**ЛАБОРАТОРНА РОБОТА №2**

СПАДКУВАННЯ.

ОБРОБКА ВИКЛЮЧНИХ СИТУАЦІЙ

Варіант 18 (6)

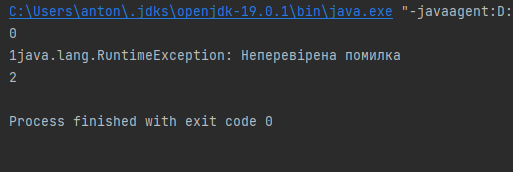
**Мета роботи**– знайомство з ієрархією класів винятків та здобуття навичок обробки помилок.

**Завдання 1**

Виконати приклади 2.1 - 2.14 лабораторної роботи, надавши письмово пояснення послідовності виконуваних команд.

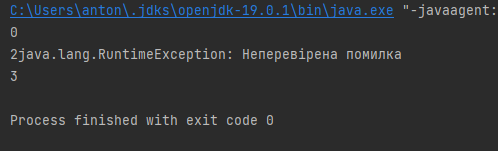
Приклад 2.1

public class Main {  
 public static void main(String[] args) {  
 try{  
 System.*out*.println("0");  
 throw new RuntimeException("Неперевірена помилка");  
 } catch (RuntimeException e) {  
 System.*out*.println("1" + e);  
 }  
 System.*out*.println("2");  
 }  
}



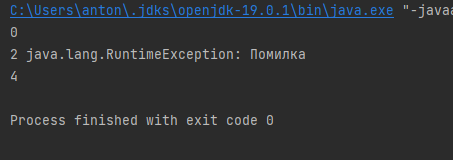
Приклад 2.2

public class Main {  
 public static void main(String[] args) {  
 try {  
 System.*out*.println("0");  
 throw new RuntimeException("Неперевірена помилка");  
 } catch (Exception e) {  
 System.*out*.println("2"+e);  
 }  
 System.*out*.println("3");  
 }  
}



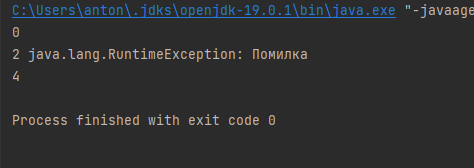
Приклад 2.3

public class Main {  
 public static void main(String[] args) {  
 try{  
 System.*out*.println("0");  
 throw new RuntimeException("Помилка");  
 }  
 catch (NullPointerException e){  
 System.*out*.println("1 " + e);  
 }  
 catch (RuntimeException e){  
 System.*out*.println("2 " + e);  
 }  
 catch (Exception e){  
 System.*out*.println("3 " + e);  
 }  
 System.*out*.println("4");  
 }  
}



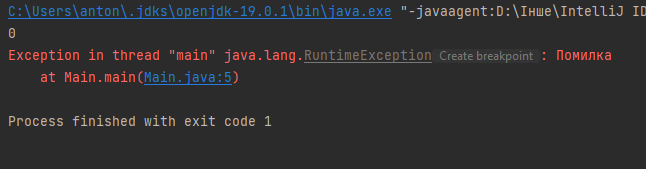
Приклад 2.4

public class Main {  
 public static void main(String[] args) {  
 try{  
 System.*out*.println("0");  
 throw new RuntimeException("Помилка");  
 }  
 catch (NullPointerException e){  
 System.*out*.println("1 " + e);  
 }  
 catch (Exception e){  
 System.*out*.println("2 " + e);  
 }  
 catch (Error e){  
 System.*out*.println("3 " + e);  
 }  
 System.*out*.println("4");  
 }  
}



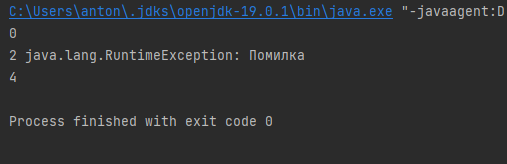
Приклад 2.5

public class Main {  
 public static void main(String[] args) {  
 try{  
 System.*out*.println("0");  
 throw new RuntimeException("Помилка");  
 }  
 catch (NullPointerException e){  
 System.*out*.println("1 " + e);  
 }  
 System.*out*.println("2");  
 }  
}



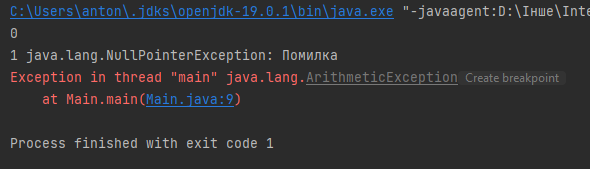
Приклад 2.6

public class Main {  
 public static void main(String[] args) {  
 try{  
 System.*out*.println("0");  
 throw new RuntimeException("Помилка");  
 }  
 catch (ArithmeticException e){  
 System.*out*.println("1 " + e);  
 }  
 catch (RuntimeException e){  
 System.*out*.println("2 " + e);  
 }  
 catch (Exception e){  
 System.*out*.println("3 " + e);  
 }  
 System.*out*.println("4");  
 }  
}



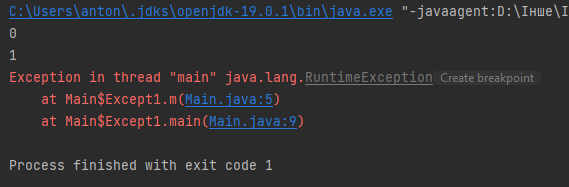
Приклад 2.7

public class Main {  
 public static void main(String[] args) {  
 try{  
 System.*out*.println("0");  
 throw new NullPointerException("Помилка");  
 }  
 catch (NullPointerException e){  
 System.*out*.println("1 " + e);  
 throw new ArithmeticException();  
 }  
 catch (ArithmeticException e){  
 System.*out*.println("2 " + e);  
 }  
  
 System.*out*.println("3");  
 }  
}



Приклад 2.8

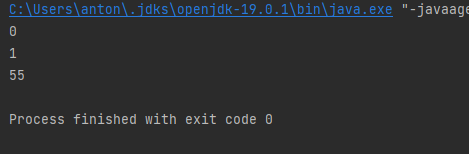
public class Main {  
 public static class Except1 { public static int m () {  
 try{  
 System.*out*.println("0");  
 throw new RuntimeException();  
 } finally {  
 System.*out*.println("1");  
 } }  
 public static void main(String[] args){ System.*out*.println(*m*());  
 }}  
}



Приклад 2.9

public class Except1 {  
 public static int m(){  
 try {  
 System.*out*.println("0");  
 return 55;  
 }  
 finally {  
 System.*out*.println("1");  
 }  
 }  
}

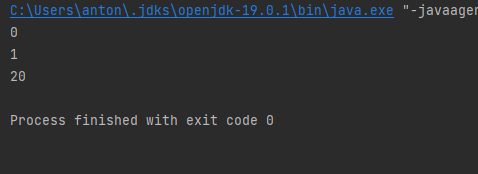
public class Program {  
 public static void main(String[] args) {  
 System.*out*.println(Except1.*m*());  
 }  
}



Приклад 2.10

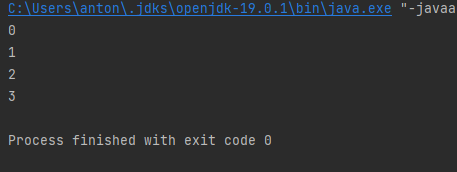
public class Except1 {  
 public static int m(){  
 try {  
 System.*out*.println("0");  
 return 15;  
 }  
 finally {  
 System.*out*.println("1");  
 return 20;  
 }  
 }  
}

public class Program {  
 public static void main(String[] args) {  
 System.*out*.println(Except1.*m*());  
 }  
}



Приклад 2.11

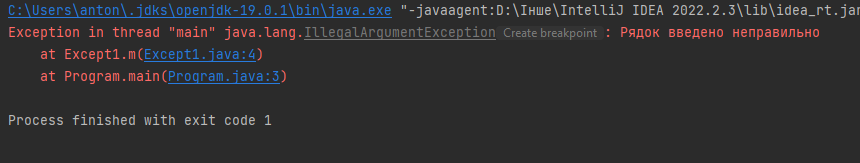
public class Program {  
 public static void main(String[] args) {  
 try {  
 System.*out*.println("0");  
 throw new NullPointerException("Помилка");  
 }  
 catch (NullPointerException e){  
 System.*out*.println("1");  
 }  
 finally {  
 System.*out*.println("2");  
 }  
 System.*out*.println("3");  
 }  
}



Приклад 2.12

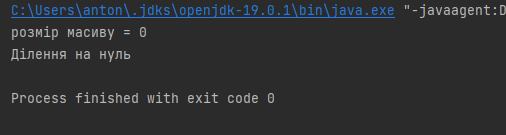
public class Except1 {  
 public static void m(String str, double chislo){  
 if(str == null){  
 throw new IllegalArgumentException("Рядок введено неправильно");  
 }  
 if(chislo > 0.001){  
 throw new IllegalArgumentException("Направильне число");  
 }  
 }  
}

public class Program {  
 public static void main(String[] args) {  
 Except1.*m*(null, 0.000001);  
 }  
}



Приклад 2.13

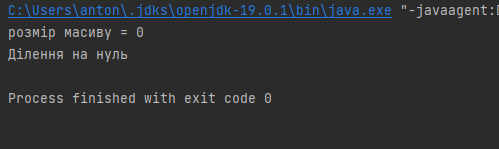
public class Program {  
 public static void main(String[] args) {  
 try {  
 int l = args.length;  
 System.*out*.println("розмір масиву = " + l);  
 int h = 10/l;  
 args[l + 1] = "10";  
 }  
 catch (ArithmeticException e){  
 System.*out*.println("Ділення на нуль");  
 }  
 catch (ArrayIndexOutOfBoundsException e){  
 System.*out*.println("Індекс не існує");  
 }  
 }  
}



Приклад 2.14

public class Except1 {  
 public static void m(int x) throws ArithmeticException{  
 int h = 10 / x;  
 }  
}

public class Program {  
 public static void main(String[] args) {  
 try {  
 int l = args.length;  
 System.*out*.println("розмір масиву = " + l);  
 Except1.*m*(l);  
 }  
 catch (ArithmeticException e){  
 System.*out*.println("Ділення на нуль");  
 }  
 }  
}



**Завдання 2.**

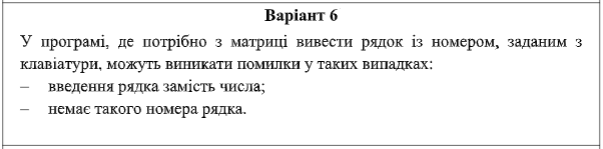
Для програми згідно з заданим варіантом (табл. 2.2):

* визначити експериментально, помилки яких класів буде згенеровано;
* створити обробники виняткових ситуацій з використанням виявлених класів та всіх секцій конструкції обробника з відповідними повідомленнями, що дозволяють коректно виконати програму.

Завдання виконати у вигляді двох проектів:

1) без використання власних методів

2) з використанням методів для кожної підзадачі, які можуть генерувати виняткову ситуацію.



Код програми:

import java.util.Scanner;  
public class Main {  
  
 public static boolean Check( String num)  
 {  
  
 if(num != null && num.matches("[-0-9.]+"))  
 {  
 float numb= Float.*parseFloat*(num);  
 }  
 else { throw new IllegalArgumentException("Ви ввели рядок");}  
 return true;  
 }  
  
 public static void OutMatrix(String [][]arr, int ryad, int stovb, int num)  
 {  
 if (num>ryad || num<=0)  
 {  
 throw new IllegalArgumentException("такого рядка не існує");  
 }  
 else {  
 System.*out*.println("\nРядок номер " + (num) + "\n");  
 for (int j = 0; j < stovb; j++) {  
 System.*out*.print(arr[(num - 1)][j] + " ");  
 }  
 }  
 }  
 public static String[][] DoMatrix(int ryad, int stovb)  
 {  
 Scanner sc=new Scanner(System.*in*, "UTF-8");  
 String[][] arr=new String[ryad][stovb];  
  
 for(int i=0; i<ryad; i++)  
 {  
 for(int j=0; j<stovb; j++)  
 {  
 System.*out*.println("Число ряд "+(i+1)+" стовб "+ (j+1)+":");  
 String num=sc.nextLine();  
 if(*Check*(num)==true){  
 arr[i][j]=num;  
 }  
 }  
 }  
 System.*out*.println("Введіть рядок матиці який хочете вивести:");  
 String num= sc.nextLine();  
 *Check*(num);  
 *OutMatrix*(arr, ryad, stovb, Integer.*parseInt*(num));  
  
 return arr;  
  
 }  
 public static void main(String[] args) {  
  
 Scanner sc =new Scanner(System.*in*, "UTF-8");  
 int ryad=2;  
 int stovb=2; //Квадтрана матриця має однакові кількість стовпців і рядків, визначимо як 4  
 System.*out*.println("Дана матриця"+ ryad+"х"+stovb+", заповніть її:");  
 *DoMatrix*(ryad, stovb);  
  
 }  
  
}

