

Simple Scala Apps

A new paradigm in Scala development

History and Future of programming

Robert "Uncle Bob" Martin

YouTube: Gave a talk on
“The Future of programming”

What are programs?

Assignments to memory

If statements

while loops

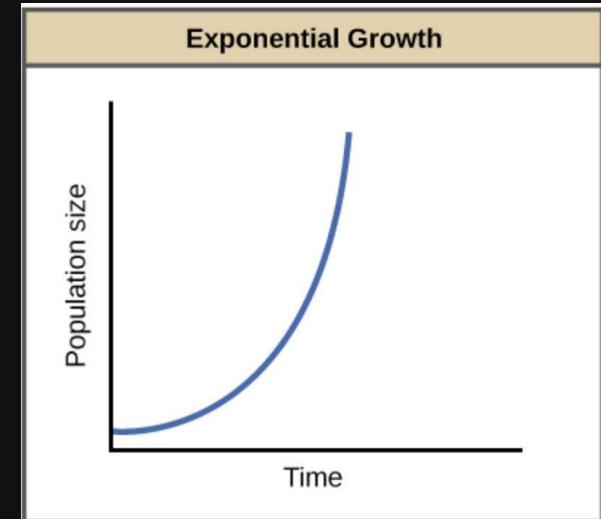
We have 2 problems

Number of computers

Number of programmers

Avg Programmer < 30 years old

50% of workforce < 5 years



Perpetual Inexperience

there will never be enough of us to meet demand

always repeating the mistakes of people before us

Notable Quotes

The function of a good software is to make the complex appear to be simple. – *Grady Booch*

Developer is an organism that turns coffee into code.
— *Unknown*

The purpose of software engineering is to control complexity, not to create it. – *Unknown*

A good programmer is someone who always looks both ways before crossing a one-way street. – *Doug Linder*

Notable Quotes

Testing can only prove the presence of bugs, not their absence. – *Edsger W. Dijkstra*

Any fool can write code that a computer can understand.
Good programmers write code that humans can
understand. – *Unknown*

Debugging is twice as hard as writing the code in the first place. Therefore, if you write the code as cleverly as possible, you are, by definition, not smart enough to debug it. – *Rajanand*

Notable Quotes (and my corallary)

Bad programmers worry about the code.

Good programmers worry about data structures and their relationships. – *Linus Torvalds*

Bad programmers worry about the code.

Good programmers worry about how its deployed and not getting paged at 2 AM. – *David Buschman*

Software Professionalism

We must have discipline

Code must be simple

Apps must be simple

It must be easy to read for young developers

Choose libraries based on their testability

“I will make sure that someone else can cover for me and my code”

-- Bob Martin

Why monoliths came about

Application Frameworks

not many tools around to help with deployment

pre-installed requirements to simplify deployments

wars versus binaries

handle log file tracking

simplified application deployment - in theory

Why monoliths came about

Package Managers

Few early on: deb and rpm for the most part

code wise - Maven was invented

now they are everywhere

brew, pip, npm, pkg, apt, etc

msi and chocolatey for Windows

And now we have containers with Docker and others

MicroApp versus MicroLith

Scaling an app down is harder than it looks

Logging - stdout versus log files/rotation, etc .

Config - env vars and defaults versus complex files

yaml/json/xml files

Execution - single threaded versus thread pools

Builds - simple versus large build pipelines

Jars - Artifactory, gitlab, ivy local

Apps - Assembly, Docker, Native

MicroApp versus MicroLith

Command Line Apps

- read from stdin sometimes

- write output to stdout

- write logging to stderr or not at all

- fire up, do one thing, and shutdown

SBT (or gradle, mvn, or pants)

```
.  
|__ sbt.sbt  
|__ lambda.sbt  
|__ scalac.sbt  
|__ cdk.json  
|__ aliases.sbt  
|__ project  
| |__ Scalac.scala  
| |__ build.properties  
| |__ sbt.sbt  
| |__ BuildInfo.scala  
| |__ Dependencies.scala  
| |__ Version.scala  
| |__ Util.scala  
| |__ plugins.sbt  
|__ scalafix.sbt  
|__ version.txt  
|__ generateVersion  
|__ build.sbt  
src  
|__ test  
| |__ scala  
| |__ cdk  
| | |__ CdkSynth.scala  
| | |__ Common.scala  
| | |__ DockerLambdaStack.scala  
| | |__ PingLambdaStack.scala  
| | |__ VpcStack.scala  
| | |__ ExampleStack.scala  
| |__ me  
| | |__ lightspeed7  
| | |__ examples  
| | | |__ LambdaTest.scala  
| | | |__ TestSuite.scala  
| | | |__ VoterIdsTest.scala  
| | |__ ExampleSuite.scala
```

SBT .scala files -> 5

SBT .sbt files -> 9

Scala CDK program files -> 6

Scala Application files -> 11

```
.  
|__ main  
| |__ scala  
| | |__ me  
| | | |__ lightspeed7  
| | | | |__ examples  
| | | | | |__ SimpleApp.scala  
| | | | | |__ .bsp  
| | | | | |__ scala-cli.json  
| | | | |__ library  
| | | | | |__ Library.scala  
| | | | | |__ AWS.scala  
| | | |__ cli  
| | | | |__ CollectMax.scala  
| | | | | |__ MainApp.scala  
| | | | | |__ CollectMain.scala  
| | | |__ lambda  
| | | | |__ LambdaMain.scala
```

*But
I just want to
run some code*

Introducing Scala CLI

Where to find:

Scala-cli - [<https://scala-cli.virtuslab.org/>]

Github - [<https://github.com/Virtuslab/scala-cli>]

brew install Virtuslab/scala-cli/scala-cli

sdk install scalacli

Introducing Scala CLI

NOT a build tool !

Scala CLI is a command-line tool to interact with the Scala language.

Scala CLI combines all the features you need to learn and use Scala in your scripts, playgrounds and (single-module) projects.

Scala.js, Scala Native, GraalVM compiles, and more

Scala CLI - First Example

```
1 // /////////////////////////////////
2 //> using scala "2.13"
3 //> using platform "jvm"
4 //> using lib "dev.zio::zio:2.0.0"
5 // /////////////////////////////////
6 package me.lightspeed7.examples
7
8 import zio.Console._
9 import zio.{Scope, ZIO, ZIOAppArgs}
10 import java.io.IOException
11
12 object SimpleApp extends zio.ZIOAppDefault {
13
14     val myAppLogic: ZIO[Any, IOException, Unit] =
15         for {
16             _ <- printLine("-" * 100)
17             _ <- printLine("hello world")
18             _ <- printLine("-" * 100)
19         } yield ()
20
21     override def run: ZIO[Any with ZIOAppArgs with Scope, Any, Any] = myAppLogic.exitCode
22 }
```

Scala CLI - First Example

```
1 $ ls -la
2 total 8
3 -rw-r--r--  1 dave  staff   614B Sep  7 07:07 SimpleApp.scala
4
5 $ scala-cli clean .
6 $
7 $ time scala-cli run SimpleApp.scala
8 Compiling project (Scala 2.13.8, JVM)
9 Compiled project (Scala 2.13.8, JVM)
10 -----
11 hello world
12 -----
13 real    0m5.364s
14
15 $ time scala-cli run --scala 3 SimpleApp.scala
16 Downloading 2 dependencies and one internal dependency
17 Compiling project (Scala 3.1.3, JVM)
18 Compiled project (Scala 3.1.3, JVM)
19 -----
20 hello world
21 -----
22 real    0m12.559s
23
24 $
```

Scala CLI - Native Lambda

```
1 package me.lightspeed7.examples.lambda
2
3 import me.lightspeed7.examples.library.{ Library, Request, Response }
4 import zio.lambda.{ Context, ZLambda }
5 import zio.{ Task, ZIO }
6
7 object LambdaMain extends ZLambda[Request, Response] {
8
9   override def apply(event: Request, context: Context): Task[Response] =
10     for {
11       _ <- ZIO.succeed(Console.println(s"Lambda Executing for ${event.toString}"))
12     } yield Library.function(event)
13
14 }
```

Scala CLI - Native Lambda

```
1 $ scala-cli clean .
2 $ scala-cli package --scala 2.13 \
3   --dependency dev.zio::zio-lambda:1.0.0-RC6 \
4   --dependency dev.zio::zio-json:0.3.0-RC11 \
5   --main-class me.lightspeed7.examples.lambda.LambdaMain \
6   --native-image \
7   sbt/src/main/scala/me/lightspeed7/examples/library/Library.scala \
8   sbt/src/main/scala/me/lightspeed7/examples/lambda/LambdaMain.scala \
9   -v -f -o target/scala-cli-lambda -f \
10  -- --enable-url-protocols=http --no-fallback \
11  -H:+ReportExceptionStackTraces
12
13 <snipped>
14
15 $ ls -lh target/scala-cli-lambda
16 -rwxr-xr-x  1 dave  staff    24M Sep  7 18:03 scala-cli-lambda*
17 $
18 $
19 $ ls -lh target/bootstrap
20 -rwxr-xr-x  1 dave  staff    58B Sep  7 18:03 bootstrap*
21 $
22 $ ls -lh target/upload.zip
23 -rw-r--r--  1 dave  staff   7.9M Sep  7 18:03 upload.zip
```

Scala CLI - Docker Lambda

```
1 $ scala-cli clean .
2 $ scala-cli package --scala 2.13 \
3     --dependency dev.zio::zio-lambda:1.0.0-RC6 \
4     --dependency dev.zio::zio-json:0.3.0-RC11 \
5     --main-class me.lightspeed7.examples.lambda.LambdaMain \
6     --docker \
7     sbt/src/main/scala/me/lightspeed7/examples/library/Library.scala \
8     sbt/src/main/scala/me/lightspeed7/examples/lambda/LambdaMain.scala \
9     --docker-from amazoncorretto:11-alpine3.15 \
10    --docker-image-repository docker-image \
11    --docker-image-tag ${VERSION}
12
13 <snipped>
14
15 $ docker images | grep docker-image
16 docker-image      1.2.3          9e5260dc1592  About a minute ago  338MB
17 amazoncorretto   11-alpine3.15  283ddb4a3d7a   3 weeks ago       325MB
```

$$338MB - 325MB = 8MB$$

Thanks and Questions

Scala CLI

Where to find:

Scala-cli - [<https://scala-cli.virtuslab.org/>]

Github - [<https://github.com/Virtuslab/scala-cli>]

brew install Virtuslab/scala-cli/scala-cli

sdk install scalacli