Praktikum Pemrograman Berbasis Objek

Pertemuan 8

Review Time 🕑

1. Type Wrapper
2. Enum Class
3. Inner Class
4. Anonymous Class
5. UTS

Materi Pertemuan 8

Gambaran Materi Pembelajaran Hari Ini

Materi Pertemuan 8

01

Exception & Error Class

Mengenal Exception & Error beserta class yang ada

02

Exception Handling: Try-Catch

Mengenal cara Exception Handling menggunakan Try-Catch

03

Exception Handling: Throw & Throws

Mengenal cara melempar Exception menggunakan Throw & Throws

04

Custom Exception Class

Mengenal cara membuat Exception Class secara custom

01

Exception & Error Class

Exception & Error Class

Exception & **Error** merupakan sebuah **kondisi abnormal** yang terjadi saat menjalankan suatu program. Keduanya merupakan subclass dari **Throwable class**.

Perbedaannya adalah :

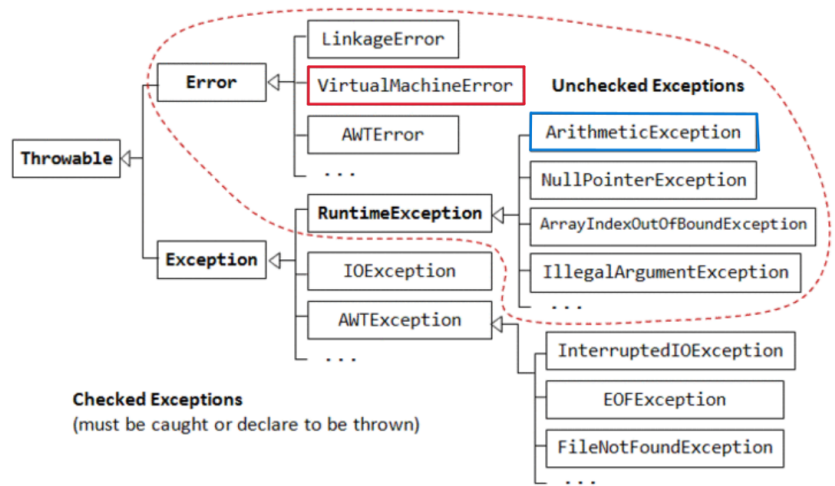
- **Error** diakibatkan oleh **lack of system** (kecacatan sistem) sehingga kita sulit untuk mendeteksi maupun menangani error tersebut.
- **Exception** diakibatkan oleh **kesalahan kode** yg ditulis oleh **developer**. Exception mudah ditangani dan dideteksi.

Exception & Error Class

Berikut adalah contoh perbedaan lebih lanjut dari Error dan Exception:

VirtualMachineError mengindikasikan **VM Java** yang **rusak** dan **kekurangan sumber daya** untuk beroperasi. Error seperti ini sebaiknya **tidak ditangani** atau **di-handle**.

ArithmeticException mengindikasikan adanya **kondisi aritmatika** yang **diluar penalaran**. **Exception** seperti ini sebaiknya **ditangani** atau **di-handle** sehingga program dapat terus beroperasi.



Exception Class

Merupakan class berbentuk **Throwable** yang mengindikasikan suatu **kondisi abnormal** yang terjadi saat menjalankan program yang diakibatkan oleh kesalahan developer.

Exception dibagi menjadi 2 :

- Checked Exception

Merupakan exception yang **dicek saat compile time**. Jika tidak ditangani, maka akan menimbulkan error ketika pertama kali dijalankan.

- Unchecked Exception

Merupakan exception yang **tidak dicek saat compile time**. Jika tidak ditangani, maka tidak akan menimbulkan error ketika pertama kali dijalankan.

Contoh **Checked** Exception

```java
import java.io.*;

class CheckedException {
    public static void main(String[] args){
        /*
        * File 'C:\\test\\a.txt' tidak ada.
        * Maka program tidak akan bisa dijalankan.
        * Program akan mengeluarkan FileNotFoundException
        * yang mana merupakan child dari IOException
        * IOException merupakan Checked exception
        */
        FileReader file = new FileReader("C:\\test\\a.txt"
        BufferedReader fileInput = new BufferedReader(file

        for(int counter = 0; counter < 3; counter++){
            System.out.println(fileInput.readLine());
        }

        fileInput.close();
    }
}
```

```
./CheckedException.java:11: error:
unreported exception FileNotFoundException;
must be caught or declared to be thrown
    FileReader file = new FileReader("C:\\test\\a.txt");
                      ^
```

Contoh **Unchecked** Exception

```java
import java.util.Scanner;

class UnheckedException {
    public static void main(String[] args){
        Scanner input = new Scanner(System.in);

        int x = 0;
        int y = 10;

        System.out.println("ENTER untuk lanjutkan");
        input.nextLine();

        int z = y / x;
        // ArithmeticException.
        // Suatu bilangan tidak boleh dibagi dengan 0

        System.out.println("Hasil z = " + z);
    }
}
```

```
ENTER untuk lanjutkan

Exception in thread "main"
java.lang.ArithmeticException: / by zero
    at UnheckedException.main(Main.java:13)
```

02

Exception Handling: Try-Catch

Try-Catch

Try-catch merupakan salah satu cara **penanganan class exception** yang muncul akibat kondisi abnormal. Statement ini terdiri dari **blok try** dan **blok catch**.

Blok try berisikan **kode** yang berkemungkinan **memunculkan suatu exception**. Kode dalam blok ini akan dieksekusi **hingga muncul suatu exception**.

Block catch berisikan **kode** yang dieksekusi apabila **muncul suatu exception** dalam blok try sebelumnya. Blok catch memiliki **parameter class exception** yang **ingin ditangani.**

```
try {
  //  Block of code to try

} catch(Exception e) {
  //  Block of code to handle errors


}
```

Contoh Try-Catch

```
import java.io.*;

class CheckedExceptionHandling {
    public static void main(String[] args){
        try{
            FileReader file = new FileReader("C:\\test\\a.txt");
            BufferedReader fileInput = new BufferedReader(file);

            for(int counter = 0; counter < 3; counter++){
                System.out.println(fileInput.readLine());
            }

            fileInput.close();
        } catch(IOException e) {
            System.out.println("Terjadi suatu error : " + e.getMessage());
        }
    }
}
```

```
Terjadi suatu error : C:\test\a.txt (No such file or directory)
```

03

Exception Handling: Throw & Throws

Throw & Throws

Throw merupakan statement yang digunakan untuk **melemparkan** suatu **exception** dalam blok kode. Hal ini berguna untuk **melempar custom exception**.

Throws merupakan **tanda** untuk suatu **method** yang dapat **melemparkan** suatu **exception** didalamnya. Hal ini berguna untuk **tidak menangani exception** dalam method itu sendiri namun dalam kode yang memanggil **method** tersebut.

Contoh Throw & Throws

```
import java.io.*;

class Throws {
    public static void bacaFile() throws FileNotFoundException {
        throw new FileNotFoundException("File tidak ditemukan");
    }

    public static void main(String[] args){
        try{
            bacaFile();
        } catch(IOException e) {
            System.out.println("Terjadi suatu error : " + e.getMessage());
        }
    }
}
```

```
Terjadi suatu error : File tidak ditemukan
```

04

Custom Exception Class

Contoh Custom Exception Class

Membuat **custom exception class** dapat dilakukan dengan membuat class seperti biasa lalu **mewarisi class Exception** dengan menggunakan **extends**.

Tujuannya adalah apabila ada kondisi abnormal lainnya yang ingin dibuat.

```
Terjadi error :
Angka yang dimasukkan melanggar ketent
```

```java
class InvalidNumberException extends Exception {
    public InvalidNumberException(){
        super("Angka yang dimasukkan melanggar ketentuan");
    }
}

public class CustomException {
    public static boolean cekAngka(int angka) throws InvalidNumberException{
        if(angka < 0 || angka > 20)
            return true;
        else
            throw new InvalidNumberException();
    }

    public static void main(String[] args){
        try{
            cekAngka(20);
        } catch (Exception e) {
            System.out.println("Terjadi error : \n" + e.getMessage());
        }
    }
}
```

Exercise!

Semua bisa karena terbiasa

Exercise `IllegalGradeException.java & Test.java`

Buatlah `custom exception class` `IllegalGradeException` yang meng-extend `Exception` dengan message `Nilai tidak valid!`.

Buat juga class `Test` yang berisi method main dari program pada file `Test.java` berisi input nilai.

Buatlah baris try-catch untuk menampilkan `IllegalGradeException` ketika nilai kurang dari 0 atau lebih besar dari 100. Tampilkan `Nilai anda adalah [NILAI]` ketika nilai tidak melanggar aturan tadi.

INPUT Nilai berupa integer

OUTPUT

```
# ketika try-catch menemukan exception
InvalidGradeException: Nilai tidak valid!
    at Test.main(blablabla)

# ketika tidak ada exception
Nilai anda adalah 80
```

Assignment!

Seperti biasa, setiap selesai praktikum pasti ada tugas

Assignment 8 Soal 1 `Simpan di folder soal01 hingga soal04`

Pilih 4 buah unchecked exception dari list dibawah, dan cari tahu:

- InterruptedIOException
- UnsupportedEncodingException
- UTFDataFormatException
- ObjectStreamException
- InvalidClassException
- InvalidObjectException
- NotSerializableException
- StreamCorruptedException
- WriteAbortedException

Jelaskan secara deskriptif tentang `Class` tersebut (tulis dalam bentuk comment) serta berikan contoh kode yang mengeluarkan keempat `Exception` tersebut!

PS: Kode harus mengeluarkan Exception dan tidak perlu berhasil compile

Assignment 8 Soal 2 `Simpan di folder soal05 hingga soal08`

Pilih 4 buah unchecked exception dari list dibawah, dan cari tahu:

- ArithmeticException
- ArrayIndexOutOfBoundsException
- ArrayStoreException
- ClassCastException
- EnumConstantNotPresentException
- NumberFormatException
- NegativeArraySizeException
- NullPointerException
- TypeNotPresentException
- UnsupportedOperationException

Jelaskan secara deskriptif tentang `Class` tersebut (tulis dalam bentuk comment) serta berikan contoh kode yang mengeluarkan keempat `Exception` tersebut!

Kemudian wrap ke dalam blok try-catch untuk menampilkan pesan `Exception`!

Assignment 8 Soal 3 `Simpan di folder soal09`

Buat Custom Exception class ala kalian sendiri dengan meng-extend class `Exception`!

Penamaan kelas dibebaskan asal representatif dengan error yang dikeluarkan!

Buat isi pesan Exception sejelas mungkin!

Jenis exception tidak boleh sama dengan yang sudah dijelaskan!

Buat class `TestException.java` untuk testing dan juga method yang dapat melempar class `Exception` yang telah anda buat!

Test method tersebut dan wrap dalam blok try-catch, kemudian tampilkan pesan errornya!

Teknis Pengumpulan

Pengerjaan dan pengumpulan tugas akan dilakukan di **Github Classroom**

Kelas A:

Link Tugas Kelas A

Kelas B:

Link Tugas Kelas B

Accept assignment terlebih dahulu lalu link akun Github dengan slot nama yang sesuai di Github Classroom

Teknis Pengumpulan

Format setiap file `.java` didahulukan dengan Nama, NPM, Kelas, Tanggal, dan Deskripsi

Cara menambah comment di java

```
// untuk single line
/* untuk multiple line */
```

Contoh Format

```
/*
  Nama  : Jane Doe
  NPM       : 99
  Kelas     : A
  Tanggal   : 1 September 2021
  Deskripsi : Class jawaban exercise-01 soal-01
*/
```

Deadline Pengumpulan ⌚

Kelas A:

7 November 2022, 23:59 WIB

Kelas B:

8 November 2022, 23:59 WIB

Waktu yang dilihat adalah waktu last commit.
Jika ada yang commit melewati deadline walaupun sudah commit sebelumnya akan dianggap telat

Terima Kasih!

Do you have any questions? Please use respective class discussion channel on Discord.

Semangat terus menjalani kuilahnya!! 🔥🔥🔥