
29             new Cohort("cohort_1", List.of(
30                 new Student("Stan", List.of("stan@mail.com", "stan@gmx.de", "stan@Hotmail.com"))
31             )
32         );
33
34         List<String> emails = getEmails(cohorts);
35         System.out.println(emails);
36     }
37
38     public static List<String> getEmails(List<Cohort> list){
39         Set<String> emails = list.stream()
40             .filter(c->c.getStudents()!=null && !c.getStudents().isEmpty())
41             .flatMap(c -> c.getStudents().stream())
42             .filter(s->s.getEmails()!=null && !s.getEmails().isEmpty())
43             .flatMap(s -> s.getEmails().stream())
44             .collect(Collectors.toSet());
45         return new ArrayList<>(emails);
46     }
47
48 }
```