



Riza Satria Perdana, S.T., M.T.

Teknik Informatika - STEI ITB

Exceptions

Exception Class

Pemrograman Berorientasi Objek

Cara Throw Exception

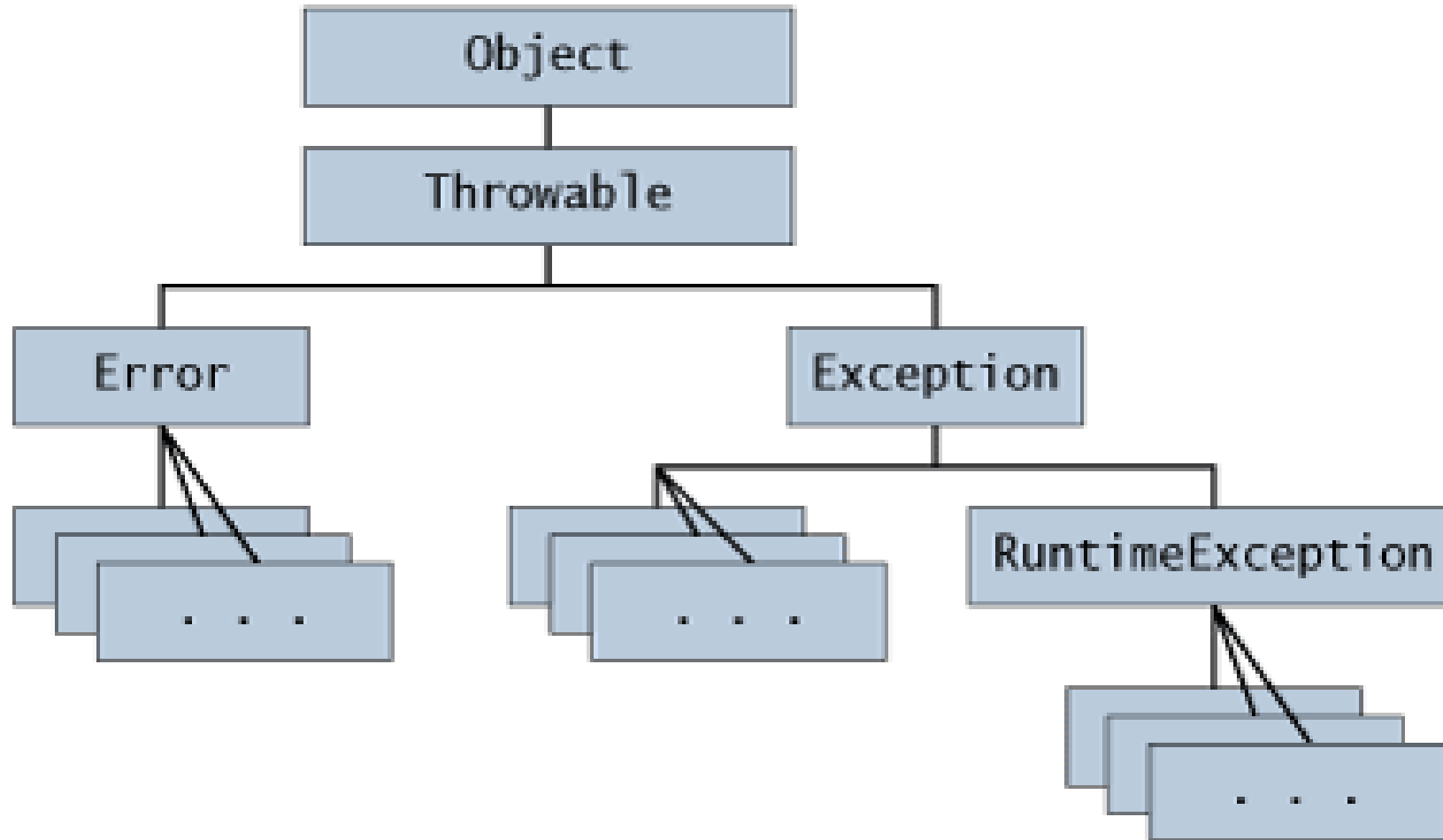
- Menggunakan **throw** statement
- Membutuhkan sebuah argumen berupa throwable objek
- Throwable objek adalah instansiasi dari kelas Throwable atau turunannya

throw *someThrowableObject*;

Throw

```
public Object pop() {  
    Object obj;  
  
    if (size == 0) {  
        throw new EmptyStackException();  
    }  
  
    obj = objectAt(size - 1);  
    setObjectAt(size - 1, null);  
    size--;  
    return obj;  
}
```

Throwable Class & Subclass



Chained Exception

- Method bisa merespon terjadinya exception dengan melempar exception lagi

```
try {  
  
} catch (IOException e) {  
    throw new SampleException("Other IOException", e);  
}
```

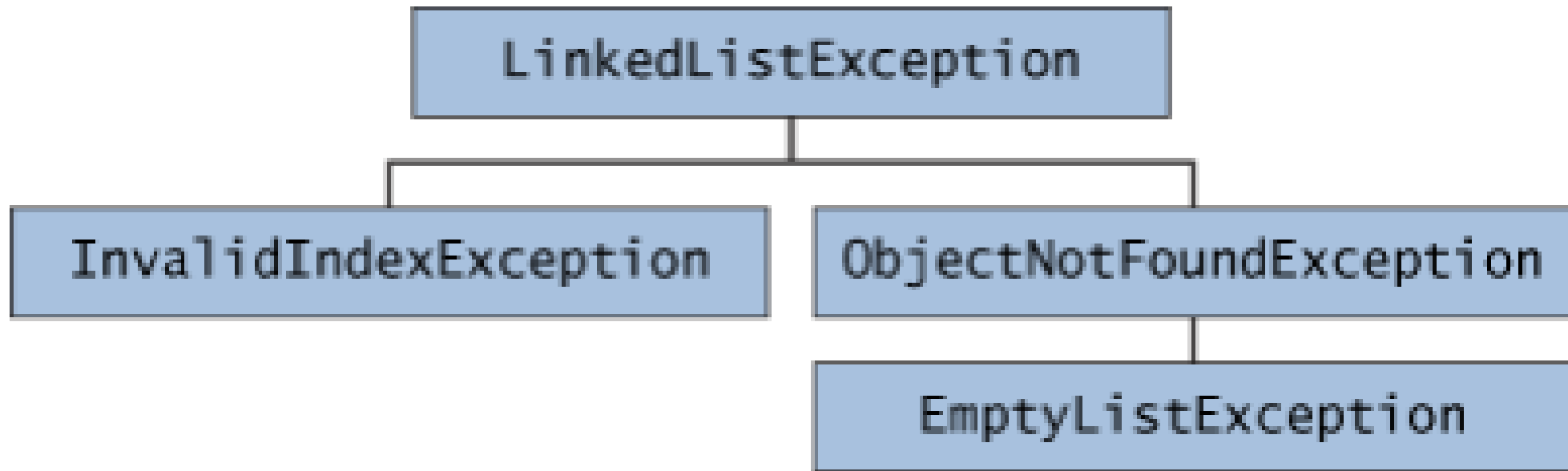
Membuat Exception Class

- Exception object dapat menggunakan kelas yang dibuat orang lain (termasuk yang ada di Java platform) atau membuat sendiri
- Exception Subclass dapat digunakan sebagai parent
- Tambahkan kata Exception di nama kelas untuk memudahkan

Membuat Exception Class

- Alasan membuat exception class sendiri:
 - Tidak ada di Java platform
 - Lebih memudahkan pengguna kelas
 - Digunakan di banyak tempat
- Perlu diperhatikan juga independensi paket atau *self-contained*

Contoh LinkedList Exception



Kelebihan Exception

- Memisahkan *error handling code* dengan *regular code*
- Mempropagasikan *error* ke atas (ke method pemanggil) di *call stack*
- Mengelompokkan dan membedakan tipe *error*

Terima Kasih