

# Jazyk SCALA – programovanie I

Kód kurzu: SCALA\_INTRO

Absolvent bude do hĺbky oboznámený s OOP aj funkcionálnymi technikami a ich využitím v silne typovom jazyku. Kurz začína základnými konštrukciami jazyka, pokračuje s OOP, potom funkcionálnymi transformáciami, typovým systémom jazyka a končí sa asynchrónnym programovaním.

## Požadované vstupní znalosti:

- Kurz predpokladá znalosť akéhokoľvek iného programovacieho jazyka a základov algoritmickej.

## Metody výuky:

- Výklad s prezentáciou
- Praktické ukážky na malých útržkoch kódu, tzv "tabule" alebo scratches v IntelliJ Idea
- Málé cvičenia pre vyskúšanie práve vysvetleného
- Veľké samostatné úlohy z tématického celku vyhodnotené individuálne pre každého študenta (s radami ako ďalej)

## Studijní materiály:

- Prezentace probírané látky v tištěné nebo online formě.

## Osnova:

Instalácia potrebných nástrojov/Scala a úvod

- Inštalácia Scala-cli
- Scala v IntelliJ Idea (Scala plugin)
- Čo je Scala, malá ukážka cieľov a smerovania kurzu aj s praktickými príkladmi
- Vytvorenie projektu

Základné konštrukcie jazyka a úvod do typov

- Meniteľné vs nemeniteľné premenné - princíp immutable
- Základné typy: numerické, reťazce, pravdivostné hodnoty
- Podmienky a cykly
- For cyklus ktorý vracia výsledok, teória výrazov - čo je výraz a čo nie
- Funkcie, rekurzia, tail rekurzia
- Interpolácia reťazcov
- Typ tuple, "rozbitie" na premenné, pattern matching
- Typ option ako jednoprvková kolekcia, reťazenie options vo for cykle, Option(null)
- List/Seq/Vector/Set/Map a operácie nad nimi
- Mutable varianty kontajnerov

Operácie nad kolekciami

- foreach, map, flatMap
- rekurzívne počítanie s kolekciami, foldLeft, sum, reduce
- find, headOption, filter, exists, contains, collect, groupBy, mkString

OOP v Scala

- Trieda a jej atribúty, konštruktor, companion object a apply
- case class, metóda copy, pattern matching
- Traits a viacnásobná dedičnosť
- Anonymné triedy
- sealed trait a enumerácie

Pokročilejšie konštrukcie jazyka

- Vyššie funkcie, funkcia ako parameter a návratová hodnota
- Kedy sa kód skutočne zavolá/vyhodnotí?
- Lambdy

### GOPAS Praha

Kodaňská 1441/46  
101 00 Praha 10  
Tel.: +420 234 064 900-3  
[info@gopas.cz](mailto:info@gopas.cz)

### GOPAS Brno

Nové sady 996/25  
602 00 Brno  
Tel.: +420 542 422 111  
[info@gopas.cz](mailto:info@gopas.cz)

### GOPAS Bratislava

Dr. Vladimíra Clementisa 10  
Bratislava, 821 02  
Tel.: +421 248 282 701-2  
[info@gopas.sk](mailto:info@gopas.sk)



Copyright © 2020 GOPAS, a.s.,  
All rights reserved

# Jazyk SCALA – programovanie I

- Obalenie primitívnych hodnôt do typov namiesto priameho použitia
- implicitné funkcie, konverzie
- implicitné triedy, doplňanie metód
- implicitné hodnoty
- Try vs try
- Reťazenie potencionálne neúspešných operácií vo for cykle, recover, orElse
- Funkcia s typovým parametrom
- Ohraničenie typov
- Náš vlastný reduce

## Asynchrónne programovanie

- Future
- Await.ready/result
- Execution Context (global, fixed thread pool, cached, work stealing pool)
- Future neznamená thread, Čo je to thread pool
- map, flatMap, skladanie vo for cykle
- andThen, recover, transform
- lazyness

### GOPAS Praha

Kodaňská 1441/46  
101 00 Praha 10  
Tel.: +420 234 064 900-3  
[info@gopas.cz](mailto:info@gopas.cz)

### GOPAS Brno

Nové sady 996/25  
602 00 Brno  
Tel.: +420 542 422 111  
[info@gopas.cz](mailto:info@gopas.cz)

### GOPAS Bratislava

Dr. Vladimíra Clementisa 10  
Bratislava, 821 02  
Tel.: +421 248 282 701-2  
[info@gopas.sk](mailto:info@gopas.sk)



Copyright © 2020 GOPAS, a.s.,  
All rights reserved