



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 № 3**

Exceptions

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 класів:

- Throwable;
- Error;
- Exception;
- RuntimeException;

а також повторити конструкції try-catch-finally, throw, throws.

2. Проаналізувати предметну область з л/р 10-12. Створити свій власний тип checked exception (підклас класу Exception), що описує порушення одного з бізнес-правил предметної області (наприклад, спробу додати студента у групу, в якій вже навчається максимально допустима кількість студентів, або пуста "" назва фільму). Додати throws-декларацію цього exception до відповідних методів, у яких потрібно «викидати» його у разі порушення бізнес-правил. Додати блок try-catch-finally для демонстрації виклику цього метода. Продемонструвати «викидання» та обробку цього exception. Також додати та продемонструвати код для «викидання» кількох стандартних RuntimeException (IllegalArgumentException, NullPointerException, ...).

### Main.java

```
import java.util.HashMap;
import java.util.HashSet;
import java.util.Iterator;
import java.util.Set;

public class Main {
    public static void main(String[] args) {
        VideoBlog blog = new VideoBlog("Nanana");
        Video video1 = new Video("Git tutorial",
            "https://youtu.be/UQ8nXehmOro", 56_000, 8300, 500);
        Video video2 = new Video("Student life",
            "https://youtu.be/dQw4w9WgXcQ", 30_000, 4800, 500);
        Video video3 = new Video("My plans for future",
            "https://youtu.be/fMSS6diyXQE", 145_000, 10000, 23);
        Video video4 = new Video("Some title", "https://youtu.be/UydUK-
            adIQs", -2222, 31, 4);
        Video video5 = new Video("", "https://youtu.be/2fhs0S_MIws", 46, 6,
            0);

        try {
            VideoBlog.addVideo(video1);
            VideoBlog.addVideo(video2);
            VideoBlog.addVideo(video3);
            VideoBlog.addVideo(video4); // Exception
            VideoBlog.addVideo(video5); // Exception
        } catch (IllegalArgumentException e) {
            System.out.println("Exception! " + e.getMessage());
        }

        Comment com1 = new Comment("I like it!", 123, 15);
        Comment com2 = new Comment("You have such a pretty dog!", 9999, 0);
        Comment com3 = new Comment("You are so lazy. One video per month!
            Unsubscribe!!", 3, 87);
```

```

        Comment com4 = new Comment("The shimmering stars in the midnight sky
illuminated the darkened landscape below. " +
        "A lone wolf howled mournfully in the distance, while a soft
breeze whispered through the trees. " +
        "It was a moment of serene beauty, a fleeting glimpse of the
wonders of nature.", 32, 1);
        Comment com5 = new Comment(null, 2, 12);
        try {
            video1.addComment(com1);
            video1.addComment(com2);
            video2.addComment(com3);
            video2.addComment(com4); // Exception
            video3.addComment(com1);
            video3.addComment(com2);
            video3.addComment(com3);
            video3.addComment(com5); // Exception
        } catch (NullPointerException e) {
            System.out.println("Exception! " + e.getMessage());
        } catch (CommentLengthExceededException e) {
            System.out.println("Exception! " + e.getMessage());
        }
    }

    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();
    System.out.println();
    System.out.println("Videos with more than 50000 views: " + task4());
}

/*1-типізований
2-for-each
3-нетипізований*/

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

//Відео, що мають більше 50 000 переглядів (реалізація HashMap)
public static HashMap<String, String> task4() {
    HashMap<String, String> vidViewsStats = new HashMap<>();
    for (Iterator<Video> i = VideoBlog.getVideos().iterator();
i.hasNext(); ) {
        Video video = i.next();
        if (video.vidviews >= 50_000) {
            vidViewsStats.put(video.name, video.URL);
        }
    }
    return vidViewsStats;
}
}

```

## CommentLengthExceededException.java

```

public class CommentLengthExceededException extends Exception{
    public CommentLengthExceededException(String exceptionMessage){
        super(exceptionMessage);
    }
}

```

## VideoBlog.java

```

import java.util.HashSet;
import java.util.Set;

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

    public static void addVideo(Video video){
        if(video.name.isEmpty()){
            throw new IllegalArgumentException("Title cannot be empty");
        }
        if(video.vidviews < 0 || video.vidlikes < 0 || video.viddislikes <
0){
            throw new IllegalArgumentException("The number of views, likes or
dislikes cannot be less than 0");
        }
        allvideos.add(video);
    }

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

```

```

    }

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

```

## Video.java

```

import java.util.HashSet;
import java.util.Objects;
import java.util.Set;

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

    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;
    }

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

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

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

    public static void addComment(Comment comment) throws
CommentLengthExceededException{
        if (comment.commentText == null){
            throw new NullPointerException("Comment text cannot be null");
        }
        if (comment.commentText.length() > 100) {
            throw new CommentLengthExceededException("Comment exceeds the
maximum length of 100 characters");
        }
        comments.add(comment);
    }

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

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;

```

```

        Video video = (Video) o;
        return Objects.equals(name, video.name) && URL.equals(video.URL);
    }

    @Override
    public int hashCode() {
        return Objects.hash(name, URL);
    }
}

```

## Comment.java

```

public class 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;
    }
}

```

## Result

```

Exception! The number of views, likes or dislikes cannot be less than 0
Exception! Comment exceeds the maximum length of 100 characters
Exception! Title cannot be empty
Exception! Comment text cannot be null

```

**Висновок:** При виконанні лабораторної роботи ми ознайомились з класом Throwable та його нащадками; створили свій власний тип checked exception, навчилися як їх обробляти та як користуватися ключовими словами try, catch, finally для управління виключеннями.