Kotlin Sugar High

Discussing about taste



Dominik Zmitrowicz

Professionaly fighting with imposter syndrome since 2009.

Pretending to know Kotlin for 2 years.

What is this about

- Kotlin
- Sugar
- Nightmares



What is this about

11 kinds of programmers

- one way enjoyers
- muh freedom!

What is this about

11 kinds of programmers

- one way enjoyers
- muh freedom!
- I am sure I can write it in one line

I will be exaggerating

Not about

- performance
- bytecode
- coroutines

Sealed classes

```
sealed class Animal(val favoriteFood: String)
class Dog(favoriteFood: String) : Animal(favoriteFood)
class Horse(favoriteFood: String) : Animal(favoriteFood)
```

Sealed classes

```
sealed class Demon()
class Cat() : Demon()
class Oni() : Demon()
class Rakshasa() : Demon()
```

Sealed classes

Nihil novi

Already exists in other languages. Also in Java.

Sub-classes in same package

Mocking issues

Mockking is fine.

But still cool!

Enumy taste

Pattern matching

Unlock the power of sealed classes

And much more

Extension functions

fun ExternalClass.missingFunction() {}

Cool way to not die from lib poisoning

Very usefull for writing transformers/converters

Top level

We don't need no util classes

Constructors and inits

Fancy shmancy

- primary
- secondary
- inits

Operator overloading

Scoped functions

Strings

Tricky tricky nullability

Using java classes

Compiler is not perfect

Bang bang!

DSL

Thank you Questions?

Hope not

Sources

Kotlin docs
6 magic sugars to keep your kotlin codebase happy
Kotlin pitfalls and how to avoid them

Slides and code

https://github.com/dogrizz/kotlin-sugar-high