



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

Лабораторна робота № 11

Колекції. Множина TreeSet

Виконали
студенти групи ІА-23:
Архип'юк К. О.,
Богаченко А. В.,
Кашуб'як С. М.
Єрмак Д.Р.

Перевірів:
Колеснік В. М.

Київ 2023

Хід роботи:

Варіант 4

1. Ознайомитись з javadoc для наступних інтерфейсів та класів:
 - Set
 - TreeSet
 - Comparable
 - Comparator
 - SortedSet
 - NavigableSet
2. Виконати завдання попередньої лабораторної роботи, замінивши списки List (ArrayList та LinkedList) на множини Set (TreeSet). При цьому необхідно щонайменше один раз використати Comparable та щонайменше один раз – Comparator.

Main.java

```
import java.util.Iterator;
import java.util.Set;
import java.util.TreeSet;

public class Main {
    public static void main(String[] args) {
        VideoBlog blog = new VideoBlog("TA");
        Video video1 = new Video("Git tutorial",
            "https://youtu.be/fMSS6diyXQE", 56000, 8300, 500);
        Video video2 = new Video("Student life",
            "https://youtu.be/Ia1EgeQFEmY", 30000, 4800, 500);
        Video video3 = new Video("My plans for future",
            "https://youtu.be/dQw4w9WgXcQ", 48000, 10000, 23);
        VideoBlog.addVideo(video1);
        VideoBlog.addVideo(video2);
        VideoBlog.addVideo(video3);

        Comment com1 = new Comment("You have such a pretty dog!", 123, 15);
        Comment com2 = new Comment("I like it!", 9999, 0);
        Comment com3 = new Comment("Bruh official rickroll", 3, 87);
        video1.addComment(com1);
        video1.addComment(com2);
        video2.addComment(com3);
        video3.addComment(com1);
        video3.addComment(com2);
        video3.addComment(com3);

        System.out.println("Total views: " + task1());
        System.out.println("Is there any comment which has more likes than
it`s video: " + task2());
        System.out.print("Video which have the biggest number of dislike: ");
        task3();
    }

    /*1-типізований
    2-for-each
```

```

3-нетипізований*/

//1. Сумарна кількість переглядів усіх відео разом
public static int task1() {
    int sumviews = 0;
    int thisviews;
    Iterator<Video> i = VideoBlog.getVideos().iterator();
    while (i.hasNext()) {
        Video video = i.next();
        thisviews = video.getVidviews();
        sumviews += thisviews;
    }
    return sumviews;
}

//2. Чи є коментар, що набрав лайків більше, ніж відео, до якого він
дописаний
static boolean task2() {
    int videolikes;
    int commentlikes;
    for (Video video : VideoBlog.allvideos) {
        videolikes = video.getVidlikes();
        for (Comment comment : Video.comments) {
            commentlikes = comment.getCommentLikes();
            if (commentlikes > videolikes) return true;
        }
    }
    return false;
}

//3. Відео, що набрало найбільшу кількість дізлайків
public static void task3() {
    Set<Video> maxdislike = new TreeSet<>(new Video.VideoComparator());
    int dislike;
    int max = 0;
    for (Iterator i = VideoBlog.getVideos().iterator(); i.hasNext(); ) {
        Video video = (Video) i.next();
        dislike = video.getViddislikes();
        if (dislike > max) max = dislike;
    }
    for (Iterator i = VideoBlog.getVideos().iterator(); i.hasNext(); ) {
        Video video = (Video) i.next();
        dislike = video.getViddislikes();
        if ((dislike == max) & (max != 0)) {
            maxdislike.add(video);
        }
    }
    for (Video n : maxdislike) System.out.print(n.name() + "; ");
}
}

```

VideoBlog.java

```

import java.util.Set;
import java.util.TreeSet;

public class VideoBlog {
    private String nickname;
    static Set<Video> allvideos = new TreeSet<Video>(new
Video.VideoComparator());

    public static void addVideo(Video video) {

```

```

        allvideos.add(video);
    }

    public static Set<Video> getVideos() {
        return allvideos;
    }

    public VideoBlog(String nickname) {
        this.nickname = nickname;
    }
}

```

Video.java

```

import java.util.Comparator;
import java.util.Set;
import java.util.TreeSet;

public class Video {
    String name;
    String URL;
    int vidviews;
    int vidlikes;
    int viddislikes;
    static Set<Comment> comments = new TreeSet<>();

    public Video(String name, String URL, int vidviews, int vidlikes, int
viddislikes) {
        this.name = name;
        this.URL = URL;
        this.vidviews = vidviews;
        this.vidlikes = vidlikes;
        this.viddislikes = viddislikes;
    }

    static class VideoComparator implements Comparator<Video>{

        @Override
        public int compare(Video first, Video second) {
            int result = first.vidviews - second.vidviews;
            if(result != 0){
                return result;
            }
            return first.viddislikes - second.viddislikes;
        }
    }

    public int getVidviews() {
        return this.vidviews;
    }

    public int getVidlikes() {
        return this.vidlikes;
    }

    public int getViddislikes() {
        return this.viddislikes;
    }

    public static Set<Comment> getComments() {
        return comments;
    }
}

```

```

    public static void addComment(Comment comment) {
        comments.add(comment);
    }

    public String name() {
        return this.name;
    }
}

```

Comment.java

```

public class Comment implements Comparable<Comment>{
    String commentText;
    int commentLikes;
    int commentDislikes;

    public Comment(String commentText, int commentLikes, int commentDislikes)
    {
        this.commentText = commentText;
        this.commentLikes = commentLikes;
        this.commentDislikes = commentDislikes;
    }

    public int getCommentLikes() {
        return this.commentLikes;
    }

    @Override
    public int compareTo(Comment other) {
        int result = this.commentLikes - other.commentLikes;
        if (result != 0){
            return result;
        }
        return this.commentText.compareTo(other.commentText);
    }

    public String toString() {
        return "Коментар: " + commentText + " к-сть лайків: " + commentLikes
+ " к-сть дизлайків: " + commentDislikes;
    }
}

```

Result:

```

Total views: 134000
Is there any comment which has more likes than it's video: true
Video which have the biggest number of dislike: Student life; Git tutorial;

```

Висновок: Під час виконання лабораторної роботи ми продовжили ознайомлюватись з Collection Framework, використовували інтерфейс Set та його нащадків, застосували на практиці інтерфейси Comparable та Comparator.