Scala核心编程

-Scala概述

尚硅谷研究院

Scala语言概述

- **why is Scala语言？**

1) **Spark—新一代内存级大数据计算框架，是大数据的重要内容。**
2) Spark就是使用Scala编写的。因此**为了更好的学习Spark, 需要掌握Scala这门语言**。

3) Scala 是 Scalable Language 的简写，是一门多范式(范式/编程方式[面向对象/函数式编程])的编程语言

4) 联邦理工学院洛桑（EPFL）的Martin Odersky于2001年开始设计Scala

5) **Spark的兴起，带动Scala语言的发展！**

- **Scala语言诞生小故事**

创始人马丁·奥德斯基（Martin Odersky）是**编译器**及编程的狂热爱好者，长时间的编程之后，**希望发明一种语言**，能够让写程序这样的基础工作变得**高效**，简单。所以当接触到JAVA语言后，对JAVA这门便携式，运行在网络，且存在垃圾回收的语言产生了极大的兴趣，所以**决定将函数式编程语言的特点融合到JAVA中，由此发明了两种语言（Pizza & Scala） 递归**

Pizza和Scala极大地推动了Java编程语言的发展。[如何理解?]

jdk5.0 的泛型，for循环增强, 自动类型转换等，都是从Pizza 引入的新特性。
jdk8.0 的类型推断，Lambda表达式就是从scala引入的特性。

且现在主流JVM的javac编译器就是马丁·奥德斯基编写出来的。Jdk5.0 Jdk8.0的编译器就是马丁·奥德斯基写的，因此马丁·奥德斯基 一个人的战斗力抵得上一个Java开发团队。

- **Scala 和 Java 以及 jvm 的关系分析图**

一般来说，学Scala的人，都会Java，而Scala是基于Java的，因此我们需要将Scala和Java以及JVM 之间的关系搞清楚，否则学习Scala你会蒙圈。

建议：如果没有任何Java基础的同学，先学Java，至少要学习JavaSE，再学习Scala。

我们分析一下：**Scala 和 Java 以及 jvm 的关系 (重要!)。**

Scala 和 Java 以及 jvm 的关系p

- **Scala语言的特点**

Scala是一门以java虚拟机（JVM）为运行环境并将**面向对象**和**函数式编程**的最佳特性结合在一起的**静态类型编程语言**。

1) Scala 是一门多范式 (multi-paradigm) 的编程语言，Scala支持面向对象和函数式编程
2) Scala源代码(.scala)会被编译成Java字节码(.class)，然后运行于JVM之上，**并可以调用现有的Java类库，实现两种语言的无缝对接**。[案例演示]

3) scala 单作为一门语言来看， 非常的**简洁高效 （三元运算， ++ ， --）**

4) Scala 在设计时，马丁·奥德斯基 是参考了Java的设计思想，可以说Scala是源于java，同时马丁·奥德斯基 也加入了自己的思想，将函数式编程语言的特点融合到JAVA中, 因此，对于学习过Java的同学，只要在学习Scala的过程中，搞清楚Scala 和 java相同点和不同点，就可以快速的掌握Scala这门语言
5) 快速有效掌握Scala的建议 [1. 学习scala 特有的语法 2. 搞清楚 scala 和java 区别 3. 如何规范的使用scala]

Scala开发环境搭建

- **Windows下搭建Scala开发环境**

安装&配置

1) Scala需要Java运行时库，安装Scala需要首先安装JVM虚拟机并配置好，推荐安装JDK1.8
2) 在http://www.scala-lang.org/ 下载Scala2.11.8程序安装包

Archive	System	Size
scala-2.11.8.tgz	Mac OS X, Unix, Cygwin	27.35M
scala-2.11.8.msi	Windows (msi installer)	109.35M
scala-2.11.8.zip	Windows	27.40M
scala-2.11.8.deb	Debian	76.02M
scala-2.11.8.rpm	RPM package	108.16M
scala-docs-2.11.8.txz	API docs	46.00M
scala-docs-2.11.8.zip	API docs	84.21M
scala-sources-2.11.8.tar.gz	Sources	

提示:
根据不同的操作系统选择不同的安装包，下载完成后，将安装包解压到安装目录

- **Windows下搭建Scala开发环境**

安装&配置

3) 配置Jdk的环境变量
4) 配置SCALA_HOME
   SCALA_HOME= D:\program\scala-2.11.8
5) 将Scala安装目录下的bin目录加入到PATH环境变量
   在PATH变量中添加：%SCALA_HOME%\bin
6) 在终端中输入“scala”命令打开scala解释器

- **Windows下搭建Scala开发环境**

Scala的REPL

- **介绍**

  上面打开的scala命令行窗口，我们称之为REPL，是指：Read->Evaluation->Print->Loop，也称之为交互式解释器。

- **说明**

  在命令行窗口中输入scala指令代码时，解释器会读取指令代码(R)并计算对应的值(E)，然后将结果打印出来(P)，接着循环等待用户输入指令(L)。
  从技术上讲，这里其实并不是一个解释器，而是指令代码被快速的编译成Java字节码并被JVM加载执行。最终将执行结果输出到命令行中

- **示意图**

- **Linux下搭建Scala开发环境**

安装&配置

Linux下安装Scala 的原理机制一样，操作的具体步骤：

Linux下安装 Scala.

```
[root@hadoop102 桌面]#
[root@hadoop102 桌面]#
[root@hadoop102 桌面]# scala -version
Scala code runner version 2.11.8 -- Copyright 2002-2016, LAMP/EPFL
[root@hadoop102 桌面]#
```

- 安装和配置Scala开发环境练习

请同学们安装和配置Scala[windows & Linux], 可以下正确的执行，如图：(8min)

- **Scala的开发工具**

IDEA介绍：

IDEA 全称IntelliJ IDEA，是用于java语言开发的集成环境（也可用于其他语言），IntelliJ在业界被公认为最好的java开发工具之一。IDEA是JetBrains公司的产品，这家公司总部位于捷克共和国的首都布拉格。

1) java开发工具很多，比如netbean,eclipse等等，单开发Scala可选的工具不多，主要使用IDEA
2) Idea工具开发Scala的快捷键也不是很多，所以使用相对比较简单
3) IDEA不是专门用于开发Scala的IDE，但是确是最适合开发Scala的工具，因为在我们实际工作中，大部分是开发项目，而大数据项目不可避免的会使用到Java，所以会进行Java 和 Scala 两种语言的混合编程。 而Idea 可以很好的支持Java和Scala的开发。

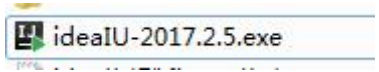

IntelliJ Idea 常用快捷键

- **Scala的开发工具**

IDEA的安装：

看老师的演示和图示：

- 说明：idea的安装 直接下一步即可(傻瓜式安装)。



- 安装成功，会看到如下界面

- **Scala的开发工具**

Scala插件安装：

默认情况下IDEA不支持Scala的开发，需要安装Scala插件，具体看老师演示和图示：

idea配置for scala.z

- **Scala快速开发入门**

需求说明

要求开发一个Hello.scala 程序，可以输出“hello,世界!" [对scala程序基本结构说明]

```
d:\scalaDemo>scala Hello
hello, 世界!
```

windows下开发步骤[先使用ed]

1) 可以直接使用文本开发工具[editplus]
2) 将 Scala 代码编写到扩展名为 Hello.scala 的文件中。
   [说明: 比如将源码在目录 d:/scalademo下]
3) 通过 scalac 命令对该 scala 文件进行编译，生成 .class 文件。[和javac类似]
4) 命令行下 执行 scala Hello 就可以看到运行效果。
5) **注意**：scala Hello.scala 命令可以直接运行 Hello.scala 程序 [内部也会有编译和运行过程]

- **Scala语言快速开发入门**

linux下开发步骤

1) 直接使用vim开发 ,一个遍历数组的案例
2) 将 Scala 代码编写到扩展名为 Hello.scala 的文件中。[代码说明]
3) 通过 scala 命令对该 scala 文件进行编译，生成 .class 字节码文件。
4) 在终端 执行 scala Hello 就可以看到运行效果。
5) **注意**：通过 scala 命令可以直接运行 Hello.scala 程序

```
object Hello {
        def main(args : Array[String]): Unit = {
                var arr = Array(10,20,30)
                for (item <- arr) {
                        println("item=" + item)
                }
        }

}
```

- **Scala语言快速开发入门**

IDE工具Idea 来开发 “hello,world”

使用文本工具开发项目可以很好的理解运行原理，但是不利于开发综合项目，所以在实际开发中我们要使用Idea来开发, 看老师演示

idea 开发 slaca 项目

Scala程序反编译

1) 看反编译代码
2) 模拟代码


```
final class TestJava$
{
    public static final TestJava$ module$;
    static
    {
        module$ = new TestJava$();
    }
    public void main(String[] args)
    {
        System.out.println("hello, Var Demo01~~~~");
    }
    // private Test100() { module = this; }
}
```

- **Scala语言快速开发入门**

Scala程序结构的说明

```
object VarDemo01 {
  def main(args: Array[String]): Unit = {
    println("hello, Var Demo01")
  }
}
```

- **Scala语言快速开发入门**

课堂小练习

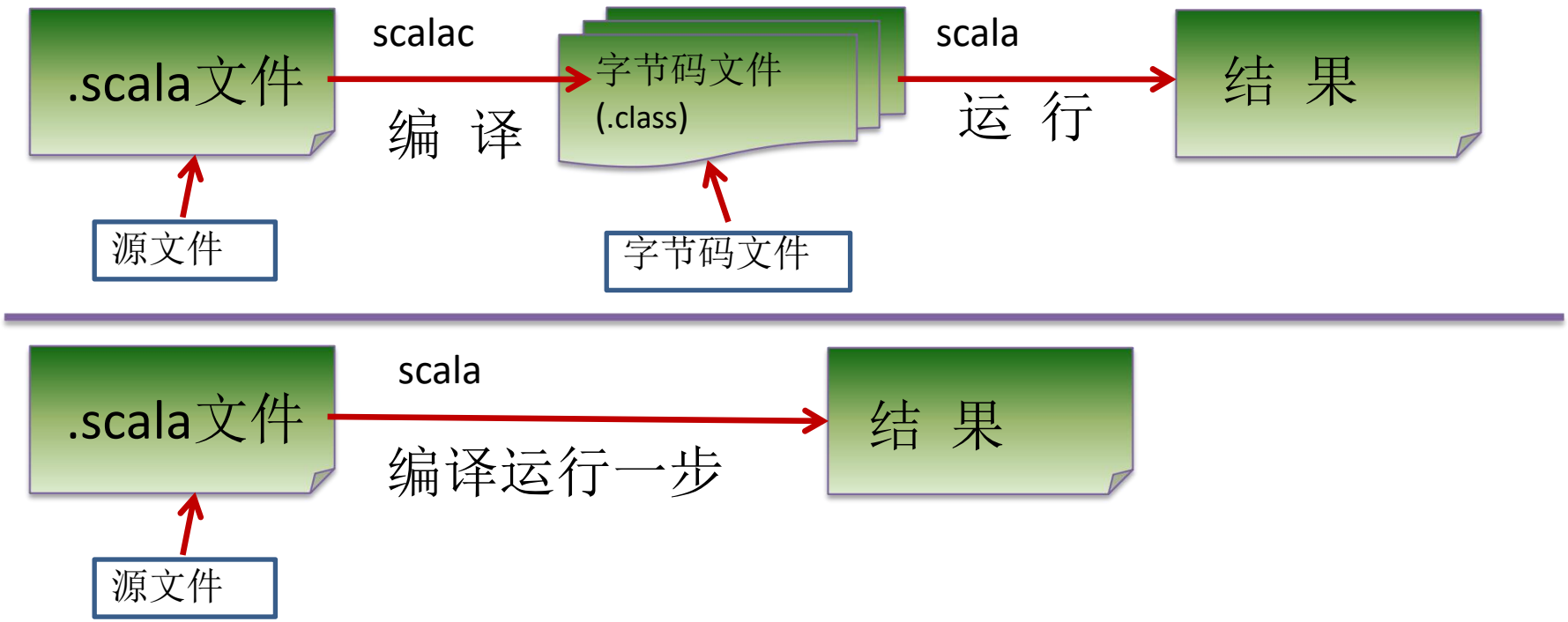
要求使用Idea 下开发一个Hi.scala 程序，可以输出“hello,scala!" (10min)

1) 包名为 com.atguigu.chapter01
2) object 名称为 Hi

- **Scala语言快速开发入门**

Scala执行流程分析

➢ 示意图



- **Scala语言快速开发入门**

Scala程序开发注意事项(重点)

1) Scala源文件以 “.scala" 为扩展名。
2) Scala程序的执行入口是main()函数。
3) Scala语言严格区分大小写。
4) Scala方法由一条条语句构成，每个语句后**不需要**分号(**Scala语言会在每行后自动加分号**)，这也体现出Scala的简洁性。
5) 如果在同一行有多条语句，除了最后一条语句不需要分号，其它语句需要分号。

- **Scala语言转义字符**

Scala常用的转义字符(escape char)

1) \t ：一个制表位，实现对齐的功能
2) \n ：换行符
3) \\ ：一个\
4) \" ：一个"
5) \r ：一个回车 println("hello\rk");

```
println("姓名\t年龄")
println("姓名\t20")
println("Hello,张三丰\nhello,郭襄")
println("C:\\Users\\Desktop\\day1_part1\\test100")
println("尚硅谷说: \"Go语言开始了\"")
println("hello\ra")
```

说明

应用实例

- **Scala语言转义字符**

课后练习

要求：请使用**一句输出语句**，达到输入如下图形的效果：

- **Scala语言输出的三种方式**

1) 字符串通过+号连接（类似java）。
2) printf用法 （类似C语言）字符串通过 % 传值。
3) 字符串通过$引用(类似PHP）。

```
val name = "ApacheCN"
val age  = 1
val url  = "www.atguigu.com"
println("name=" + name + " age=" + age + " url=" + url)
printf("name=%s, age=%d, url=%s \n", name, age, url)
println(s"name=$name, age=$age, url=$url")
```

- **Scala源码的查看的关联**

在使用scala过程中，为了搞清楚scala底层的机制，需要查看源码，下面看看如果关联和查看Scala的源码包

1) 查看源码, 选择要查看的方法或者类, 输入 ctrl + b
2) 关联源码,看老师演示

注释

- **注释(comment)**

介绍：

用于注解说明解释程序的文字就是注释，注释提高了代码的阅读性；
注释是一个程序员必须要具有的良好编程习惯。将自己的思想通过注释先整理出来，再用代码去体现。

Scala中的注释类型

1) 单行注释
2) 多行注释
3) 文档注释

- 注释(comment)

单行注释：

- 基本格式

格式： //注释文字

- 应用实例

多行注释：

- 基本格式

格式： /* 注释文字 */

- 应用实例

- 注释(comment)

文档注释：

注释内容可以被工具 scaladoc 所解析，生成一套以网页文件形式体现的该程序的说明文档

案例演示

```
package com.atguigu.chapter01.Demo01

object Hello {
  /**
    * @deprecated xxx
    * @example testing coding
    * @param args
    */
  def main(args: Array[String]): Unit = {
     println("helllo")
  }
}
scaladoc -d d:/ Hello.scala
```

- 规范的代码风格

正确的注释和注释风格：

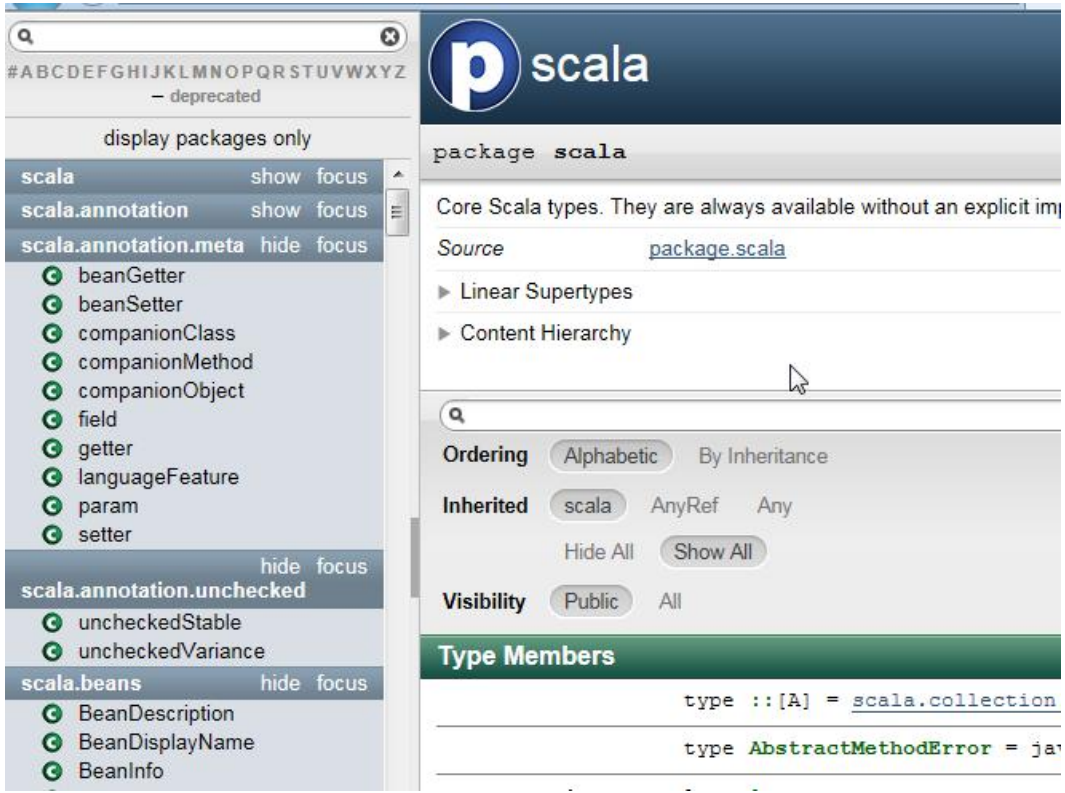
带看Scala源码

正确的缩进和空白

1) 使用一次tab操作，实现缩进,默认整体向右边移动，时候用shift+tab整体向左移
2) 或者使用 ctrl + alt + L 来进行格式化 [演示]
3) 运算符两边习惯性各加一个空格。比如：2 + 4 * 5。
4) 一行最长不超过80个字符，超过的请使用换行展示，尽量保持格式优雅

- **Scala官方编程指南**

√



课后练习

- 本章知识回顾

- Scala语言的sdk是什么?
- Scala环境变量配置及其作用。
  配置SCALA_HOME = d:\program
  配置Path = % SCALA_HOME %\bin
- Scala程序的编写、编译、运行步骤是什么? 能否一步执行?
  编写：就是使用工具，开发scala程序
  编译：就是将 .scala 文件编译成 .class [scalac]
  运行：就是使用scala 来将.class文件加载到jvm并运行
  可以直接运行.scala, 但是速度慢. cmd>scala xx.scalaq

- Scala程序编写的规则。//基本上规范和java类似。但是语句后面不需要加上分号
- 简述：在配置环境、编译、运行各个步骤中常见的错误。

谢谢！ 欢迎收看！