

БУДУЩЕЕ JAVA РАЗРАБОТКИ

ИЛИ

ТЯЖЕЛОЕ ПОЛОЖЕНИЕ JAVA В ЭПОХУ КОНТЕЙНЕРОВ.

Ibragimov Ruslan / ruslan@ibragimov.by

JAVA 9

Release date: from 22 september 2016 to january 2016!!!

JSR 294: Modularization of the JDK under Project Jigsaw

JSR 354: Money and Currency API

JEP 222: jshell: The Java Shell (a Java REPL)

Android will support it in **few month!**

JAVA 10

1 year release cycle!

Properties!

Serializable deprecated and removed

Java 10 approx in the first decade of 2017

ХОТЯ НЕТ, ВРУ

Java 9 - ~ 22 sep 2016

Java 8 in Android - ???

Java 10 - ???

TIMELINE

Java 1.0 - 1996

Java 1.1 - 1997

Java 2 - 1998

Java 3 - 2000

Java 4 - 2002

Java 5 - 2004

Java 5 - 2004

Java 6 - 2006

Java 7 - 2011

Java 8 - 2014

Java 9 - 2016

ЕСТЬ ЛИ СВЕТ В КОНЦЕ ТОННЕЛЯ?

FANTOM

1. 2005
2. 4.2k commits
3. 5 forks

GOSU

1. Guidewire Software
2. June 2014?
3. 1.2k commits
4. 2 contributors with ~300 commits
5. 19 forks

CEYLON

1. Red Hat/Jboss
2. April 2011
3. 6.6k commits
4. 4 contributors with ~800+ commits
5. 64 forks

XTEND

1. Eclipse/itemis
2. November 2011
3. 20k commits (xTend + xtext)
4. 5 contributors with ~1000+ commits
5. 87 forks

CLOJURE

1. 2007
2. 2.8k commits
3. 1 contributors with ~1000+ commits
4. 817 forks
5. Bus factor - 1

GROOVY

1. 2003
2. 12.8k commits
3. 4 contributors with ~1000+ commits
4. 600 forks

SCALA

1. Typesafe
2. May 2011
3. 24.7k commits
4. 8 contributors with ~800+ commits
5. 1252 forks

KOTLIN

1. JetBrains
2. July 2011
3. 27k commits
4. 10 contributors with ~900+ commits
5. 297 forks

2011

JAVA 7

- Scala
- Kotlin
- xTend
- Ceylon

Java Virtual Machine Support for Non-Java Languages: Java SE 7 introduces a new JVM instruction that simplifies the implementation of dynamically typed programming languages on the JVM.

НА КАКОМ ЖЕ ЯЗЫКЕ ПИСАТЬ JAVA РАЗРАБОТЧИКУ?

Ceylon, Fantom, Scala, Groovy, Kotlin, xTend, Clojure, Gosu?

КОМЬЮНИТИ, ПОПУЛЯРНОСТЬ

~~Gosu, Fantom, Ceylon, xTend~~

ДИНАМИЧЕСКИ ТИПИЗИРОВАННЫЕ

- просто писать маленькие программы
- просто учить

СТАТИЧЕСКИ ТИПИЗИРОВАННЫЕ

- не тормозит
- поддержка IDE *
- проще поддерживать большие проекты

Clojure

~~Ceylon, Fantom~~, Scala, Groovy, Kotlin, ~~xTend, Clojure, Gosu?~~

SCALA

ПЛЮСЫ

- КОМЬЮНИТИ
- стандартная библиотека
- экосистема

SCALA

МИНУСЫ

- может быть слишком сложной
- много неявного
- тяжело написать хорошую IDE
- время компиляции
- как правильно никто не использует совместно с существующим кодом на Java

GROOVY

ПЛЮСЫ

- КОМЬЮНИТИ
- простота
- экосистема
- gradle
- Барух [@jbaruch](#)

GROOVY

МИНУСЫ

- тормозит
- легко отбивать пальцы и стрелять в ноги
- переживает не лучшие времена (Pivotal)
- Есть только один Барух

ТАК ПОЧЕМУ МЫ НЕ ПИШЕМ НА SCALA И GROOVY?

1. IDE
2. Сложность языка
3. Работа с Legacy
4. Поддержка существующими фреймворками и либами



KOTLIN:

- Полная совместимость/возможность взаимодействия с Java
- Как минимум такой же быстрый как Java
- Backend, Web, Android
- Отличная поддержка в IDE: Ultimate Idea, Community Idea, Eclipse Plugin

А ТАКЖЕ:

- Миграция старого Kotlin кода
- Конвертация из Java кода
- REPL
- Скриптинг

```
package space

fun main(args: Array<String>) {
    println("Hello, world!")
}
```



```
var x = 5  
x += 1
```

```
val a: Int = 1
```

```
val b = 1
```

```
val c: Int  
c = 1
```

```
val text = ""  
    for (c in "foo")  
        print(c)  
    ""
```

```
fun sum(a: Int, b: Int): Int {  
    return a + b  
}  
  
fun sum(var a: Int, b: Int): Int {  
    a+=1  
    return a + b  
}  
  
fun sum(a: Int, b: Int = 42): Int {  
    return a + b  
}  
  
sum(1) // 43
```

```
fun main(args: Array<String>) {  
    if (args.size() == 0) return  
  
    print("First argument: ${args[0]}")  
}
```

```
fun getStringLength(obj: Any): Int? {  
    if (obj !is String)  
        return null  
  
    // `obj` is automatically cast to `String` in this branch  
    return obj.length  
}
```

```
fun cases(obj: Any) {  
    when (obj) {  
        1 -> print("One")  
        "Hello" -> print("Greeting")  
        is Long -> print("Long")  
        !is String -> print("Not a string")  
        else -> print("Unknown")  
    }  
}
```

```
data class Customer(val name: String, val email: String)
val customer = Customer("Space", "space@space.com")
println(customer)

// Customer(name=Space, email=space@space.com)
```



```
fun String.replaceEvilWithAwesome() { ... }  
  
println("Kotlin is evil".replaceEvilWithAwesome())  
// Kotlin is awesome
```

```
val lazyVal: String by lazy {
    println("computed!")
    "Hello"
}

fun main(args: Array<String>) {
    println(lazyVal)
    println(lazyVal)
}
// computed!
// Hello
// Hello
```

THE BILLION DOLLAR MISTAKE

```
String dumb = null;  
dumb.length();
```

```
val a : Int = 1;  
var b : Int? = null;
```

```
var a: String = "abc"  
a = null // compilation error
```

```
var b: String? = "abc"  
b = null // ok
```

```
var b: String? = "abc"  
val l = b.length() // error: variable 'b' can be null  
val l = if (b != null) b.length() else -1
```



```
val l = b?.length() ?: -1
```

```
b?.length()
```

```
bob?.department?.head?.name
```

```
val l = b!!.length()
```

ПРОЕКТЫ

KTOR

```
class HelloApplication(config: ApplicationConfig) : Application(config) {
    init {
        routing {
            get("/") {
                response.status(HttpStatusCode.OK)
                response.contentType(ContentType.Text.Html)
                response.write {
                    appendHTML().html {
                        head {
                            title { +"Hello World." }
                        }
                        body {
                            h1 {
                                +"Hello, World!"
                            }
                        }
                    }
                }
            }
        }
        ApplicationRequestStatus.Handled
    }
    get("/bye") {
        response.sendText("Goodbye World!")
    }
}
```

KOTLIN.HTML

```
html {  
    body {  
        div {  
            a("http://kotlinlang.org") {  
                target = ATarget.blank  
                +"Main site"  
            }  
        }  
    }  
}
```

SPEK

```
class TaxCalculatorSpecs: Spek() { init {  
    given("Tax rate calculator with default locale settings") {  
        val taxRateCalculator = TaxRateCalculator()  
        on("calculating the rate for an income of 200 and an average change of 10 per se  
            val value = taxRateCalculator.calculateRate(200, 10)  
            it("should result in a value of 300") {  
                assertEquals(300, value)  
            }  
        }  
    }  
}}
```


EXPOSED

```
object Users : Table() {
    val id = varchar("id", ColumnType.PRIMARY_KEY, length = 10)
    val name = varchar("name", length = 50)
    val cityId = integer("city_id", ColumnType.NULLABLE, references = Cities.id)

    val all = id + name + cityId
    val values = id + name + cityId // The columns required for insert statement
}

object Cities : Table() {
    val id = integer("id", ColumnType.PRIMARY_KEY, autoIncrement = true)
    val name = varchar("name", 50)

    val all = id + name
    val values = name
}
```

EXPOSED

```
var db = Database("jdbc:h2:mem:test", driver = "org.h2.Driver")

db.withSession {
    create (Cities, Users)

    val minskId = insert (Cities.values("Minsk")) get Cities.id
    insert (Users.values("ruslan", "Ruslan", minskId))
}
```

```
CREATE TABLE Cities (  
  id INT PRIMARY KEY AUTO_INCREMENT NOT NULL,  
  name VARCHAR(50) NOT NULL)  
  
CREATE TABLE Users (  
  id VARCHAR(10) PRIMARY KEY NOT NULL,  
  name VARCHAR(50) NOT NULL,  
  city_id INT NULL)  
  
INSERT INTO Cities (name) VALUES ('Minsk')  
INSERT INTO Users (id, name, city_id)  
VALUES ('andrey', 'Andrey', 1)
```

LEARN KOTLIN

- **Kotlin Reference**
- **Webinar: Functional Programming with Kotlin** (22 october)

LEARN KOTLIN

- **Book: Kotlin in Action** (Still in progress)
- **Book: Kotlin for Android Developers** (Still in progress)

LEARN KOTLIN

- **JavaBy/awesome-kotlin**

LEARN KOTLIN

- Slack Chat 461+ users
- Gitter Ru Chat 43+ users

DEMO

JAVA PROFESSIONALS BY MEETUP #7

 **October 20, 2015**

 Imaguru (4, Fabriciusa str.)



7:00 pm



Java for Internet of Things.

How is things going in reality?

Dmitry Ogievich, Aliaksandr Salnikau, Minsk



8:30 pm



Low latency in Java

Andrew Kondratovich, Minsk