Avast

Android lecture 1

Android history, Kotlin

Agenda

- History of Android OS
- Development options
- Kotlin

Why mobile development

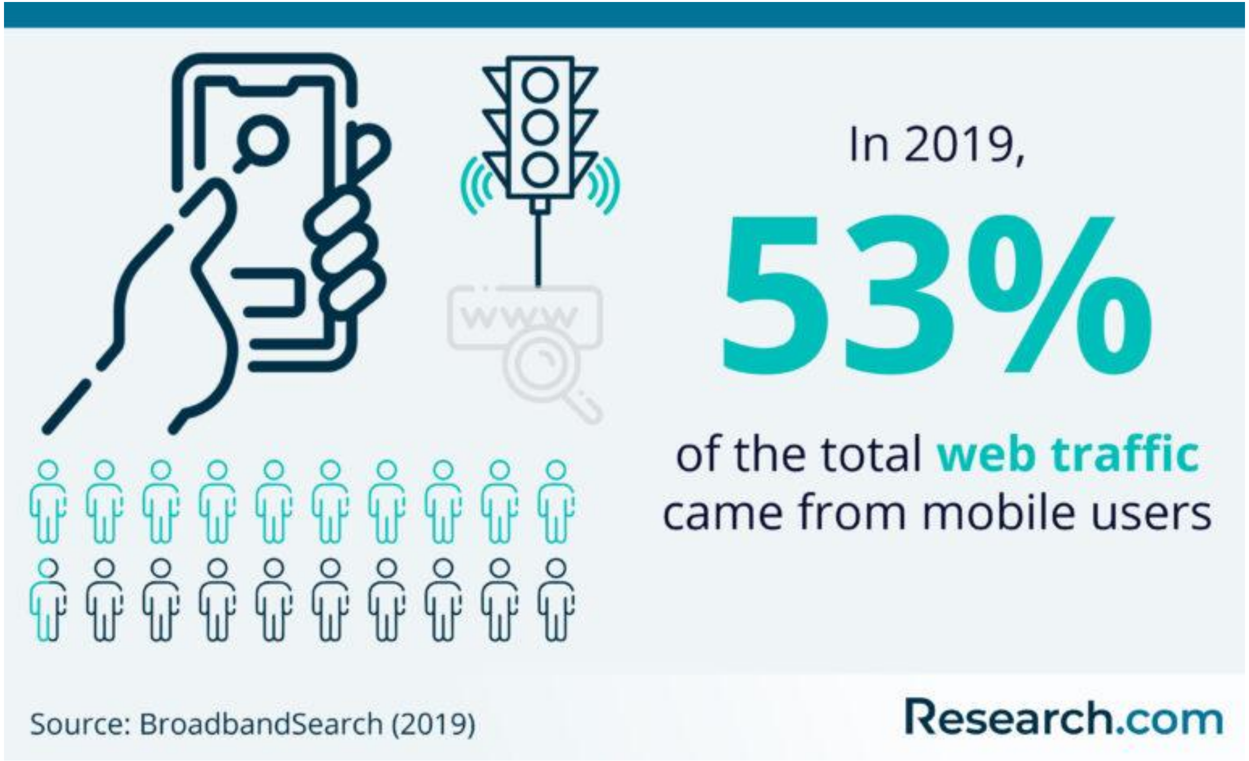
Why mobile matters

Ubiquitous

Quick

Convenient

Why mobile matters



In 2019,

53%

of the total **web traffic** came from mobile users

Source: BroadbandSearch (2019)

Research.com

https://research.com/software/mobile-vs-desktop-usage

Mobile devices specifics

- Not enough computation power
- Changing state - screen orientation
- Unstable network connection
- Small battery

Avast

History

Mobile devices history

- Psion, handhelds
- April 2000 - Pocket PC 2000 (Windows Mobile)
- October 2003 - Android Inc.
  - Intended as software for digital cameras
- July 2005 - Android Inc. acquired by Google
- November 5, 2007 - Open Handset Alliance
  - Consortium of technology companies (Google), manufacturers (HTC, Sony,...), carriers (Sprint, T-mobile,..) and chipset makers (Qualcomm, …)
- June 29, 2007 - iPhone
- October 22, 2008 - T-mobile G1 (HTC)
- May 2009 - Windows Mobile 6.5
- November 8, 2010 - Windows phone 7 release
- 2017 - Microsoft discontinued Windows 10 Mobile development

T-Mobile G1 (HTC Dream)

Nokia N95 (2007), iPhone 2G(2007), T-mobile G1(2008), HTC HD2(2009)

Android

- Linux based OS
  - No bash, no access to root (by default)
  - GNU C library (glibc) replaced by bionic
- Open source
  - https://source.android.com/
  - Just the OS, not the Play store or Google play services

Android versions

- 1.0 - 23 September 2008 (API 1)
- 1.1 - 9 February 2009 (API 2)
- 1.5 - Cupcake 27 April 2009 (API 3)
- 1.6 - Donut 26 September 2009 (API 4)
- 2.0 -2.1 - Eclair October 2009 (API 5 - 7)
- 2.2 - Froyo May 2010 (API 8)
- 2.3.x - Gingerbread December 2010 (API 9 - 10)

Android versions

- 3.x - Honeycomb February 2011 (API 11 - 13) - Tablet only
- 4.0.x - Ice cream sandwich October 2011 (API 14-15)
- 4.1 - 4.3 - Jelly Bean July 2012 (API 16 - 18)
- 4.4 - Kitkat October 2013 (API 19)
- 4.4W - Kitkat for wearables (API 20)
- 5.0 - 5.1 - Lollipop November 2014 (API 21 - 22)
    - Material design
    - ART runtime - Just In Time vs Ahead Of Time
- 6.0 - Marshmallow October 2015 (API 23)
    - Runtime permissions
    - Doze mode

Android versions

- 7.0 - Nougat (API 24) - August 2016
  - Multi window
  - Doze on the go
  - Vulkan API
  - Quick settings tiles
- 7.1 Nougat (API 25) - October, 2016
  - App shortcuts

Android versions

- 8.0 Oreo (API 26) August, 2017
  - Picture in picture
  - Notification channels
  - Custom fonts and downloadable fonts
  - Autosizing textview
  - Multi display support
  - Permissions
    - Fix granting permission to whole group
  - Some packages from java 8
    - java.time
    - java.nio.file
    - java.lang.invoke
  - Project treble
    - Sony, Nokia, OnePlus

Android versions

- 8.1 Oreo (API 27) December 5, 2017
  - Neural Network API
  - Programmatic safe browsing actions
  - Video thumbnail extractor
  - Wallpapers color API
  - Fingerprint updates
    - New error codes
  - Cryptography updates
    - Prefer Conscrypt over Bouncy castle

Android versions

- 9.0 Pie (API 28) August, 2018
  - Display cutout
  - Notifications - messaging
  - Multi camera support
  - ImageDecoder
  - Animation
    - GIF, WebP animated images
  - HDR VP9, HEIF and Media APIs

Android versions

- 10 (API 29) September, 2019
    - Gesture navigation
    - Smart replies
    - Dark theme
    - Foldable devices
    - More privacy control

Android versions

- 11 (API 30) September, 2020
  - Chat bubbles
  - Notification history
  - One time permission
  - Permission auto-reset
  - 5G detection API

Android versions

- 12 (API 31) October 4, 2021
  - Material You - design refresh
  - Splash screen API
  - Bluetooth permissions
    - Scan for nearby devices doesn’t require location permission anymore

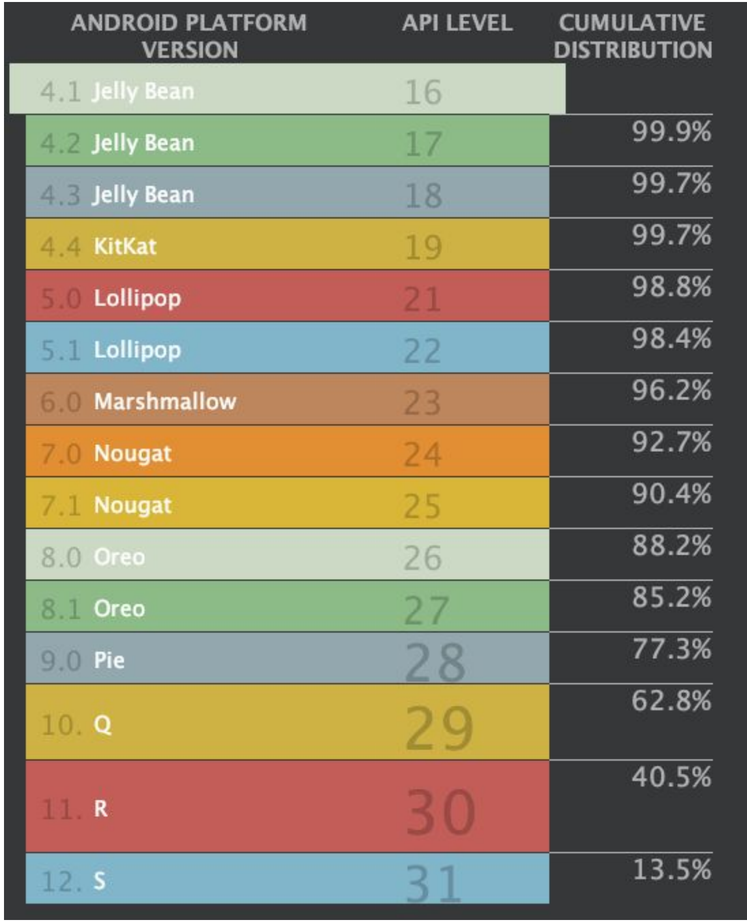
Android versions

- 12L (API 32) March, 2022
    - Optimize for large screens and foldables

Android versions

- 13 (API 33) August, 2022
  - Notification permission
  - Nearby wifi devices permission

Android today - versions



ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.1 Jelly Bean	16	
4.2 Jelly Bean	17	99.9%
4.3 Jelly Bean	18	99.7%
4.4 KitKat	19	99.7%
5.0 Lollipop	21	98.8%
5.1 Lollipop	22	98.4%
6.0 Marshmallow	23	96.2%
7.0 Nougat	24	92.7%
7.1 Nougat	25	90.4%
8.0 Oreo	26	88.2%
8.1 Oreo	27	85.2%
9.0 Pie	28	77.3%
10. Q	29	62.8%
11. R	30	40.5%
12. S	31	13.5%

https://developer.android.com/about/dashboards/
https://gs.statcounter.com/android-version-market-share/mobile-tablet/worldwide
https://techjury.net/blog/android-market-share/#gref
https://www.statista.com/statistics/921152/mobile-android-version-share-worldwide/
https://www.appbrain.com/stats/top-android-sdk-versions

Android

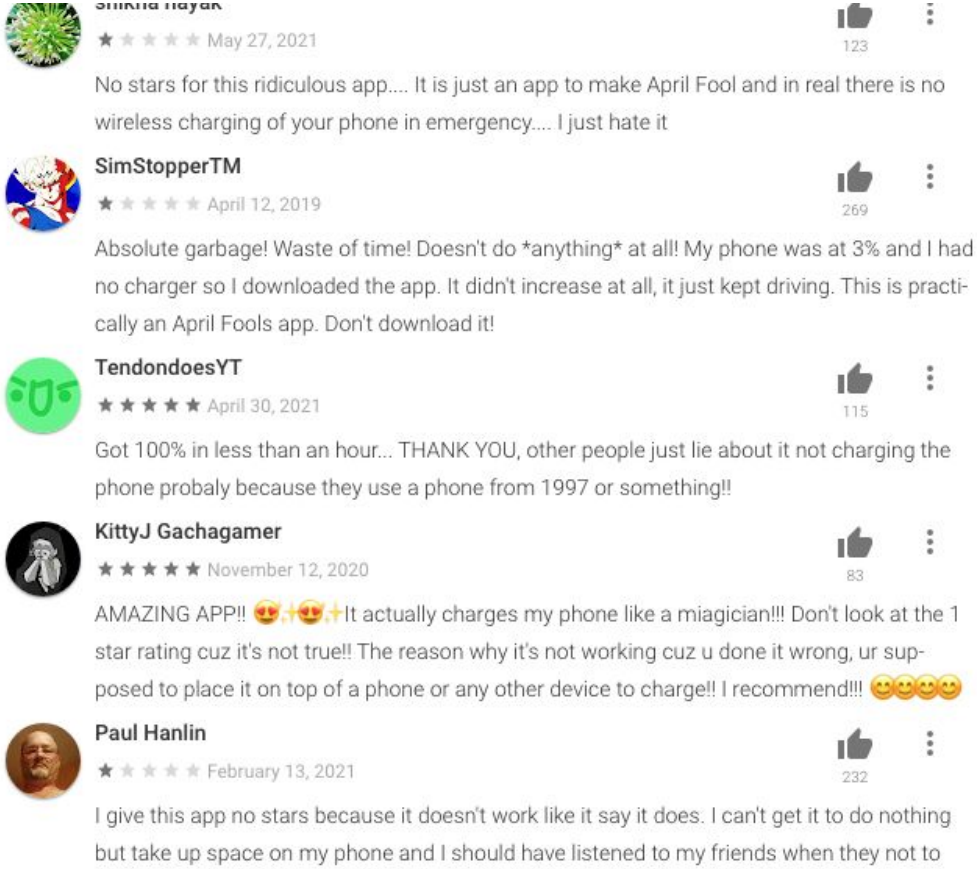
- Android - phones and tablets
- Wear OS - smartwatch
  - Extended notification center
- Android TV - televisions, consoles
  - Native TV app experience for IPTV
- Chrome OS devices
- Android auto - cars
  - Mirroring driving optimized UI to car multimedia system
- Android automotive
  - Standalone OS in cars
- Android things (deprecated January 2022)
  - Platform for IoT
- Google glass
  - Glass explorer program - retired
  - Now glass for enterprise - to be used in logistic, manufactures… https://www.google.com/glass/start/

Android ecosystem

- Apps are available on Google Play store
- Paid and free apps
- Subscription
- In-app purchases
- Android users not like to pay for apps => a lot of advertisements
- Available in most countries

Android - cons

- Fragmentation, slow upgrades, manı
- Low-end devices
- Low-quality apps in Google play, mal
  - Wireless Charger Simulator



Shikha nayak
★☆☆☆☆ May 27, 2021
123
No stars for this ridiculous app.... It is just an app to make April Fool and in real there is no wireless charging of your phone in emergency.... I just hate it

SimStopperTM
★☆☆☆☆ April 12, 2019
269
Absolute garbage! Waste of time! Doesn't do *anything* at all! My phone was at 3% and I had no charger so I downloaded the app. It didn't increase at all, it just kept driving. This is practically an April Fools app. Don't download it!

TendondoesYT
★★★★★ April 30, 2021
115
Got 100% in less than an hour... THANK YOU, other people just lie about it not charging the phone probaly because they use a phone from 1997 or something!!

KittyJ Gachagamer
★★★★★ November 12, 2020
83
AMAZING APP!! 😍✨😍✨It actually charges my phone like a miagician!!! Don't look at the 1 star rating cuz it's not true!! The reason why it's not working cuz u done it wrong, ur supposed to place it on top of a phone or any other device to charge!! I recommend!!! 😊😊😊😊

Paul Hanlin
★☆☆☆☆ February 13, 2021
232
I give this app no stars because it doesn't work like it say it does. I can't get it to do nothing but take up space on my phone and I should have listened to my friends when they not to

Android - pros

- Open source
- Lot of users
- Lot of apps
- Freedom (keyboard, launcher, …)
- Developers one time fee 25$ (Apple 99$ annual)
  - Alpha, beta channels, staged rollout
  - Basic crash reporting
  - Pre-launch reports
  - Android vitals
- Dev tools available for Linux, Windows and Mac
- A lot of OSS libraries
  - Retrofit, OkHttp, Dagger, Flipper, RxJava
- Nexus/Pixel devices

Avast

Android - security

- Root user is not available by default
  - Rooting devices to get more features, often void warranty
- Permissions
  - Until API-23 before installation “all or nothing”
  - Since API-23 permissions are requested at runtime
  - Empty data or crash, depends on TargetSDK
- Bouncer
  - Service that scan google play store for malicious apps

Development

Development options

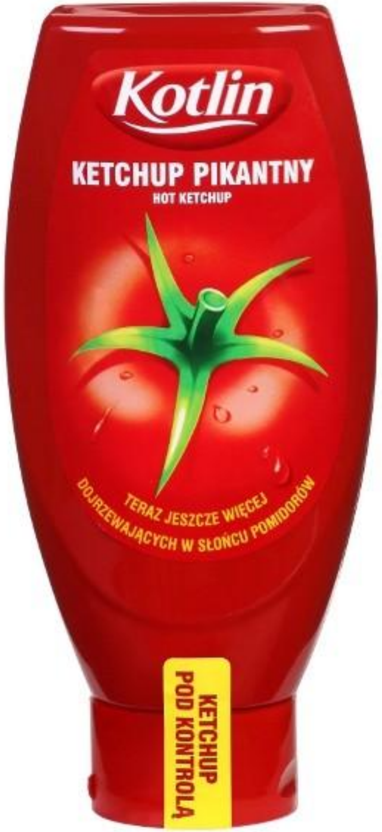
- App-like mobile web
- Other language frameworks (Xamarin, flutter.io)
- WebView based frameworks (PhoneGap)
- Native
  - Kotlin - officially announced support for android development on Google IO 2017
  - Java
  - C/C++ (mainly games and libraries)

Development tools

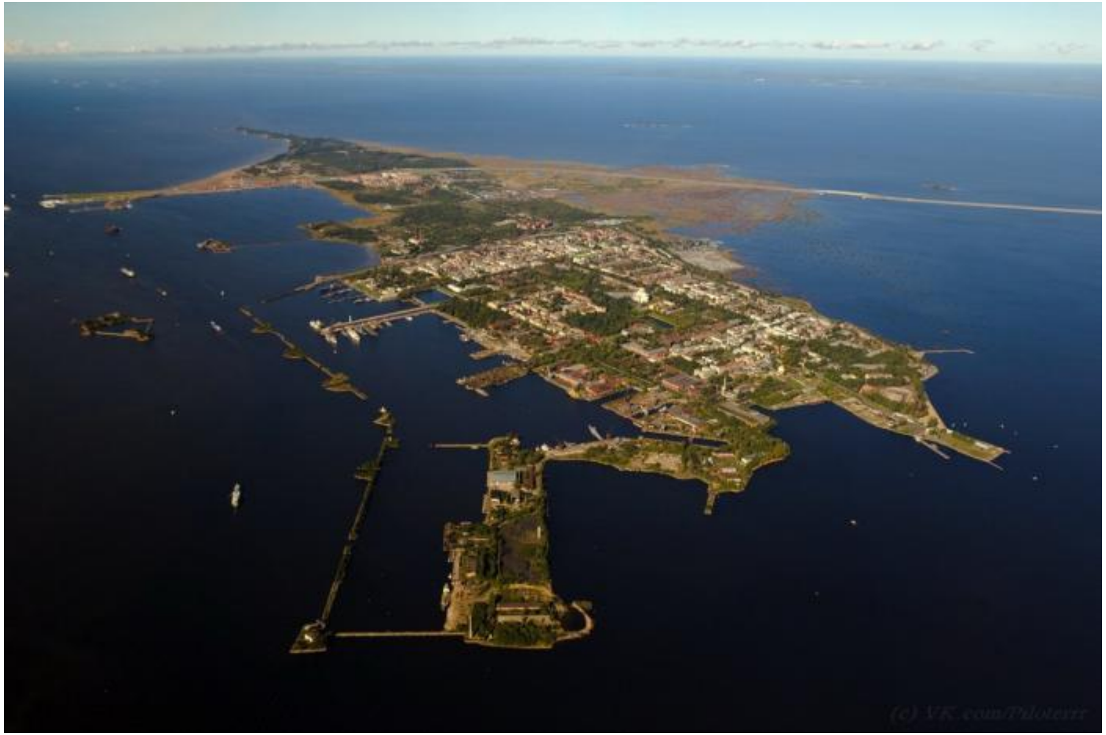
- Android studio
- SDK
  - ADB
  - Lint, UiAutomatorViewer
  - Emulator
  - Aapt, aidl, dx, (jack & jill - java 8, now deprecated)
  - D8
  - Proguard/R8
- NDK (C/C++ development)
  - Cross compile
  - Native libraries
- Gradle
  - Build system
  - Use Groovy (kotlin) as DSL
  - Extensible

Kotlin

Ketchup


Kotlin
KETCHUP PIKANTNY
HOT KETCHUP
TERAZ JESZCZE WIĘCEJ
DOJRZEWAJĄCYCH W SŁOŃCU POMIDORÓW
KETCHUP
POD KONTROLĄ

Island



(c) VK.com/Pilotsrr

Kotlin

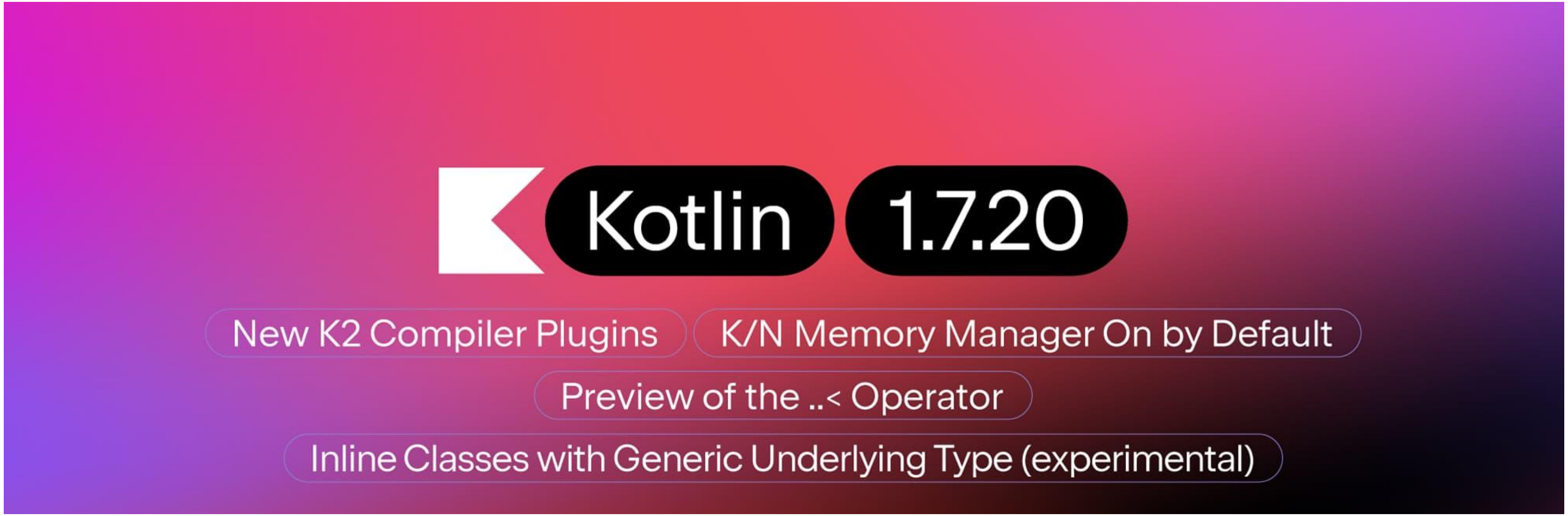
https://kotlinlang.org/

History

- Developed by JetBrains
  - No other language except scala did not have features, they needed
- July 2011 unveiled Project Kotlin
- February 2012 open sourced
- February 2016 Kotlin v1.0
- Google IO 2017 first-class support on Android
- March 2017 Kotlin 1.1 with coroutines
- November 2017 Kotlin v1.2
- September 2018 1.3-RC
- April 2020 1.3.72
- September 2020 1.4.0

https://kotlinlang.org/lp/10yearsofkotlin/

Current version


Kotlin
1.7.20
New K2 Compiler Plugins
K/N Memory Manager On by Default
Preview of the ..< Operator
Inline Classes with Generic Underlying Type (experimental)

Why Kotlin

- Concise
  - Reduce boilerplate code
- Safe
  - Avoid NullPointerExceptions
- Interoperable
  - Works well with legacy java code
- Tool-friendly
  - Support in IDEs (IntelliJ IDEA, Android Studio, Eclipse, Netbeans, standalone compiler)

Kotlin - concise

- Data classes
- Named parameters
- Operator overloading
- Smart casting
- Default parameter values
- Extension functions
- Lambda expressions
- String interpolation

Data classes

- Methods generated by compiler
  - equals()/hashCode()
  - toString()
  - copy()
  - Only for fields defined in primary constructor

```
data class Person(
    val name: String,
    val surname: String,
    val street: String,
    val buldingNumber: Int
)
```

```
val bruceWayne = Person(
    name = "Bruce",
    surname = "Wayne",
    street = "Wayne Manor",
    buldingNumber = 1
)

val robin = bruceWayne.copy(name = "Robin")
```

Operator overloading

```kotlin
data class ComplexNumber(
   val real: Double,
   val imaginary: Double
) {
   operator fun plus(increment: ComplexNumber): ComplexNumber {
       return ComplexNumber(real + increment.real, imaginary + increment.imaginary)
   }
}
```

Smartcast

```kotlin
fun smartCastDemo(x: Any) {
    when (x) {
        is Int -> println("Integer value abs(x): ${x.absoluteValue}")
        is String -> println("String x.toLowerCase: ${x.toLowerCase(Locale.getDefault()) }")
        else -> println("x is not Integer or String")
    }
}
```

Kotlin - safe

- Null safety
- Immutable by default
  - List vs. MutableList, Map vs. MutableMap

Kotlin - interoperability

- Java
- Swift
- Kotlin multiplatform

Avast

Thank you
Q&A

Feedback is appreciated

lukas.prokop@avast.com

Please use [mff-android] in subject