

Pemrograman Berorientasi Objek

MODUL 6 EXCEPTION HANDLING

ASISTEN LABORATORIUM

Tujuan Pembelajaran :

1. Mengetahui apa itu exception
2. Mengetahui apa itu try...catch
3. Mengetahui apa itu throws dan throw

Exception

Exception adalah sebuah event yang terduga atau tidak diinginkan yang terjadi selama program berjalan sehingga mengganggu aliran normal dari instruksi program.

Error adalah sebuah permasalahan yang serius sehingga tidak perlu ditangkap sedangkan exception adalah kondisi yang dimanam masih memungkinkan untuk ditangkap.

Tipe dari Java Exception

1. Checked Exception

Kelas-kelas yang secara langsung mewarisi kelas Throwable kecuali RuntimeException dan Error dikenal sebagai pengecekan pengecualian, mis. IOException, SQLException dll. Pengecualian yang diperiksa diperiksa pada waktu kompilasi.

2. Unchecked Exception

Kelas yang mewarisi RuntimeException dikenal sebagai pengecualian yang tidak dicentang, mis. ArithmeticException, NullPointerException, ArrayIndexOutOfBoundsException dll. Pengecualian yang tidak dicentang tidak diperiksa pada waktu kompilasi, tetapi mereka diperiksa saat runtime.

3. Error

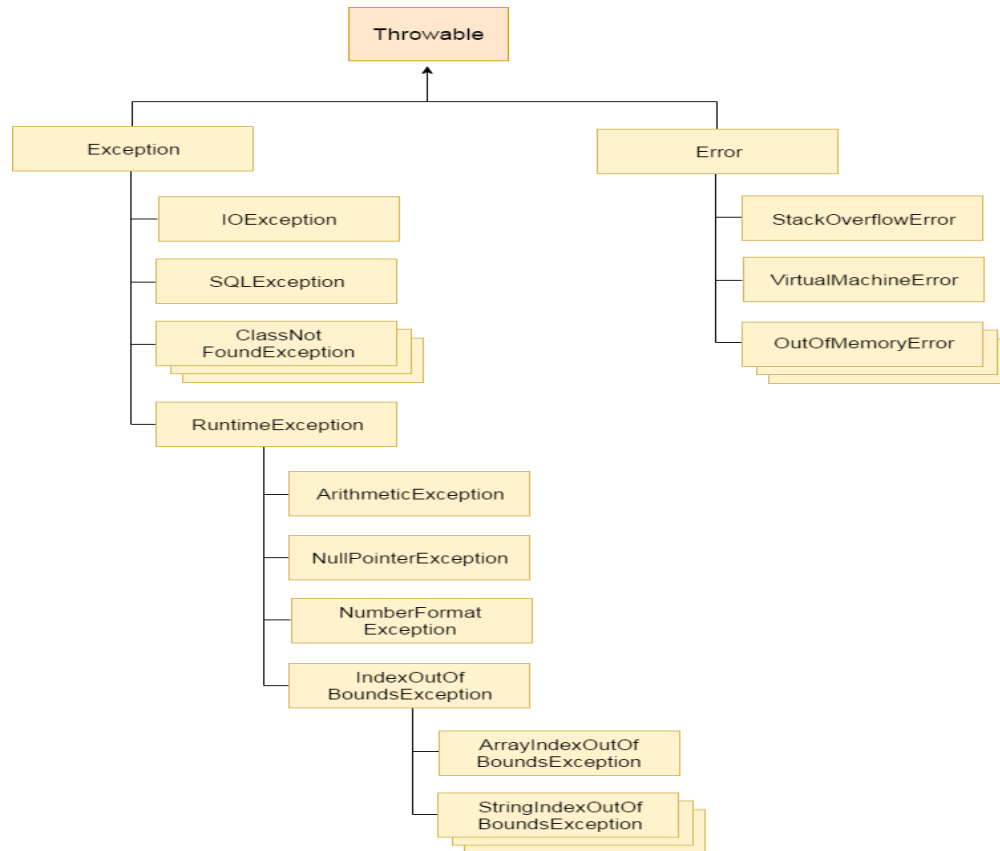
Kesalahan tidak dapat dipulihkan, mis. OutOfMemoryError, VirtualMachineError, AssertionError dll.

Hierarchy of Java Exception classes

Semua exception dan error adalah sub class dari throwable, dimana adalah kelas base dari hirarki. Salah satunya adalah exception, dimana kelas ini digunakan untuk kondisi exceptional yang harus ditangkap oleh program user. Serperti NullPointerException dan yang lainnya. Sedangkan error

MODUL 6 EXCEPTION HANDLING

digunakan oleh sistem run-time Java (JVM) untuk mengindikasikan kesalahan yang berkaitan dengan lingkungan run-time itu sendiri (JRE). `StackOverflowError` adalah contoh kesalahan seperti itu.



Java Exception Keywords

Keyword	Description
Try	Kata kunci "try" digunakan untuk menentukan blok di mana kita harus meletakkan kode pengecualian. Blok percobaan harus diikuti oleh catch atau finally. Artinya, kita tidak bisa menggunakan blok try saja.
Catch	Blok "catch" digunakan untuk menangani pengecualian. Itu harus didahului oleh try block yang artinya kita tidak bisa menggunakan catch block sendirian. Itu bisa diikuti dengan finally memblokir nanti.

MODUL 6 EXCEPTION HANDLING

Finally	Blok "finally" digunakan untuk mengeksekusi kode penting program. Ini dijalankan apakah pengecualian ditangani atau tidak.
Throw	Kata kunci "throw" digunakan untuk melempar pengecualian.
Throws	Kata kunci "throws" digunakan untuk menyatakan pengecualian. Itu tidak membuang pengecualian. Ini menentukan bahwa mungkin ada pengecualian dalam metode ini. Itu selalu digunakan dengan metode signature.

Contoh Exception Handling

```
public class JavaExceptionExample{
    public static void main(String args[]){
        try{
            //code that may raise exception
            int data=100/0;
        }catch(ArithmeticException e){System.out.println(e);}
        //rest code of the program
        System.out.println("rest of the code...");
    }
}
```

Contoh throw keyword

```
public class TestThrow1{
    static void validate(int age){
        if(age<18)
            throw new ArithmeticException("not valid");
        else
            System.out.println("welcome to vote");
    }
    public static void main(String args[]){
        validate(13);
        System.out.println("rest of the code...");
    }
}
```

MODUL 6 EXCEPTION HANDLING

Contoh throws keyword

```
import java.io.IOException;
class Testthrows1{
    void m()throws IOException{
        throw new IOException("device error");//checked exception
    }
    void n()throws IOException{
        m();
    }
    void p(){
        try{
            n();
        }catch(Exception e){System.out.println("exception handled");}
    }
    public static void main(String args[]){
        Testthrows1 obj=new Testthrows1();
        obj.p();
        System.out.println("normal flow...");
    }
}
```

Perbedaan throw keyword dengan throws keyword

Throw	Throws
Kata kunci throw Java digunakan untuk secara eksplisit melempar pengecualian	kata kunci java throws digunakan untuk menyatakan exception.
exception yang diperiksa tidak dapat disebarkan menggunakan throw saja.	exception yang diperiksa dapat diperbanyak dengan throws.
Throw diikuti oleh sebuah instance.	Throws diikuti oleh kelas.
Throw digunakan dalam metode.	lemparan digunakan dengan signature metode
tidak dapat membuang lebih dari satu exception.	dapat mendeklarasikan beberapa pengecualian, misal public void () throws IOException, SQLException.