



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційні систем та технологій

Лабораторна робота №2
із дисципліни «*Основи програмування*»
Тема: «Колекції. Множина HashSet. Асоціативні масиви Map.»

Виконали:
Студенти групи ІА-24
(бригада №1)
Чайка А.П.
Коханчук М.М.
Котлярчук М.С.

Перевірив:
Колеснік Валерій
Миколайович

Хід роботи:

1. Ознайомитись з javadoc для наступних інтерфейсів, класів та методів:

- Set
- HashSet
- Object.equals(), Object.hashCode()
- Map
- HashMap

2. Виконати завдання лабораторної роботи №10, замінивши списки List (ArrayList та LinkedList) на множини Set (HashSet). Проаналізувати предметну область та на власний розсуд додати функціональність, для реалізації якої використати Map (TreeMap або HashMap).

3. Відповісти на контрольні питання.

Завдання:

Номер	Завдання
1	<p>Класи: Інститут (назва, список факультетів) Факультет (назва, список студентів) Студент (ім'я, прізвище, номер залікової книжки, середній бал)</p> <p>Задача: 1) Знайти загальну кількість студентів, що навчається в інституті 2) Знайти факультет, на якому навчається найбільша кількість студентів 3) Скласти список студентів, у яких середній бал в діапазоні 95..100</p>

Результати + код:

```
Number of students in institute:
6
The most popular faculty:
FICT
Students with average mark in range of 95-100
Kotlyarchuk Maxim
Yablonskiy Danya
Sidenko Daria
Chayka Anton
```

```

Main.java x Institute.java x Faculty.java x Student.java x
1      import java.util.*;
2
3      public class Main {
4      public static void main(String[] args) {
5          dataBase();
6          task1();
7          task2(Institute.getAllFaculties());
8          task3();
9      }
10
11      1 usage
12      static void task1() {
13          System.out.println("Number of students in institute:");
14          int allSt = 0;
15          for (Iterator i = Faculty.getAllStudents().iterator(); i.hasNext(); ) {
16              i.next();
17              allSt++;
18          }
19          System.out.println(allSt);
20      }
21
22      static void task2(HashMap<Integer, String> map) {
23          System.out.println("The most popular faculty:");
24          Integer val = 0;
25          HashSet<Integer> num = new HashSet<>();
26          for (Map.Entry<Integer, String> entryKey : Institute.getAllFaculties().entrySet()) {
27              num.add(entryKey.getKey());
28              for (Integer i : num) {
29                  if (val < i) {
30                      val = i;
31                  }
32              }
33          }
34          System.out.println(map.remove(val));
35      }
36      1 usage
37      static void task3() {
38          System.out.println("Students with average mark in range of 95-100");
39          HashSet<Student> stud = new HashSet<>();
40          for (Student i : Faculty.getAllStudents()) {
41              if (i.getAverageMark() >= 95 & i.getAverageMark() <= 100) {
42                  stud.add(i);
43              }
44          }
45          for (Student i : stud) {
46              System.out.println(i.fullName());
47          }
48      }

```

```

1 usage
46 public static void DataBase() {
47     Student student1 = new Student( name: "Anton", surname: "Chayka", number: 10001, averageMark: 96);
48     Student student2 = new Student( name: "Daria", surname: "Sidenko", number: 10002, averageMark: 95);
49     Student student3 = new Student( name: "Maxim", surname: "Kotlyarchuk", number: 10003, averageMark: 98);
50     Student student4 = new Student( name: "Anna", surname: "Orlovska", number: 10004, averageMark: 90);
51     Student student5 = new Student( name: "Yulia", surname: "Meleshko", number: 10005, averageMark: 89);
52     Student student6 = new Student( name: "Danya", surname: "Yablonskiy", number: 10006, averageMark: 99.5);
53
54     HashSet<Student> fpmSpace = new HashSet<>();
55     fpmSpace.add(student5);
56     fpmSpace.add(student6);
57
58     HashSet<Student> fictSpace = new HashSet<>();
59     fictSpace.add(student1);
60     fictSpace.add(student2);
61     fictSpace.add(student3);
62     fictSpace.add(student4);
63
64     Faculty.getAllStudents().addAll(fictSpace);
65     Faculty.getAllStudents().addAll(fpmSpace);
66
67     Institute.getAllFaculties().put(fpmSpace.size(), "FPM");
68     Institute.getAllFaculties().put(fictSpace.size(), "FICT");
69 }
70 }

```

```

Main.java x Institute.java x Faculty.java x Student.java x
1 import java.util.*;
2
3 public class Institute {
4     2 usages
5     String name;
6     1 usage
7     static HashMap<Integer, String> putFaculties = new HashMap<>();
8
9     public Institute(String name) { this.name = name; }
10
11     public String getName() { return name; }
12     4 usages
13
14     public static HashMap<Integer, String> getAllFaculties() { return putFaculties; }
15
16
17
18 }

```

```
Main.java × Institute.java × Faculty.java × Student.java ×
1  import java.util.*;
   4 usages
2  public class Faculty{
   1 usage
3      HashSet<Student> space;
   1 usage
4      Integer quantity;
   1 usage
5      static HashSet<Student> allStudents = new HashSet<>();
6
7      public Faculty(HashSet<Student> space, Integer quantity){
8          this.space = space;
9          this.quantity = quantity;
10     }
   4 usages
11     public static HashSet<Student> getAllStudents() { return allStudents; }
14 }
```

```
Main.java × Institute.java × Faculty.java × Student.java ×
1  import java.util.*;
   24 usages
2  public class Student {
   3 usages
3      private String name;
   3 usages
4      private String surname;
   5 usages
5      private int number;
   5 usages
6      private double averageMark;
   6 usages
7      public Student(String name, String surname, int number, double averageMark){
8          this.name = name;
9          this.surname = surname;
10         this.number = number;
11         this.averageMark = averageMark;
12     }
13 }
```

```

14      @Override
15      public boolean equals(Object obj) {
16          if (this == obj) {
17              return true;
18          }
19          if (obj == null) {
20              return false;
21          }
22          if (getClass() != obj.getClass()) {
23              return false;
24          }
25          final Student other = (Student) obj;
26          if (!Objects.equals(this.number, other.number)) {
27              return false;
28          }
29          if (!Objects.equals(this.averageMark, other.averageMark)) {
30              return false;
31          }
32          return true;
33      }
34
35      @Override
36      public int hashCode() {
37          int hash = 13;
38          hash = 31 * hash * Objects.hashCode(this.number);
39          hash = 31 * hash * Objects.hashCode((int) this.averageMark);
40          return hash;
41      }
42
43      public String getName() { return name; }
44
45      public String getSurname() { return surname; }
46
47      public int getNumber() { return number; }
48      2 usages
49
50      public double getAverageMark() { return averageMark; }
51      1 usage
52
53      public String fullName() { return surname + " " + name; }
54
55
56
57
58      }

```

Висновок:

Працюючи над цією роботою ми дізнались що таке хешування, ознайомились з класом HashSet, його методами – hashCode+equals(які потрібні для хешування об'єктів), асоціативним масивом HashMap та його методами. Узагальнили знання із створення об'єктів, конструкторів та методів у Java.