

# JAVA programavimo kalba

Java išimtys (Exceptions)

# 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 {
```

```
}
```

# Try-with-resources

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

- Išimtys gali būti sugautos ir perkeltos kitur.
- Išimtys dažniausiai turi pabaigą dar prieš baigiantis programai.

```
void method cl() throws Exception {}  
void method de() {  
    try {  
        cl();  
    } catch (Exception e ) {...}  
}
```

```
public static void main(String[] args) {  
    de();  
}
```

# Išmesti išimtį

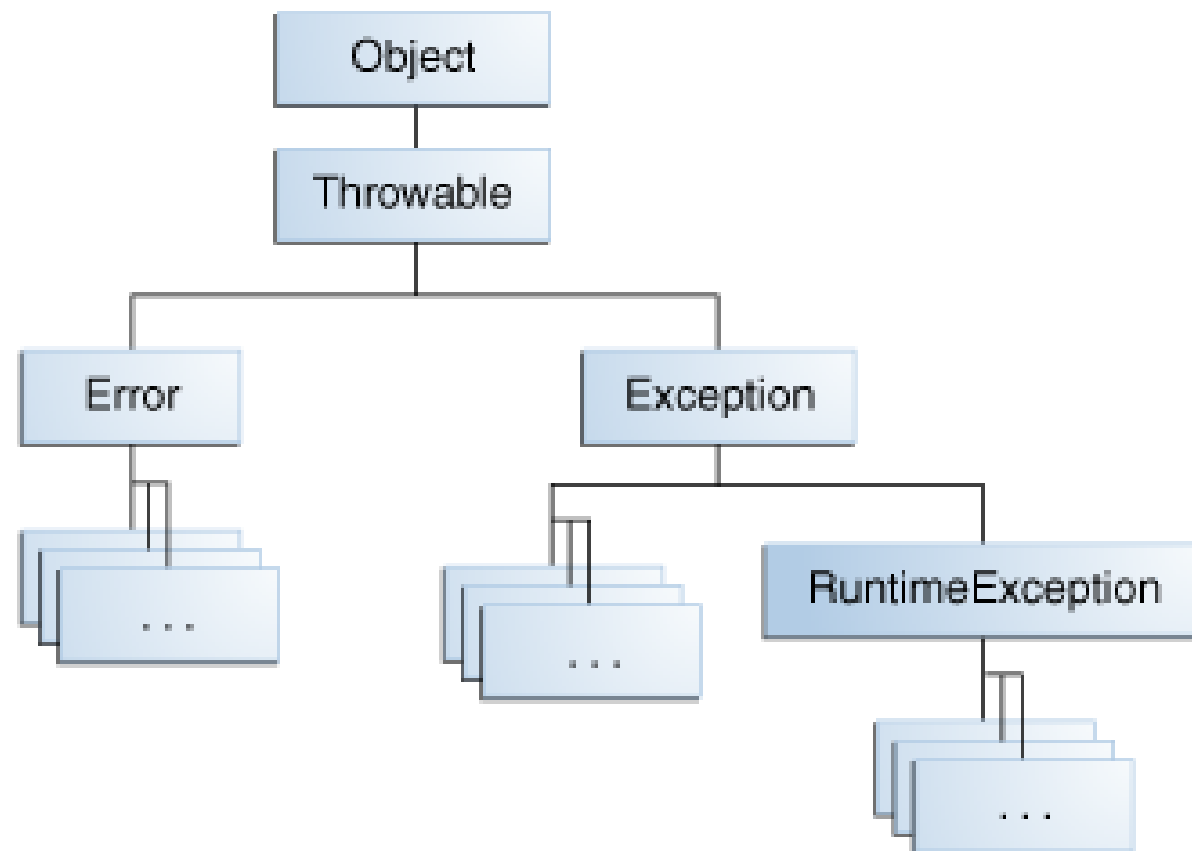
- Galima ir patiems mėtyti (throw) kokias tik norime išimtis (exceptions)

**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) išimtį tik tada, kad atsitinka tokia klaida, kurios duotoje programos vietoje jau neįmanoma pataisyti ar jos išvengti. Jei klaida pataisomatai reiktų naudoti žymėtas išimtis (Exception).*

# Uždaviniai

1. Sukurkite savo exception, kuri turetu savo klaidos koda ir to tam kodui atitinkama klaidos pranesima. Sukurkite tokį metodą, kuriame išmesite exception o public static void main metode pagausite tą išimtį ( exception ) ir atspausdinsite klaidos kodą bei pranešimą.
2. Panaudokite prieš tai sukurtą exception klasę ir parašykite metodą kuris gražintų sveikų skaičių dalybą. Pvz metodas(Integer a , Integer b). Jei metode paduodamos null reikšmes ismeskite exception. Pabandykite iskviesti ta metoda su try-with-resources ir jeigu yra klaida atspausdinti ja.