

# Jazyk Java 1

## Seminář 6

Daniel Bazala

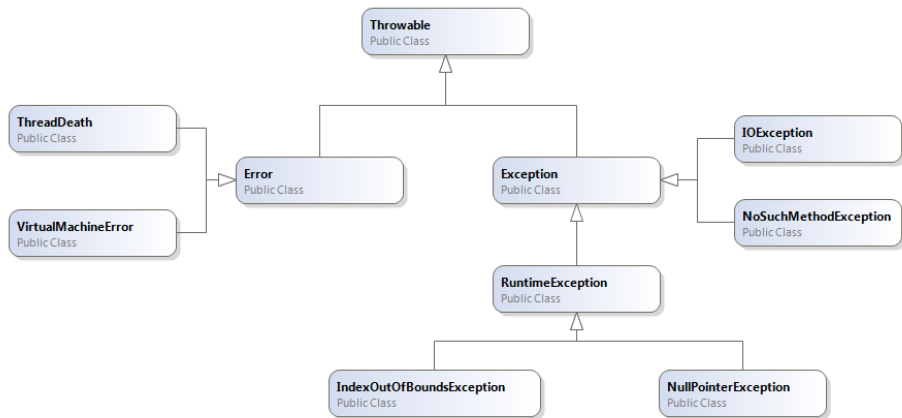


Katedra informatiky  
Univerzita Palackého v Olomouci

# Výjimky

- událost, která může vzniknout během vykonávání programu při narušení jeho běžné funkcionality.
- některým běhovým chybám nelze předejít, ale můžeme je ošetřit
- např. uživatel zadá neplatný vstup, soubor neexistuje..
- kritické chyby bychom neměli ošetřovat, mají vést k zastavení programu a výpisu chyby

# Základní hierarchie výjimek



- třída Throwable:

<https://docs.oracle.com/en/java/javase/16/docs/api/java.base/java/lang/Throwable.html>

- třída Exception:

<https://docs.oracle.com/en/java/javase/16/docs/api/java.base/java/lang/Exception.html>

- tutorial:

<https://docs.oracle.com/javase/tutorial/essential/exceptions/index.html>

# Základní typy výjimek

- třída `Error` - kritické chyby, neošetřujeme
- třída `RuntimeException` - běhové výjimky, situace, které často způsobil programátor (neplatný index pole, přístupuování k nullové referenci..)
- třída `Exception` - situace, které jsme schopni ošetřit (neplatný vstup, soubor neexistuje..)

# Vyvolání výjimky

- klíčové slova `throw`, `throws`

```
public static void main(String[] args) throws Exception {  
    throw new Exception("Doslo k vyjimce.");  
}
```

```
Exception in thread "main" java.lang.Exception: Doslo k vyjimce.  
    at com.company.Main.main(Main.java:30)
```

# Konstrukce try-catch-finally

- blok try provádí kritický kód
- bloky catch odchyťávají výjimky
- blok finally se provede vždy

```
try {  
    int a = 10 / 0;  
} catch (Exception e) {  
    e.printStackTrace();  
    System.out.println("Doslo k odchytceni vyjimky.");  
} finally {  
    System.out.println("Provede se vzdy.");  
}
```

```
java.lang.ArithmeticException: / by zero  
    at com.company.Main.main(Main.java:31)  
Doslo k odchytceni vyjimky.  
Provede se vzdy.
```



# Konstrukce try-catch-finally

- catch odchyťává od konkrétnější vyjímky k obecnější

```
try {  
    int a = 10 / 0;  
} catch (ArithmeticException e) {  
    System.out.println("Aritmeticka vyjimka.");  
} catch (RuntimeException e) {  
    System.out.println("Behova vyjimka.");  
} catch (Exception e) {  
    System.out.println("Vyjimka.");  
}
```

Aritmeticka vyjimka.

# Vlastní výjimky

- můžeme zavést vlastní výjimky

```
class MyException extends Exception {  
    MyException(Throwable e) {  
        super("Moje_vyjimka", e);  
    }  
}
```

```
public static void main(String[] args) throws MyException {  
    try {  
        int a = 10 / 0;  
    } catch (ArithmeticException e) {  
        throw new MyException(e);  
    }  
}
```

```
Exception in thread "main" com.company.MyException: Moje vyjimka  
    at com.company.Main.main(Main.java:33)  
Caused by: java.lang.ArithmeticException: / by zero  
    at com.company.Main.main(Main.java:31)
```

# Výčtové datové typy

# Výčtové datové typy

- uživatelsky definované datový typy, který nabývá pouze omezeného a předem daného množství hodnot
- jedná se o třídu bez klíčového slova class (je možné přidat metody)
- enum lze použít jako přepínač v switch a lze přes něj iterovat

# Enum

- enum pro dny v týdnu

```
public enum Day {  
    SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY;  
}
```

- položky enumu mohou obsahovat více hodnot

```
public enum Day {  
    MONDAY(1), TUESDAY(2), WEDNESDAY(3), THURSDAY(4), FRIDAY(5), SATURDAY(6), SUNDAY(7);  
}
```

```
private final int day;
```

```
public int getDayNumber() {  
    return day;  
}
```

```
Day(int i) {  
    this.day = i;  
}
```

```
}
```

```
Day d = Day.FRIDAY;  
System.out.println(d.getDayNumber()); // 5
```

# Optional

# Optional

- generická třída pro reprezentaci volitelných hodnot

```
Optional<String> empty = Optional.empty();  
System.out.println(empty.isPresent()); // false
```

```
String name = "Boris";  
Optional<String> opt = Optional.of(name);  
System.out.println(opt.isPresent()); // true  
System.out.println(opt.get()); // Boris
```

```
String name = null;  
Optional.of(name); // Vyjimka
```

```
String name = null;  
Optional<String> opt = Optional.ofNullable(name);  
System.out.println(opt.isPresent()); // false
```

# Úkol



# Úkol seminář 6

- <http://marcus.webly3d.net/ukol6>