

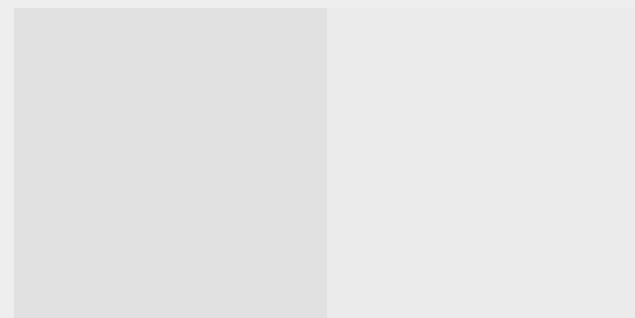
Praktikum **Algoritma dan Pemrograman II**

Exception Handling

5002221053 - **Putri Ghaida Tsuroyya**

5002221072 - **Mohammad Febryan Khamim**

Kelompok 5



Outline Pembahasan

- Definisi
- Lima Kata Kunci dalam Exception
- Tipe-tipe Exception



Definisi

Exception Handling

Exception Handling adalah sebuah mekanisme penanganan *error* atau kesalahan ketika melakukan running program. Mekanisme ini bermanfaat agar program tetap kembali berjalan / beroperasi atau tak berhenti secara paksa.

```
public class Main {  
    public static void main(String[] args) {  
        try {  
            int[] myNumbers = {1, 2, 3};  
            System.out.println(myNumbers[10]);  
        } catch (Exception e) {  
            System.out.println("Something went wrong.");  
        } finally {  
            System.out.println("The 'try catch' is finished.");  
        }  
    }  
}
```

Sumber : W3School

Gambaran Exception



Mengapa harus Exception?!

Mengapa harus Exception?!

Java Exception Handling sangat erat kaitannya dengan user experience (UX) karena dengan adanya mekanisme ini pengguna dapat dengan nyaman berinteraksi dengan sistem / fitur yang ada.

5 Block Code Kunci pada Exception

Try

Try adalah sebuah blok kode yang kemungkinan menjadi letak kesalahan atau error muncul dan ingin diproses.

Catch

Catch adalah suatu blok untuk menangkap kesalahan atau *error* yang terjadi dalam block **try** lalu melakukan penanganan kesalahan.

Finally

Finally adalah suatu *keyword* yang menunjukkan bahwa *block* program tersebut akan selalu dieksekusi kendatipun kesalahan muncul maupun tidak ada.

Throw

Throw adalah *keyword* yang berfungsi untuk **melemparkan kesalahan dengan sengaja** agar *block catch* dikerjakan atau dieksekusi oleh program.

Throws

Throws adalah *keyword Java* yang digunakan untuk memberi tahu bahwa dalam suatu **method** kemungkinan memiliki *error* atau kesalahan, tetapi **method** tersebut tak menangani ***error***-nya sendiri.

- ***Error*** akan “dilempar” ke kode yang memanggil **method** tersebut.

Throw vs Throws

Throw	Throws
Melempar exception secara manual dalam sebuah method	Mendeklarasikan bahwa method mungkin melemparkan exception
Lokasi di dalam tubuh <i>method</i> atau blok kode	Lokasi di dalam deklarasi <i>method</i> (bagian <i>signature</i>)
Dapat digunakan untuk exception secara manual, baik check maupun unchecked	Digunakan untuk checked exceptions
Tidak dapat melempar lebih dari satu exception secara langsung	Dapat mendeklarasikan lebih dari satu jenis exception

Throw

```
import java.util.Scanner;
public class throw_java {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Masukkan umur : ");
        int umur = input.nextInt();
        if (umur < 0) {
            throw new IllegalArgumentException("Umur tak boleh negatif!");
        }
        System.out.println("Umur : "+umur);
    }
}
```

Throws

```
2  import java.util.Scanner;
3  public class throws_java {
4      public static void main(String[] args) {
5          Scanner input = new Scanner(System.in);
6          System.out.print("Masukkan umur : ");
7          int umur = input.nextInt();
8          try {
9              validateAge(umur);
10             System.out.println("Umur : "+umur);
11         } catch (IllegalArgumentException e) {
12             System.out.println(e.getMessage());
13         }
14     }
15     public static void validateAge(int age) throws IllegalArgumentException{
16         if (age<0){
17             throw new IllegalArgumentException("Umur tak boleh negatif!");
18         }
19     }
20
21 }
```



Tipe-tipe Exception

Tipe-tipe Exception

Secara umum, exception di Java dibagi menjadi dua kategori utama:

- **Checked Exceptions:** Exception yang diperiksa pada waktu kompilasi. Contoh umum termasuk IOException, SQLException, dan ClassNotFoundException.
- **Unchecked Exceptions:** Exception yang terjadi pada waktu runtime dan tidak diperiksa selama kompilasi. Mereka adalah subclass dari RuntimeException. Contoh termasuk NullPointerException

<https://stackify.com/types-of-exceptions-java/>



```
import java.io.*;
public class ContohIOException {
    public static void main(String[] args) throws FileNotFoundException, IOException {
        String pathFile = "data_tidak_ada.txt";
        BufferedReader reader = null;

        try {
            reader = new BufferedReader(new FileReader( fileName:pathFile));
            String baris;

            while ((baris = reader.readLine()) != null) {
                System.out.println( x:baris);
            }

        } catch (IOException e) {
            System.out.println("Terjadi IOException: " + e.getMessage());
        } finally {
            try {
                if (reader != null)
                    reader.close(); // Menutup file
            } catch (IOException e) {
                System.out.println("Gagal menutup file: " + e.getMessage());
            }
        }
    }
}
```

IOException

ArrayIndexOutOfBoundsException

```
import java.util.Scanner;
public class ArrayIndexOutOfBoundsException {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int[] angka = {10, 20, 30, 40, 50};
        System.out.print("Masukkan indeks yang ingin diakses: ");
        int indeks = input.nextInt();

        try {
            System.out.println("Nilai pada indeks ke-" + indeks + " adalah: " + angka[indeks]);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Terjadi error: indeks " + indeks + " di luar batas array!");
        }

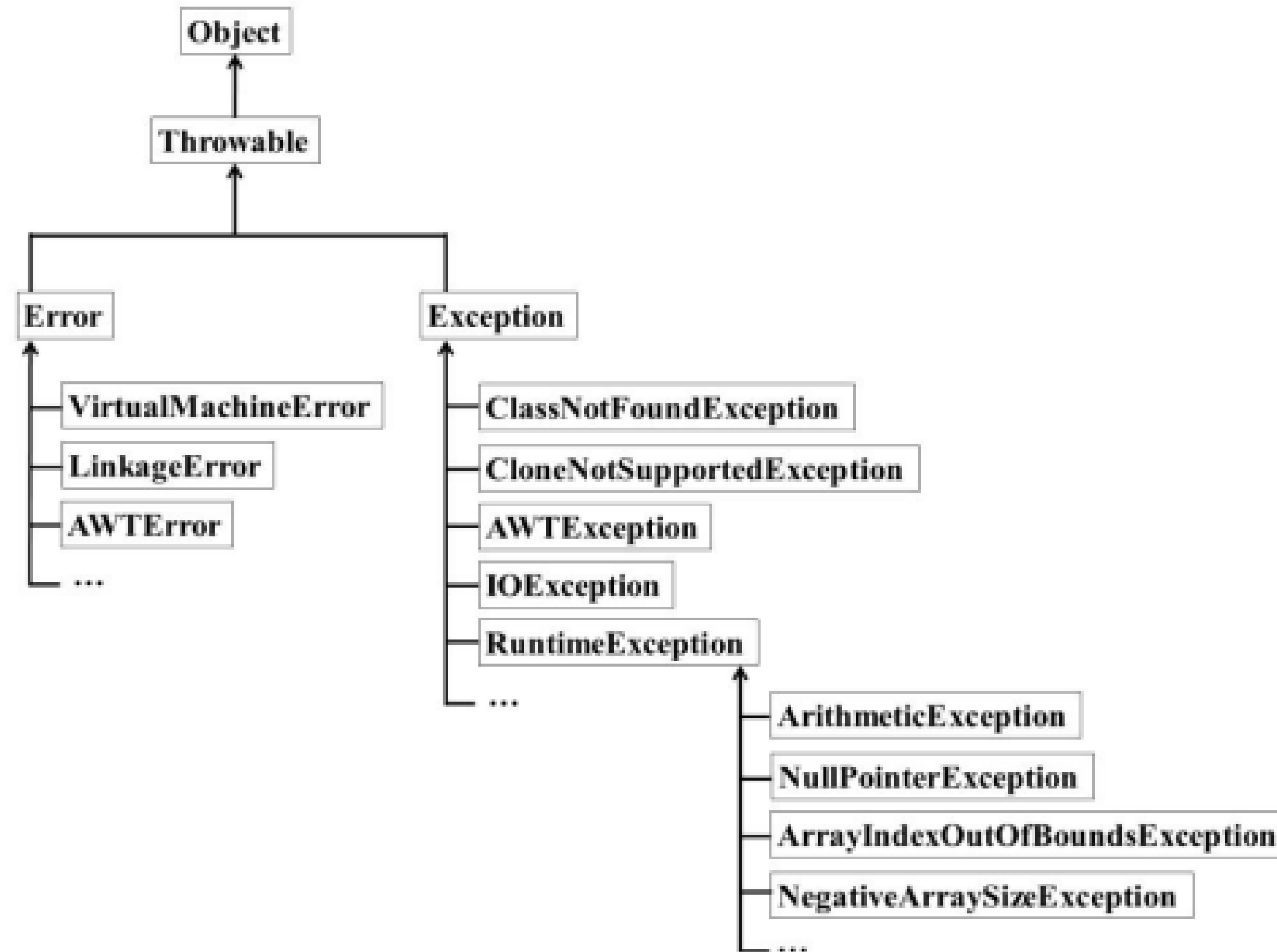
        input.close();
    }
}
```

NumberFormatException

```
public class NumberFormatException {  
    public static void main(String[] args) {  
        try {  
            // Mencoba mengubah teks bukan angka menjadi integer  
            String teks = "Halo123";  
            int angka = Integer.parseInt(s:teks); // ERROR di sini  
            System.out.println("Hasil konversi: " + angka);  
        } catch (Exception e) {  
            System.out.println(x: "Terjadi NumberFormatException: Teks tidak bisa dikonversi menjadi angka");  
        }  
    }  
}
```

ArithmeticException

```
public class ContohArithmeticException {  
    public static void main(String[] args) {  
        try {  
            int pembilang = 30;  
            int penyebut = 0;  
  
            int hasil = pembilang / penyebut;  
            System.out.println("Hasil: " + hasil);  
        } catch (ArithmeticException e) {  
            System.out.println("Terjadi ArithmeticException: Tidak bisa membagi dengan nol!");  
        }  
    }  
}
```



Hierarki dari Exception

Multiple Catch

Dalam Java, **Multiple catch blocks** untuk menangani berbagai jenis **Exception** yang mungkin terjadi selama program berjalan. Ini penting karena setiap jenis kesalahan (error) bisa disebabkan oleh hal yang berbeda.

Misalnya, kesalahan karena indeks array di luar batas (`ArrayIndexOutOfBoundsException`) berbeda dari kesalahan pembagian dengan nol (`ArithmeticException`).

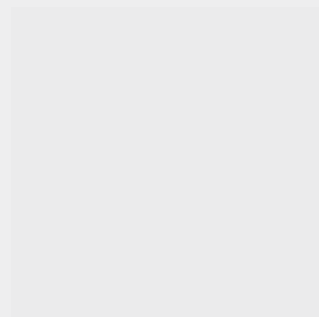
Multiple Catch

Aturan eksekusi **Multiple Catch**

- Hanya satu **catch** yang dieksekusi
- Pemilihan **catch** yang dieksekusi sesuai urutan prioritas :
Kesesuaian - Urutan peletakan



Mari Mencoba!





Latihan 1

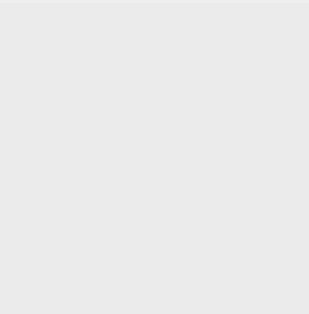
Buatlah program sederhana untuk menerapkan prinsip ***try***, ***catch***, dan ***finally*** pada Exception!

- ArithmeticException
- ArrayIndexOutOfBoundsException
- NullPointerException

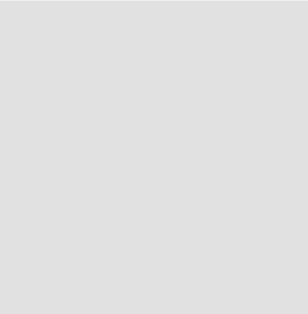
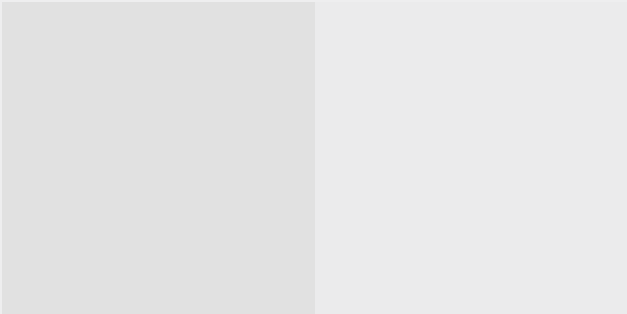
Latihan 2

```
public class catch_catch {  
    public static void main(String[] args) {  
        try {  
            int[] angka = {1, 2, 3};  
            int hasil = 10 / 0;  
            System.out.println(angka[3]);  
        }  
        catch (ArrayIndexOutOfBoundsException e) {  
            System.out.println("Indeks array salah!");  
        }  
        catch (ArithmeticException e) {  
            System.out.println("Pembagian dengan nol!");  
        }  
        finally {  
            System.out.println("Selesai");  
        }  
    }  
}
```

Sebutkan output dari program tersebut dan jelaskan algoritmanya!



QnA

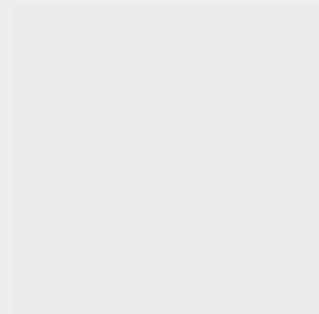


Departemen Matematika



Tugas

Kelompok 5





Kerjakan soal berikut!

intip.in/Pertemuan4Alprokom2

Pada soal nomor 1 kerjakan sesuai format laporan, sementara soal nomor 2 kerjakan tulis tangan tanpa format laporan.



Thank You

