Inside Kotlin/Wasm
(or how your language could benefit from new proposals)

Zalim Bashorov
@bashorov
Kotlin

Programming Language

- Statically typed
- Concise and expressive
- Pragmatic but elegant
- Imperative and functional
- Null-safety
Kotlin

Ecosystem

- Multiple target platforms
  - JVM
  - Native (iOS, Linux, Windows, ...)
  - JavaScript
  - WebAssembly
- JVM based tooling
- Compiler plugins (Experimental)
Kotlin/Wasm new compiler

- Fast Compilation
- Incremental Compilation (later)
Kotlin/Wasm new compiler

- Fast Compilation
- Incremental Compilation (later)
- Great integration with hosts
Kotlin/Wasm new compiler

- Fast Compilation
- Incremental Compilation (later)
- Great integration with hosts
- Good Interop with hosts

```kotlin
import kotlinx.browser.document
import kotlinx.dom.appendText

fun main() {
    val p = document.createElement("p")
    p.innerHTML = "Hello World!"
    document.body?.appendChild(p)
}
```
Kotlin/Wasm new compiler

- Fast Compilation
- Incremental Compilation (later)
- Great integration with hosts
- Good Interop with hosts
- Small binaries

```kotlin
import kotlinx.browser.document
import kotlinx.dom.appendText

fun main() {
    val p = document.createElement("p")
    p.innerHTML = "Hello World!"
    document.body?.appendChild(p)
}
```
WebAssembly proposals we are using

- Reference Types
- Exception Handling
- Extended Name Section
- Typed Function References
- Garbage Collection
Trial for Wasm GC in Chrome 112.*

🎉🎉🎉

Turn on Wasm GC for your site!

https://zal.im/tryWasmGC
WebAssembly proposals we are interested in:

- **Interop**
  - Component Model
  - Multiple Memory
  - ESM integration
  - JS types

- **Performance**
  - Threads
  - Call-tags

- **Size**
  - Stringref
  - JS PI
  - Stack switching
Kotlin/Wasm deep dive
Class kotlin.Any

```kotlin
open class Any {
    open fun equals(other: Any?): Boolean
    open fun hashCode(): Int
    open fun toString(): String
}
```
Class kotlin.Any

struct Any
- vtable
- itables
- typeInfo
- hashCode

struct Any_vtable
- equals
- hashCode
- toString

fun Any.equals
fun Any.hashCode
fun Any.toString
Class extension

```kotlin
struct Any
    vtable
    itables
    TypeInfo
    hashCode
    baz

struct Foo
    vtable
    equals
    hashCode
    toString
    bar

struct Any_vtable
    equals
    hashCode
    toString

struct Foo_vtable
    equals
    hashCode
    toString

fun Any.equals
    ...

fun Any.hashCode
    ...

fun Any.toString
    ...

fun Foo.toString
    ...

fun Foo.bar
    ...
```
Access to fields

Local:

```
struct Foo
{
    vtable
    itables
    typeInfo
    hashCode
    baz: i32 = 42
}
```

Stack:

Instructions:
Access to fields

Local:  
- d: Foo

Stack:
- <ref>: Foo

Instructions:
- local.get $d

struct Foo
- vtable
- itables
- typeInfo
- hashCode
- baz: i32 = 42
Access to fields

Local:
- `d:Foo`

struct `Foo`
- `vtable`
- `itables`
- `typeInfo`
- `hashCode`
- `baz: i32 = 42`

Stack:
- `42`

Instructions:
- `local.get $d`
- `struct.get $baz`
Virtual Method call

Local:
- `d: Foo`
- `vtable`
- `itables`
- `typeInfo`
- `hashCode`
- `baz: i32 = 42`

Stack:
- `d: Foo`

Instructions:
- `fun Foo.bar`
- `equals`
- `hashCode`
- `toString`
- `bar`

...
Virtual Method call

Local:
- d: Foo

struct Foo
- vtable
- itables
- typeInfo
- hashCode
- baz: i32 = 42

struct Foo_vtable
- equals
- hashCode
- toString
- bar

Stack:
- <ref>: Foo
- <ref>: Foo

Instructions:
- local.get $d
- local.get $d

fun Foo.bar
...
Virtual Method call

Local:

struct Foo
d: Foo

Foo:

vtable
itables
typeInfo
hashCode
baz: i32 = 42

Foo_vtable:
equals
hashCode
toString
bar

Instructions:

Stack:

<ref>: Foo
dllocal.get $d
<ref>: Foo_vtable
local.get $d
struct.get $vtable

fun Foo.bar

...
Virtual Method call

Local:
- d: Foo

struct Foo
- vtable
- itables
- typeInfo
- hashCode
- baz: i32 = 42

struct Foo_vtable
- equals
- hashCode
- toString
- bar

Stack:
- <ref>: Foo
- <ref>: fun Foo.bar

Instructions:
- local.get $d
- local.get $d
- struct.get $vtable
- struct.get $bar

fun Foo.bar
...

Virtual Method call

Local:
- d: Foo

struct Foo
- vtable
- itables
- typeInfo
- hashCode
- baz: i32 = 42

struct Foo_vtable
- equals
- hashCode
- toString
- bar

fun Foo.bar

Stack:
- 27

Instructions:
- local.get $d
- local.get $d
- struct.get $vtable
- struct.get $bar
- call_ref
Static Method calls

call $getTime
Static Method calls

<table>
<thead>
<tr>
<th>i32.const 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>i32.const 2</td>
</tr>
<tr>
<td>call $add</td>
</tr>
</tbody>
</table>
Interfaces

struct Any

struct Foo

vtable
itables
typeInfo
hashCode
baz

struct ITables_1

Timer
Logger
...

fun Foo.start

fun Foo.stop

fun Foo.value

struct Timer_itable

start
stop
value
...

...
## Different kind of calls

<table>
<thead>
<tr>
<th>Static function</th>
<th>Class / virtual method</th>
<th>Interface method</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>call $getTime</code></td>
<td><code>local.get $d</code></td>
<td><code>local.get $d</code></td>
</tr>
<tr>
<td></td>
<td><code>local.get $d</code></td>
<td><code>local.get $d</code></td>
</tr>
<tr>
<td></td>
<td><code>struct.get $vtable</code></td>
<td><code>struct.get $vtable</code></td>
</tr>
<tr>
<td></td>
<td><code>struct.get $bar</code></td>
<td><code>struct.get $bar</code></td>
</tr>
<tr>
<td></td>
<td><code>call_ref</code></td>
<td><code>call_ref</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>struct.get $itables</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>ref.cast $Itables_1</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>struct.get $Timer</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>struct.get $value</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>call_ref</code></td>
</tr>
</tbody>
</table>
Strings
Strings are everywhere!
String naïve impl

struct String

cchars: CharArray
String optimized

```headers
struct String {
    chars: CharArray,
    left: String|null,
    length: i32,
}
```
String optimized: example

(chars: length=4, left=null)

(chars: length=8, left)

(chars: length=4, left=null)

(chars: length=10, left)

(F E S T)

(2 3)
String optimized: example

- **chars**: H A C K
  - **length**: 4
  - **left**: null

- **chars**: F E S T
  - **length**: 4
  - **left**: null

- **chars**: 2 3
  - **length**: 10
  - **left**: null

- **.chars**: H A C K F E S T 2 3
String transfer to JS

chars
left=null
length=3

String.fromCharCode.apply(null, ...

Memory

H I ! ...

JS Uint16Array

Copy

String.fromCharCode.apply(null, ...
String transfer from JS

JS String: "Hoi"
Memory: ... H o i ...
String literals

- **Intern** all compile time string constants
- Store in **data section**
  - Fast at runtime (array.new_data)
- Runtime **cache** for string literals
- Optimize for Latin1
Stringref proposal 🚀

Preliminary results:

- 60 times faster on interop microbenchmarks
- Up to 3 times faster on DBMonster benchmark

1. https://zal.im/wasm/dbmonster/
Kotlin/Wasm applications today and tomorrow
Jetpack Compose (Android)

source: https://github.com/android/compose-samples
Compose Multiplatform for Desktop

fun CircleOfCirclesWithSettings() = application {
    Window(title = "Circles with Settings") {
        MaterialTheme {
            var settings by remember {
                mutableStateOf(Settings())
            }
            Row(modifier = Modifier.padding(5.dp)) {
                Canvas(...) {
                    if (settings.drawOuterOrbit) {
                        outerOrbit(settings)
                    } else {
                        innerOrbit(settings)
                    }
                }
                SettingsPanel(settings) { settings = it }
            }
        }
    }
}
Compose Multiplatform Kotlin/Wasm Demo

- Draw inner orbit
- Draw outer orbit
- Orbits radius: 300.0
- Steps for inner orbit: 3
- Steps for outer orbit: 5
- Inner orbit circles radius: 100.0
- Outer orbit circles radius: 200.0
- Random: 0.10
private fun DrawScope.outerOrbit(settings: Settings) {
    for (ang in 0.0..359 step settings.stepsOuterOrbit) {
        drawOffsetCircle(
            ang,
            settings.orbitRadius.toFloat(),
            settings.outerCirclesRadius.toFloat(),
            randomCoefficient = settings.randomCoefficient
        )
    }
}

private fun DrawScope.innerOrbit(settings: Settings) {
    for (ang in 0.0..359 step settings.stepsInnerOrbit) {
        drawOffsetCircle(
            ang,
            settings.orbitRadius.toFloat(),
            settings.innerCirclesRadius.toFloat(),
            strokeWidth = 1.3f,
            zeroColor = Color.Gray
        )
    }
}

private fun DrawScope.drawOffsetCircle(
    angle: Int,
    offsetRadius: Float,
    circleRadius: Float,
    strokeWidth: Float = 1.1f,
    randomCoefficient: Double = 1.0
) {
    val rad = convertDegreesToRadians(angle)
    // The offsets create minor glitches in the overall
    // that make it look as if these are hand-drawn
    val offsetX = (offsetRadius * cos(rad) + Random.nextFloat())
    val offsetY = (offsetRadius * sin(rad) + Random.nextFloat())
    drawCircle(
        color = Color.Gray,
        radius = circleRadius
    )
}
Beyond the browsers
KoWasm ([kowasm.org](http://kowasm.org))

by Sébastien Deleuze ([@sdeleuze](https://twitter.com/sdeleuze))

Server-side and full stack development with Kotlin and WebAssembly, by leveraging WASI and Component Model
Deployment

Today

```
node --experimental-wasi-unstable-preview1 --experimental-wasm-gc app.mjs
```

app.wasm

```
docker run -dp 8080:8080 --runtime=io.containerd.wasmtime.v1
# or --runtime=io.containerd.wasmedge.v1
--platform=wasi/wasm32
sdeleuze/app
```

Tomorrow

```
docker run -dp 8080:8080 --runtime=io.containerd.wasmtime.v1
# or --runtime=io.containerd.wasmedge.v1
--platform=wasi/wasm32
sdeleuze/app
```

Cloud/Edge

Kubernetes
What’s next?

- General availability of Wasm GC in browsers (soon)
- Kotlin/Wasm
  - Developer experience (DX)
  - Compose for Web with Kotlin/Wasm
  - Target standalone runtimes (Wasmtime, WasmEdge, etc.)
Thank you!

Zalim Bashorov
@bashorov
Interface method

Local: d: Any

struct Foo
  vtable
  itables
  typeInfo
  hashCode
  baz

struct ITables_1
  Timer
  Logger

struct Timeritable
  start
  stop
  value

fun Foo.start
fun Foo.stop
fun Foo.value
**Interface method**

Local:

- `d: Any`

struct `Foo`

- `vtable`
- `itables`
- `typeInfo`
- `hashCode`
- `baz`

struct `ITables_1`

- `Timer`
- `Logger`

struct `Timer_itable`

- `start`
- `stop`
- `value`

fun `Foo.start`

fun `Foo.stop`

fun `Foo.value`

Stack:

- `<ref>`: Any
- `<ref>`: Any

Instructions:

- `local.get $d`
- `local.get $d`
Interface method call

Local: d:Any

struct Foo

vtable

itude

typeInfo

hashCode

baz

struct ITables_1

Timer

Logger

...

struct Timeratical

fun Foo.start

fun Foo.stop

fun Foo.value

Stack:

<ref>: Any

<ref>: Any

local.get $d

local.get $d

struct.get $itables

Instructions:

local.get $d

local.get $d

struct.timer $itables

...
Interface method call

Local:
- d: Any

from struct Any
- TypeInfo
- hashCode
- baz

struct Foo
- vtable
- itables

struct ITables_1
- Timer
- Logger
- ...

struct Timer_itable
- start
- stop
- value

struct
- Any
- ITables_1

Stack:
- local.get $d
- <ref>: Any
- <ref>: ITables_1
- struct.get $itables
- ref.cast $Itables_1

Instructions:
- local.get $d
- ...
- fun Foo.value
- ...
- ...
- ...
- ...
- ...
Interface method

Local:
- d: Any

Struct Foo
- vtable
- itables

Struct ITables_1
- Timer
- Logger
...
struct Timer

itable

Interface method

typeInfo

hashCode

baz

from struct Any

vtable

itables

typeInfo

hashCode

baz

struct Foo

struct ITables_1

struct Timer

Logger

...

struct get $itables

ref.cast $ITables_1

struct.get $Timer

start

stop

value

struct.get $value

...
Interface method call

struct Timer
itable

Function call:
foo
start
stop
value

Local:
struct Foo
d: Any

Stack:
local.get $d
local.get $d
struct.get $itables
ref.cast $ITables_1
struct.get $Timer
struct.get $value
call_ref
Browsers Setup/Requirements

Chrome 112+
Origin Tail [https://zal.im/tryWasmGC](https://zal.im/tryWasmGC)

Chrome 110+ – in [chrome://flags](https://chrome://flags)

WebAssembly Garbage Collection
Firefox 112+ – in [about:config](https://about:config)
- javascript.options.wasm_function_references
- javascript.options.wasm_gc
Introducing KoWasm
https://kowasm.org

Server-side and full stack development with Kotlin and WebAssembly, by leveraging WASI and Component Model
Introducing KoWasm
https://kowasm.org
Visio

Component Model bindings

warg repository

Browser

Cloud/Edge

WASI

WasmGC
All modules:

- **core**
  This module provides core KoWasm infrastructure.

- **wasi**
  This module exposes a Kotlin/Wasm API inspired from WASI Preview2 in order to expose low level WASI Preview1 capabilities.

- **web**
  This module allows to create a web server and defines HTTP routes using a proper DSL.
Memory allocator to bridge WasmGC with linear memory

```kotlin
@WasmImport("wasi_snapshot_preview1", "clock_res_get")
private external fun rawClockResGet(
    arg0: Int,
    arg1: Int,
): Int

fun clockResGet(id: ClockId): Timestamp {
    withScopedMemoryAllocator { allocator ->
        val pointer = allocator.allocate(0)
        val returnCode = rawClockResGet(id.ordinal, pointer.address.toInt())
        return if (returnCode == 0) {
            (Pointer(pointer.address.toInt().toUInt())).loadLong()
        } else {
            throw WasiError(Errno.values()[returnCode])
        }
    }
}
```
Search Everywhere Double Shift
Go to File Ctrl+Shift+N
Recent Files Ctrl+E
Navigation Bar Alt+Home
Drop files here to open them
WIT record to Kotlin

```kotlin
record person {
    name: string,
    age: option<u32>,
}
```

```kotlin
data class Person(
    val name: String,
    val age: UInt? = null
)
```
WIT variant to Kotlin

```kotlin
variant filter {
    all,
    none,
    some(list<string>),
}
```

```kotlin
sealed interface Filter {
    object All : Filter
    object None : Filter
    class Some(val value: List<String>): Filter
}
```
WIT result to Kotlin

```kotlin
enum code {
    too-big,
    too-small,
}

class CodeException(val code: Code) : Exception()

fun a(input: String): String {
    if (input.isNotEmpty()) return input
    else throw CodeException(Code.TOO_SMALL)
}
```
fun main() {
    fun greet() = listOf("Hello", "Hallo", "Hola", "Servus").random()

    renderComposable("greeting") {
        var greeting by remember { mutableStateOf(greet()) }
        Button(attrs = { onClick { greeting = greet() } }) {
            Text(greeting)
        }
    }
}
Fullstack rendering with Compose for Web and Kotlin/Wasm

- **Compose Component**
  - **Compiler plugin**
    - **Client-side binder**
      - **Browser**
      - **Initial rendering**
      - **Server**
      - **Site generator**
      - **Resume**
Kotlin

Mobile

Server-side

Ktor

Build