

BALTIC TALENTS ACADEMY

---

# JAVA EXCEPTIONS

# TESTAS #19 - LOKALĖ

## IŠIMTYS (EXCEPTIONS)

- ▶ Jei programos vykdymo metu atsitinka kažkas nenumatyto, kaip kad dalyba iš nulio arba bandymas iškviesti null elemento metodą, tai java sugeneruoja (“išmeta”) išimtinį įvykį
- ▶ Jei programa nepasiruošusi “pagauti” tokį išmestą įvykį, tai ji “nulūžta”

## TRY - CATCH

- ▶ Kaip pagauti?

```
try {  
    ...  
} catch (ExceptionType1 variableName) {  
    ...  
} catch (ExceptionType2 variableName) {  
    ...  
}
```

## TRY - CATCH

- ▶ Skirtingus Exception tipus galima apjungti:

```
try {  
    ...  
} catch (ExceptionType1 | ExceptionType3 variableName) {  
    ...  
} catch (ExceptionType2 variableName) {  
    ...  
}
```

## TRY - CATCH - FINALLY

- ▶ Pabaigoje gali eiti neprivalomas **finally** blokas, kuris visada(!!!) bus vykdomas paskutinis po **try** ir/ar **catch** bloko. Jei yra **finally** blokas, tai **catch** bloko/blokų gali ir nebūti:

```
try {  
    ...  
} catch (ExceptionType1 | ExceptionType3 variableName) {  
    ...  
} finally {  
    ...  
}
```

## RESURSŲ TRY SAKINYS

- ▶ Tam tikri objektai, kuriuos po panaudojimo būtinai reikia “uždaryti” ir realizuojantys `AutoCloseable`, gali būti naudojami taip vadiname resursų `try` sakinyje. Jame taip pat gali būti `catch` ir/ar `finally` blokai:

```
try (Resursas r = resurso sukūrimas) {
```

```
    ...
```

```
}
```

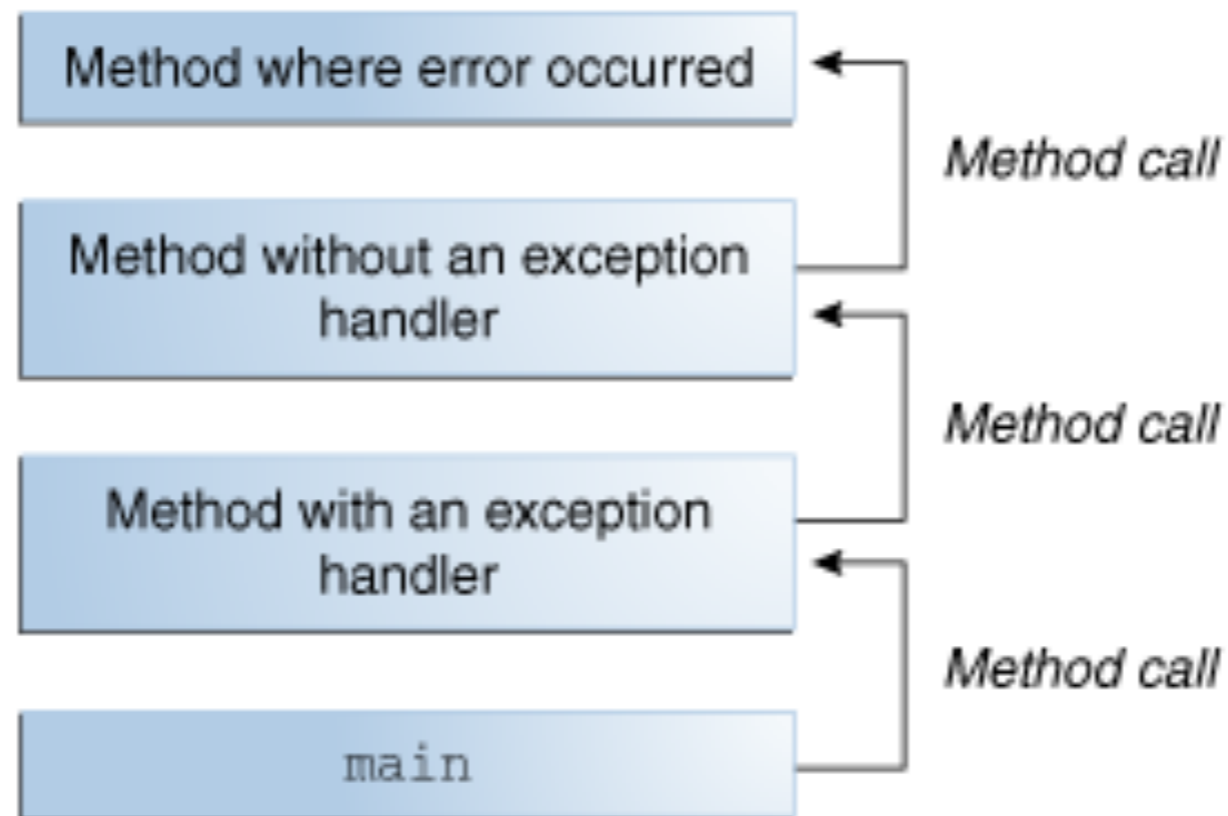
arba

```
try (resurso1 sukūrimas; resurso2 sukūrimas; ...) {
```

```
    ...
```

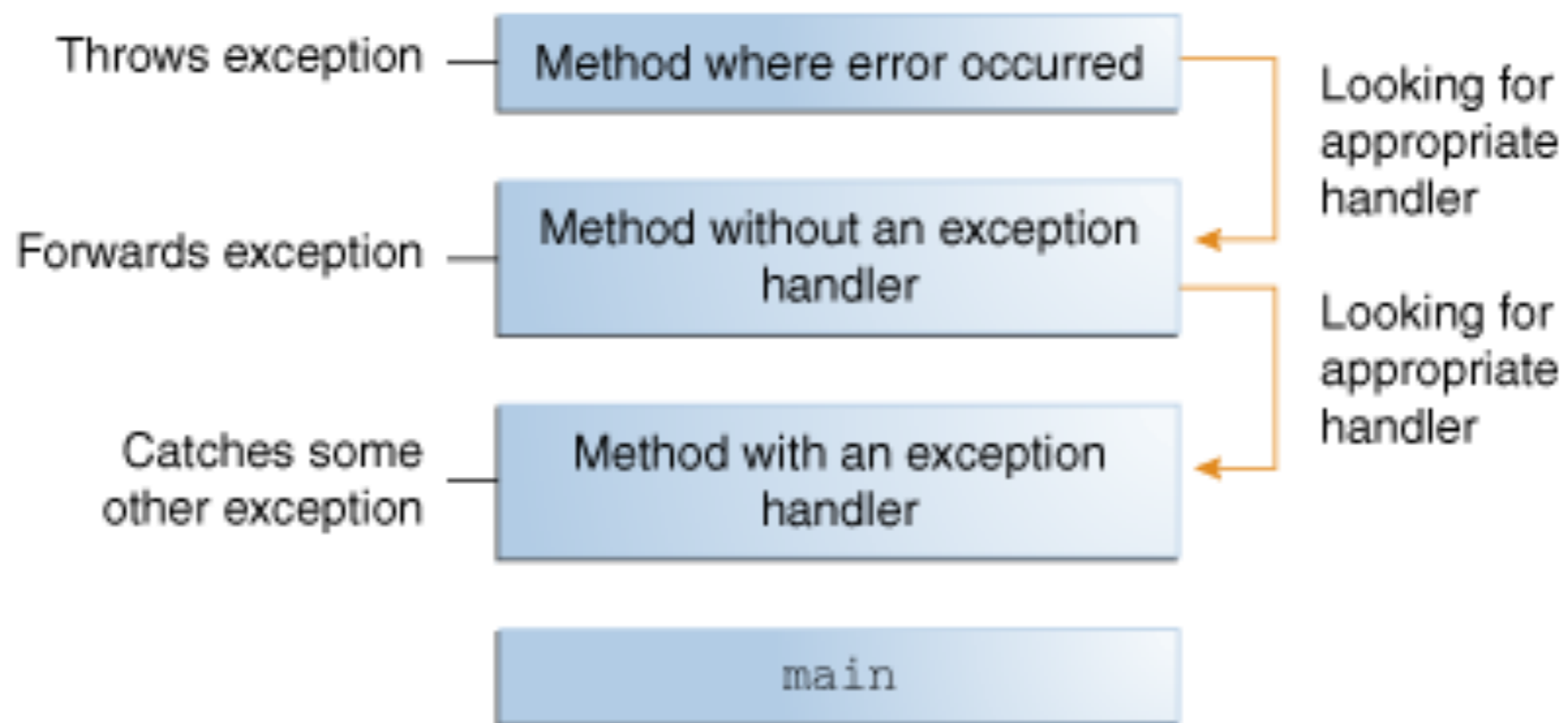
```
}
```

# IŠIMČIŲ GAUDYMAS





# IŠIMČIŲ GAUDYMAS



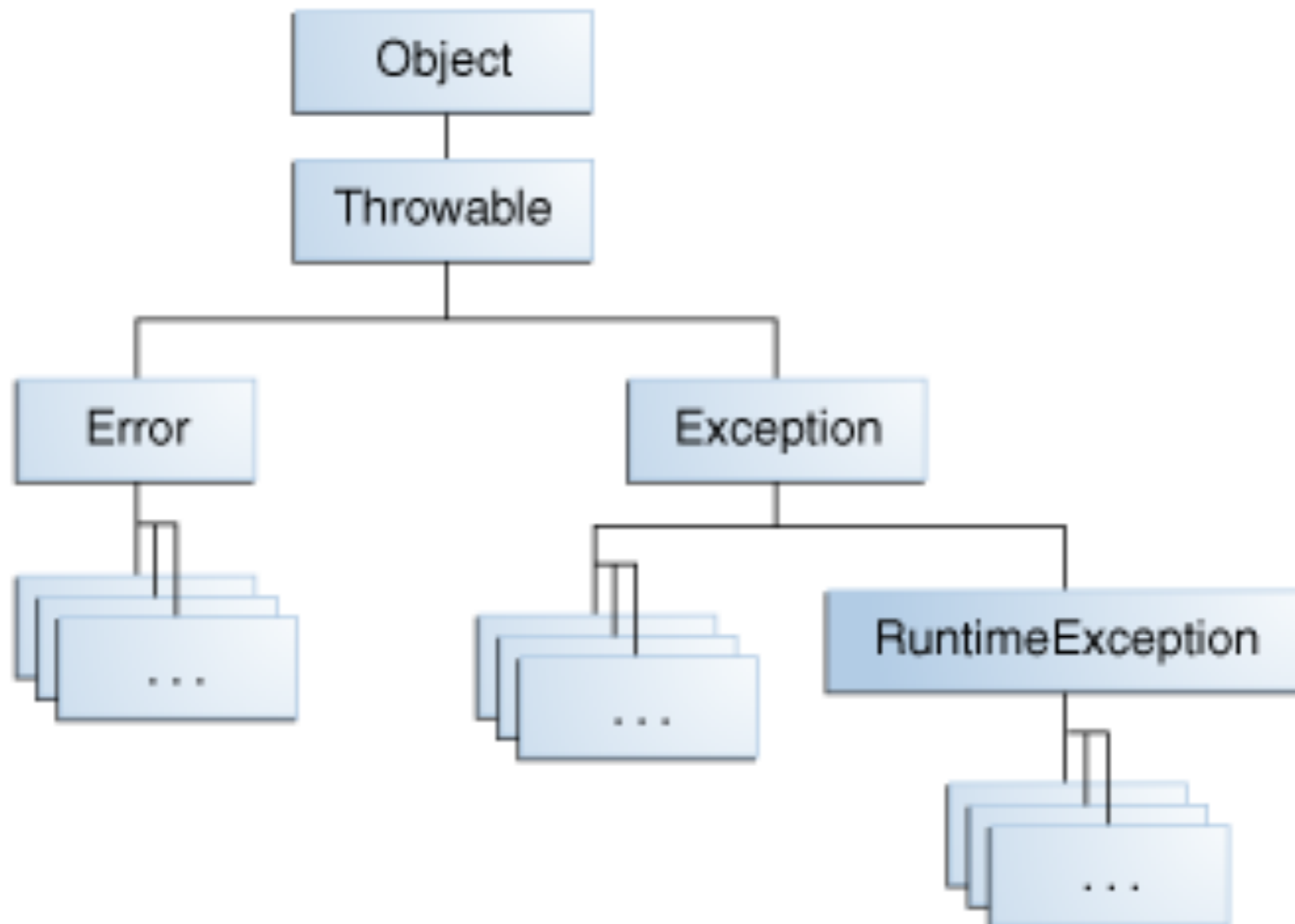
## IŠMESTI IŠIMTĮ (THROW EXCEPTION)

- ▶ Galima ir patiems mėtyti (throw) kokias tik norime išimtis (exceptions)

**throw new** *ExceptionType*(...)

- ▶ Galima išmesti “išmesti” bet kokį objektą, kuris išplečia **Throwable** klasę

# IŠMETAMŲ KLASIŲ SCHEMA



## EXCEPTION RŪŠYS

- ▶ Exception yra dviejų rūšių
  - ▶ Nežymėtos (unchecked)
  - ▶ Žymėtos (checked)

## NEPAŽYMĖTOS (UNCHECKED)

- ▶ Tai tokios išimtys, kurios gali būti išmestos vykdant paprastas normalias operacijas
- ▶ Tokios išimtys neprivalo būti pažymėtos prie metodo ar konstruktoriaus, kad jos gali būti išmestos (todėl ir vadinamos nežymėtomis)
- ▶ Tokių išimčių tėvinė klasė yra **RuntimeException**
- ▶ Pvz.: **ArithmeticException**, **IndexOutOfBoundsException**

## ŽYMĖTOS (CHECKED)

- ▶ Tokios išimtys privalo būti pažymėtos prie metodo ar konstruktoriaus, kad jos gali būti išmestos
- ▶ Tokių išimčių tėvinė klasė yra **Exception**

## SUKURTOS IŠIMTYS

- ▶ Kai neužtenka standartinių išimčių, tai paprastai sukuriamos išimtys pritaikytos programai
- ▶ Galia kurti ir žymėtas, plečiant klasę **Exception**, ir nežymėtas, plečiant **RuntimeException**, išimtis
- ▶ *Pastaba: reiktų laikytis tokio principo - mesti nežymėtą (RuntimeException) išimtis tik tada, kad atsitinka tokia klaida, kurios duotoje programos vietoje jau neįmanoma pataisyti ar jos išvengti. Jei klaida pataisomatai reiktų naudoti žymėtas išimtis (Exception)*

# UŽDAVINYS