Десятая независимая научно-практическая конференция «Разработка ПО 2014»

23 - 25 октября, Москва

2014 CEE-SEC(R) Разработка ПО

Kotlin на Android

Лёгкий способ перестать программировать на Java.

Илья Рыженков

JetBrains

А что, Java хороший язык!

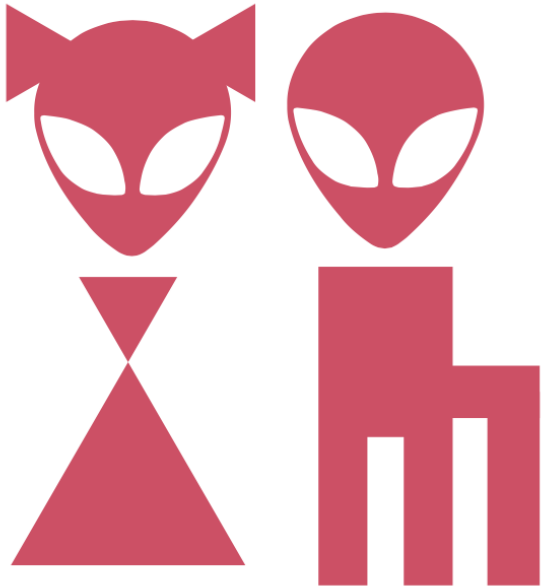
- Одна парадигма.
- Богатый опыт.
- Вагон инструментов.
- Превосходная совместимость.

Оборотная сторона

- Одна парадигма.
- Богатый опыт.
- Вагон инструментов.
- Превосходная совместимость.

Мобильные приложения

Прототипы, реактивность, данные, лаконичность, фрагментация, сервисы.



Kotlin

Язык программирования.

Сделано в JetBrains.

Kotlin

- Для промышленной разработки.
- Для краткого и понятного кода.
- Для безопасности и совместимости.
- Для удобства и продуктивности.

Kotlin @ Android

- Работает с Java 6, Android SDK и Android Studio.
- Маленькая библиотека (800Кб).
- Не создаёт лишних объектов.
- Не требует переписывания всего проекта.

Функционально-императивный.

```
fun String.compact(predicate: (Char) -> Boolean): String {
    val sb = StringBuilder()
    var lastMatch = false
    for (ch in this) {
        val thisMatch = predicate(ch)
        if (lastMatch != thisMatch)
            sb.append(ch)
        lastMatch = thisMatch
    }
    return sb.toString()
}

fun main(args: Array<String>) {
    val compacted = "Functional Meets Imperative"
            .compact { it.isJavaLetter() }
            .filterNot { it.isWhitespace() }
    println(compacted) // FMI
}
```

Расширяемый.

```
trait Service {
    fun execute(request: Request, success: (RequestResult) -> Unit)
}

fun Service.executeAsync(request: Request,
                         success: (RequestResult) -> Unit) {
    val service = this
    val task = object : AsyncTask<Request, Int, Unit>() {
        override fun doInBackground(vararg request: Request): Unit {
            service.execute(request.single(), success)
        }
    }
    task.execute(request)
}


fun userCode(service: Service) {
    val request = createRequest()

    service.execute(request) { result ->
        // handle result
    }

    service.executeAsync(request) { result ->
        // handle result
    }
}
```

Компактный.

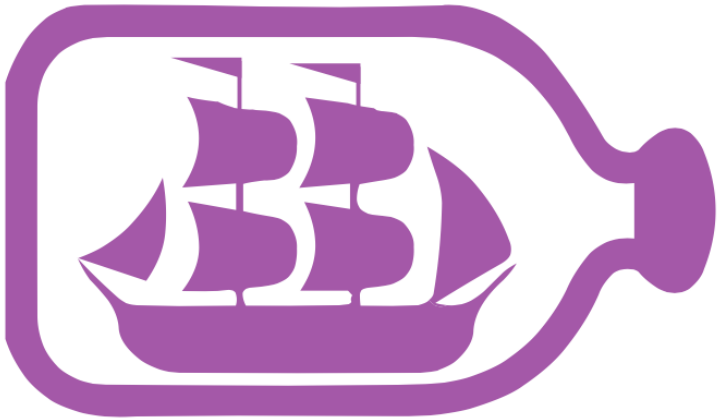
```
class Person(val name: String, val age: Int)
class Document(val title: String, val author: Person)
class User(val person: Person, val role: Role)

enum class Role {
    User
    Administrator
}

fun User.canEdit(document: Document) = when (role) {
    Role.User -> document.author == person
    Role.Administrator -> true
}

val Person.documents : Stream<Document>
    get() = Repository.allDocuments().filter { it.author == this }


fun User.editableDocuments() = person.documents.filter { canEdit(it) }
```

Безопасный.

```
class Item {
    fun update() {}
}

fun tryCreate(): Item? = null
fun create(): Item {
    if (true)
        return null
    else
        return Item()
}

fun consume(item: Item) {}
fun use() {
    // item cannot be null
    val item = create()
    item.update()
    consume(item)

    // try with nullable
    val nullableItem = tryCreate()

    nullableItem.update()
    nullableItem?.update()
    consume(nullableItem)

    if (nullableItem != null) {
        nullableItem.update()
        consume(nullableItem)
    }
}
```

Android.

- UI, сервисы, data flow.
- Компактно, быстро.
- Больше, чем Java 8.
- Сегодня.



Координаты.

- http://kotlinlang.org
- Twitter: @project_kotlin
- ETA: 2015