Stream Programming Summer 2019

Introduction to Scala and OOP vol. 5 — 26.03, 2019

Dr inż. Wojciech Macyna *Scribe: Krzysztof Agieńczuk*

1 Overview

In the last lecture we talked about **collections**, **function closures** and **functions**.

In this lecture we covered **collections**.

2 Collections

One of the available collections in Scala is a set. Sets can be concatenated using double + sign. Second container are lists. They are immutable and can contain elements of different type if type is not explicitly specified.

Maps are collection of type $< Key, Value >$. Keys should be distinct as it is not allowed to declare item with the same keys. Such instruction will overwrite value for given key.

3 Streams

Dumb guide to streams: imagine you have a computer. Imagine you have data coming in. You work on data as it comes, and send it further. This is stream. You never have access to full stream (eg. you can access head, but not tail).

In Scala streams are undefined. Simple definition of stream
`val stream = 1 #:: 2 #:: 3 #:: Stream.empty`

Print 3 elements of stream `stream take 3 foreach println`

Streams have some useful methods like `filter` or `take`. Again, for more I send you to documentation, Odersky's book and StackOverflow.

Rest of lectures was basically showing examples from Odersky's book on using the streams.