

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ  
Національний аерокосмічний університет ім. М.Є. Жуковського  
“Харківський авіаційний інститут”

Кафедра комп’ютерних систем, мереж і кібербезпеки

## Лабораторна робота № 5

з дисципліни “Технології програмування”

# ОБРОБКА ВИКЛЮЧНИХ СИТУАЦІЙ. АБСТРАКТНІ КЛАСИ ТА ІНТЕРФЕЙСИ

ХАІ.503.525і1.22о.125, ПЗ

Виконав студент гр. 525і1 Проценко Д.І.  
(№ групи) (П.І.Б.)

---

(підпис, дата)

Перевірів ст. викладач каф.503  
(науковий ступінь, вчене звання, посада)

Здоровець Ю. В.

---

(підпис, дата)

(П.І.Б.)

Харків 2022

**Тема роботи:** Обробка виключних ситуацій. Абстрактні класи та інтерфейси

**Мета роботи:** Розгляд виняткових ситуацій та механізму їх обробки; набуття практичних навичок розробки програм мовою Java у середовищі IntelliJ IDEA, в яких передбачено обробку виключних ситуацій; набуття практичних навичок реалізувати абстрактні класи або інтерфейси, залучити до самостійної діяльності та прийняття рішень при формуванні програмного коду.

## Завдання 1.

### Частина 1. Умови завдання

#### ЗАВДАННЯ 1.

1. Написати програму мовою Java, в якій відповідно до заданого варіанту:

- визначити експериментально, помилки яких класів будуть генеровані;
- створити обробники виняткових ситуацій з використанням виявлених класів і всіх секцій конструкції обробника з відповідними повідомленнями, що дозволяють коректно виконати програму.

Завдання виконати у вигляді двох проектів: без використання власних методів і з використанням методів для кожної підзадачі, які можуть генерувати виключну ситуацію;

- провести unit test;
- зробити висновки.

Рис. 1 Умови завдання

#### Варіант 11

У програмі, де для квадратної матриці з елементами, введеними з клавіатури, потрібно вивести максимум серед парних чисел на головній діагоналі, можуть виникати помилки в наступних випадках:

- введення рядка замість числа; - немає парних чисел.
- передбачити інші виняткові ситуації, які генеруються програмою.

Рис. 2 Персональне завдання

### Частина 2. Текст програми

```
/*      Task1.java      */
import javax.lang.model.type.NullType;
import java.util.*;
import java.io.*;

public class Task1
{
    public Task1 ( )
    {
    }
}
```

```

    public void setMatr(int[][] matr)
    {
        System.out.println("Введите матрицу размером " + matr.length + "x" +
matr.length);
        Scanner s = new Scanner(System.in);
        outerloop: for(int i = 0; i < matr.length;i++)
        {
            for(int j = 0; j < matr.length;j++)
            {
                try {
                    matr[i][j] = s.nextInt();
                } catch (InputMismatchException e) {
                    System.err.println("Введен рядок вместо цифры");
                    break outerloop;
                }
            }
        }
    }
    public void getMatr(int[][] matr)
    {
        for(int i = 0; i < matr.length;i++)
        {
            for(int j = 0; j < matr.length;j++)
            {
                System.out.print( matr[i][j] );
            }
            System.out.println();
        }
    }
    public int AmountEven(int[][] matr) {
        int amount = 0;
        for(int i = 0; i < matr.length;i++)
        {
            if((matr[i][i] % 2 == 0) && (matr[i][i]!=0))
                amount++;
        }
        if (amount <= 0)
            throw new ArithmeticException("Нет четных в диагонали");
        return amount;
    }
}

```

```

/*      Main.java      */
import java.util.Arrays;
import java.util.InputMismatchException;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        int[][] matr;
        System.out.println("Введите размер матрицы:");
        int a = 0;
        do {
            try {
                a = s.nextInt();
            } catch (InputMismatchException e) {
                System.err.println("Введен рядок вместо цифры");
            }
            if (a <= 0)

```

```

        throw new IndexOutOfBoundsException("Размер матрицы не может быть
<= 0");
    }while(a<=0);
    matr = new int[a][a];
    for(int i = 0; i < a;i++)
    {
        for(int j = 0; j < a;j++)
        {
            matr[i][j] = 0;
        }
    }
    Task1 bob = new Task1();
    bob.setMatr(matr);
    bob.getMatr(matr);
    try {
        System.out.println("В матрице - " + bob.AmountEven(matr) + " четных");
    } catch (ArithmeticException e) {
        throw new RuntimeException(e);
    }
}

}

/*      MainTest.java      */

import java.util.Arrays;
import java.util.InputMismatchException;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        int[][] matr;
        System.out.println("Введите размер матрицы:");
        int a = 0;
        do {
            try {
                a = s.nextInt();
            } catch (InputMismatchException e) {
                System.err.println("Введен рядок вместо цифры");
            }
            if (a <= 0)
                throw new IndexOutOfBoundsException("Размер матрицы не может быть
<= 0");
        }while(a<=0);
        matr = new int[a][a];
        for(int i = 0; i < a;i++)
        {
            for(int j = 0; j < a;j++)
            {
                matr[i][j] = 0;
            }
        }
        Task1 bob = new Task1();
        bob.setMatr(matr);
        bob.getMatr(matr);
        try {
            System.out.println("В матрице - " + bob.AmountEven(matr) + " четных");
        } catch (ArithmeticException e) {
            throw new RuntimeException(e);
        }
    }
}

```

### Частина 3. Діаграма структури класів

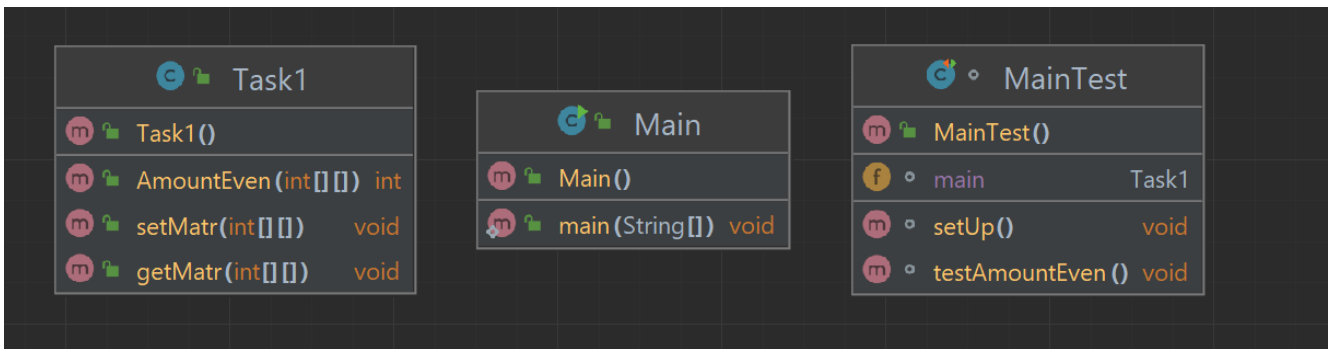


Рис. 3 Діаграма класів

### Частина 4. Скріншоти виконання завдання

```
Введите размер матрицы:
0
Exception in thread "main" java.lang.IndexOutOfBoundsException Create breakpoint : Размер матрицы не может быть <= 0
    at Main.main(Main.java:18)
Process finished with exit code 1
```

Рис. 4 Скріншот виконання завдання (1 частина)

```
Введите размер матрицы:
5
Введен рядок вместо цифры
Exception in thread "main" java.lang.IndexOutOfBoundsException Create breakpoint : Размер матрицы не может быть <= 0
    at Main.main(Main.java:18)
```

Рис. 5 Скріншот виконання завдання (2 частина)

```
Введите размер матрицы:
3
Введите матрицу размером 3x3
0 0 0
0 0 0
0 0 0
000
000
000
Exception in thread "main" java.lang.RuntimeException Create breakpoint : java.lang.ArithmeticException: Нет четных в диагонали
    at Main.main(Main.java:34)
Caused by: java.lang.ArithmeticException Create breakpoint : Нет четных в диагонали
    at Task1.AmountEven(Task1.java:49)
    at Main.main(Main.java:32)
```

Рис. 6 Скріншот виконання завдання (3 частина)

```
Введите размер матрицы:  
3  
Введите матрицу размером 3x3  
4 0 0  
5 1 3  
Введен рядок вместо цифры  
400  
000  
000  
В матрице - 1 четных
```

Рис. 7 Скріншот виконання завдання (4 частина)

```
Введите размер матрицы:  
3  
Введите матрицу размером 3x3  
0 1 2  
3 4 5  
6 7 8  
012  
345  
678  
В матрице - 2 четных
```

Рис. 8 Скріншот виконання завдання (5 частина)

```
@Test
void testAmountEven()
{
    int[][] x = new int[][]{
        { 0, 0, 0},
        { 1, 2, 7},
        {-1, 10, 1}
    };

    int expect = 2;
    int result = main.AmountEven(x);
}

✖ Tests failed: 1 of 1 test – 35 ms

"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" ...

org.opentest4j.AssertionFailedError: Результат не совпадает с действительностью ==>
Expected :2.0
Actual   :1.0
<Click to see difference>

<4 internal lines>
at MainTest.testAmountEven(MainTest.java:26) <29 internal lines>
at java.base/java.util.ArrayList.forEach(ArrayList.java:1511) <9 internal lines>
at java.base/java.util.ArrayList.forEach(ArrayList.java:1511) <27 internal lines>
```

Рис. 9 Скріншот модульного тестування (1 частина)

```
@Test
void testAmountEven()
{
    int[][] x = new int[][]{
        { 0, 0, 0},
        { 1, 2, 7},
        {-1, 10, 1}
    };

    int expect = 1;
    int result = main.AmountEven(x);
}

✔ Tests passed: 1 of 1 test – 24 ms

"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" ...

Process finished with exit code 0
```

Рис. 10 Скріншот модульного тестування (2 частина)

## Завдання 2.

### Частина 1. Умови завдання

#### ЗАВДАННЯ 2.

Розробити програму з використанням абстрактних класів та інтерфейсів відповідно до завдання.

- використовувати при розробці успадкування і поліморфізм;
- зобразити UML діаграму класів програми.
- зробити висновки.

Рис. 11 Умови завдання

11. interface Фільм ← abstract class Український фільм ← class Комедія.

Рис. 12 Персональне завдання

### Частина 2. Текст програми

```
/*      Film.java      */
public class Film {
    protected String Name;
    protected String Budget;

    public String getName() {
        return Name;
    }

    public void setName(String name) {
        Name = name;
    }

    public String getBudget() {
        return Budget;
    }

    public void setBudget(String budget) {
        Budget = budget;
    }
}

/*      UkrainianFilm.java      */
abstract class UkrainianFilm extends Film{
    protected String language;
    protected boolean Subtitles;

    @Override
    public abstract String getBudget();

    @Override
    public abstract void setBudget(String budget);

    protected abstract String getLanguage();

    protected abstract void setLanguage(String language);
}
```



```
    protected abstract boolean isSubtitles();

    protected abstract void setSubtitles(boolean subtitles);
}
```

```
/*      Comedy.java      */
```

```
class Comedy extends UkrainianFilm
{
    @Override
    public String getName() {
        return super.getName();
    }

    @Override
    public void setName(String name) {
        super.setName(name);
    }

    @Override
    public String getBudget() {
        return null;
    }

    @Override
    public void setBudget(String budget) {

    }

    private boolean IsFunny;
    @Override
    protected String getLanguage() {
        return null;
    }

    @Override
    protected void setLanguage(String language) {

    }

    @Override
    protected boolean isSubtitles() {
        return false;
    }

    @Override
    protected void setSubtitles(boolean subtitles) {

    }
}
```

### Частина 3. Діаграма структури класів

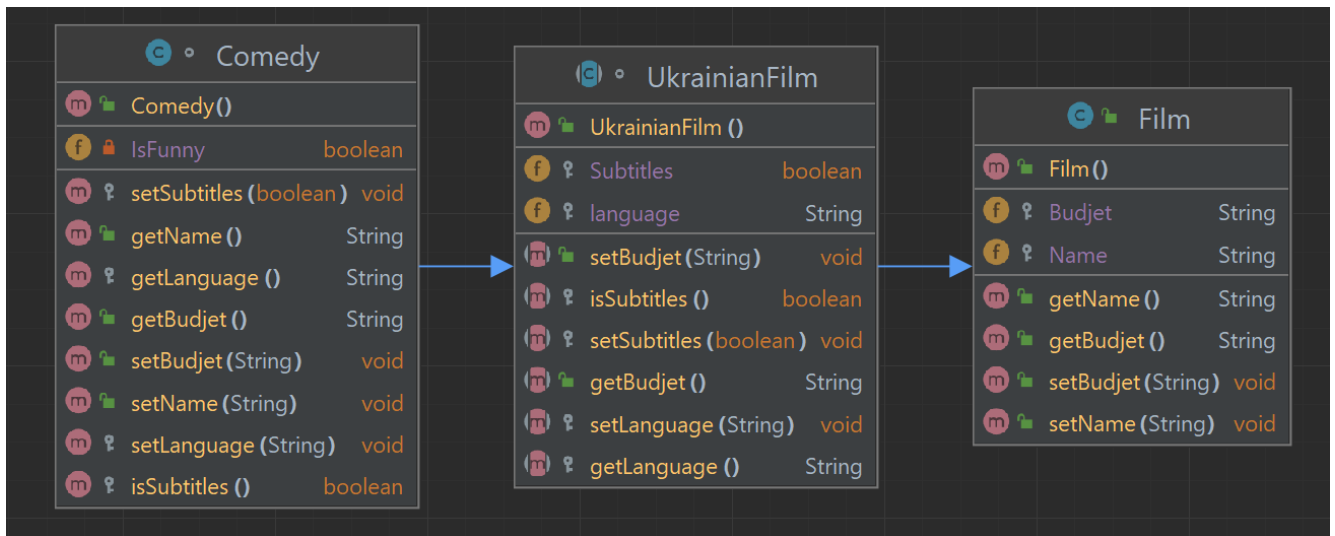


Рис. 13 Діаграма класів

### Висновки

Розглянув виняткові ситуації та механізми їх обробки; набув практичних навичок розробки програм мовою Java у середовищі IntelliJ IDEA, в яких передбачено обробку виключних ситуацій; набув практичних навичок реалізувати абстрактні класи або інтерфейси, залучився до самостійної діяльності та прийняття рішень при формуванні програмного коду.

### Використані джерела

1. Лекції з “Технології програмування” в
2. <https://www.youtube.com/watch?v=ZspkReG8L2E>