Scala

An Introduction

Introduction :

Let us get introduce d to the characters:

Hi everyone! myself Arnold and today I am going to give a brief introduction about "Scala".

Hello Arnold! We all are the Java developers and it is a pleasure to know about a new programming language"Scala".

Pain points of Java and C++ :

Do you have any pain points or difficulty in Java/C++?

Yes, I have faced some pain points they are:
a).Error handling
b).Writing a lot of codes,
for example: sometimes we have to write 20 lines in Java programming.

Pain points of Java and C++ :

I shall give you an example:

If you are doing custom sorting and you have class detecting C sorting policy, you have to write the class for it and inside it write anonymous function. For doing the word count you have to write 120 lines in Java.

Programming Paradigm :

What are the programming paradigm do you encourage as a developer?

- Java encourages object oriented programming.

- Java does not have pointers. It handles the memory for you.

Programming Paradigm :

- In C++, people have to write a lot of platform related code, so the Java designers developed a common layer on all the platforms,the"Java RunTime" it takes the byte code and runs the program for you.

- Whatever Java fail to address Scala is catching up them, it took the best part of Java and few more features.

ONLINE LEARNERS

Scala Designer :

Do you know who designed Scala?

Scala Designer :

The Scala was designed by Martin Odersky, who was also the designer of Java Generics.

Generics :

What are the advantages of Generics?

????

Generics Advantages:

The advantage of Generics are:

- It helps to eliminate the boiler plate codes; you need not have to write same function again.

Ex: consider a fine function; it takes an array and single element as two arguments. The function iterates through array and if it matches the element then it returns the position or element otherwise it returns as null.

Functions of Generics :

What is the function of Generics?

Functions of Generics :

- The algorithm or the implementation is independent of your data statement. Generic tries to solve. Scala also takes the same thought.

Functions of Generics :

- What is the disadvantage of Java?

it is more dependent.

Functions of Generics :

- Java is heavily dependent on the object oriented paradigm a lot, which is not a very good representation or modelling of real built object as things change state a lot.

Can you give an example?.

Functions of Generics :

Take the example of Zookeeper of Hadoop, in which two nodes try to communicate each other with zookeeper as the mediation. This is not a good representation using object oriented system, because the state keeps on mutating frequently. AS node might go down and come up at different time state and may hold different set of values.

Nice! Thank you.

Features of Scala :

Let us point out the features of Scala:

- Scala maintains the backend compatibility.

- Reduces the verbosity.

- It reduces the obstructions.

- It had to be dependent on JVM, and then only you can run the code which is compatible with Java.

Implications of Scala :

What are the implications of Scala?

BDD :

What is a BDD?

Scala has a good BDD support. A BDD is a type of testing which is like unit testing or integration testing, where you test the system against its features.

BDD :

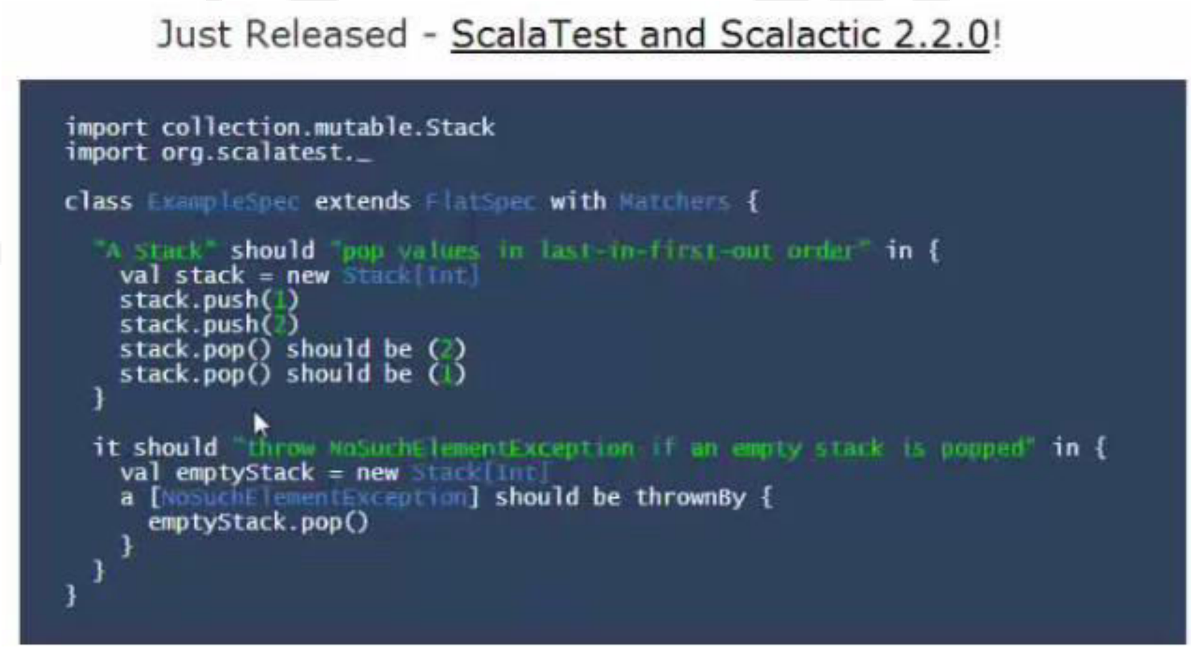
Let me explain you a scenario:

If you are testing calculator, you specify the test case like this, If I have"1+1" then it should print"2", you literally specify the system in terms of its behaviour and expected output instead of writing the test code.

Behavioural driven testing is almost invented by Scala. It is the biggest productivity hack in Agile programming.

BDD :

See the below image for BDD



Just Released - ScalaTest and Scalactic 2.2.0!

```
import collection.mutable.Stack
import org.scalatest._

class ExampleSpec extends FlatSpec with Matchers {

  "A Stack" should "pop values in last-in-first-out order" in {
    val stack = new Stack[Int]
    stack.push(1)
    stack.push(2)
    stack.pop() should be (2)
    stack.pop() should be (1)
  }

  it should "throw NoSuchElementException if an empty stack is popped" in {
    val emptyStack = new Stack[Int]
    a [NoSuchElementException] should be thrownBy {
      emptyStack.pop()
    }
  }
}
```

Compile and Interpreted languages :

What is the difference between Compile and Interpreted languages?

Compile and Interpreted languages :

The differences are:

•A compile language is a visual basic and quite old language which will compile page by page. Ex: Java, C++. People use these compile languages for prototyping.

•Compile language goes through the compiler in one pass converts the machine code and builds in executive. Where as an interpreted language does it on the fly.

Compile and Interpreted languages :

What is the advantage of Compile language?

It is fast, where as Interpreted languages are comparatively.

Compile and Interpreted languages :

What is the advantage of interpreted languages?

In case of interpreted languages writing code, building and compiling and deploying cycle is very less.

Ex: Java, Python, Ruby, Java script.

ONLINE LEARNERS

Compile and Interpreted languages :

Is there any language which can be used interpreted as well compiled?

•For prototyping I can use the interpretation mode and for deploying I can use the compiling mode, Scala do this work. It can be used for interpretation and compiling. These things improve your productivity.

Functional programming:

What is a Functional programming?

Scala is a proponent of functional programming. It supports objects but it brings a lot of new concepts called functional programming.

Ex: what if you treat a function like you treat a variable; can I have an array of functions? Can I pass function as an argument to another function? Can I return a function?

I cannot do any of the above possibilities in Java.

- Your functions are treated the same as your variables are treated.
- I can have an array of functions.
- I can pass function as the parameter to another function; make a function to return a function.

Functional programming:

what is the advantage of all these treatment of function as the first class variable?

It increases the obstructions. These obstructions make your code easy to understand.

Ex: "C++" is an obstruction on "C", if you increase the number of obstructions; it increases the developer's productivity.

Spark:

Let me introduce you all to“Spark”.

Spark extends the popular MapReduce model to efficiently support many types of computations, including interactive queries and stream processing.

•It was designed as it is highly accessible, offering simple APIs in Python, Java, Scala and SQL.

• It can run in Hadoop clusters like AWS, Elastic MR.

Ex:

Interactive Algorithms

Graph Processing

SQL-Spark

Spark:

Why Spark was introduced?

Spark was introduced because :

•In case of Hadoop it is difficult to think everything in terms of"Map" and"Reduce".So spark was invented.

• Spark was developed independently of Hadoop they didn't do it same time.

•Spark guarantees the backend compatibility. Suppose you have written a lot map reduce program, you can still use them and improve using Spark.

Spark Components :

What are the components of Spark?

It has few components built along with it, they are:

1.SQL: you can write native first class queries in SQL directly in Spark, it will compile fast and not have any performance issues.

2.Standard Connectivity: Spark SQL offers connectivity with the standard components like JDBC, ODBC etc.

Streaming :

What is Streaming? What is the difference between batch computing and Stream computing?

Stream computing is real time.

For example: If you go to an e-commerce store and then you are browsing at the new end of sales sections and your websites log detects that this user from this ID has been doing this from 10 minutes. At the backend it processes the log and generates a dynamic coupon and gives it you when you refresh the page next time.

Spark provides streaming built in. You can do the streaming competition without switching frame work.

Streaming :

Spark provides streaming built in. You can do the streaming competition without switching frame work.

- High Fault tolerance
- Integration with HDFS, Flume and Kafka
- It can run in batch and interactive modes

Spark MLib :

Let us know about Spark Mlib:

- It is the collection of common machine learning algorithms.
- It runs 100 times faster than MapReduce
- It can be deployed on existing hadoop clusters.

RDD :

What is RDD?

RDD is the Spark equivalent, which offers much improved interface to distributed file system. It offers better fault tolerance as compared to HDFS, you can do without replications. They act as:

- Basic abstraction of Spark
- Immutable collection of elements which can be operated in parallel
- Higher fault tolerance

RDD :

What is the full form of RDD?

Resilient Distributed Dataset(RDD) which is equivalent to HDFS, but more stable than it.

Spark library can operate in Java, Scala and Python.

ONLINE LEARNERS

This all about a brief introduction to Scala and Spark.
Thanks for attending!

Thanks for giving such an excellent class.

ONLINE LEARNERS

Thank You

Email us – support@intellipaat.com

Visit us - https://intellipaat.com