

ackee
nedu

We know how
We know how
We know how

OK

ackee

nedu

nedu





Jakub Olejník

iOS Dev @ Ackee



Async programming

Obsah

- ❏ Otázky z minula?
- ❏ Closures
- ❏ Combine
- ❏ Swift Concurrency

Closures

Basic přístup pro asynchronní programování

“Až dokončíš, co potřebuješ udělat, zavolej mi
tenhle closure s výsledkem”

Basic přístup pro basic operace

Špatně se čte při zřetězení více async operací

Closures

Live coding 🎉

Reactive streams

Producer  operations  consumer

Operations mohou obsahovat nastartování dalších producerů, transformace hodnot...

Producer generuje eventy - typicky values, completions a failures

Streamy lze kombinovat - musí mít kompatibilní errory

Combine

Framework pro reaktivní programování

Inspirováno Rx (RxSwiftem)

Basic přístup pro basic operace

Easy řetězení async operací

Combine terminologie

Producer ↔ **Publisher**

Consumer ↔ **Subscriber**

A spousta operátorů...🕶️

Publisher events

 Output / Value

 Completion
 finished
 failure

 Cancel

Publisher types

Různé typy publisherů se různě chovají k eventům

Nemusí generovat values, failures

Mají omezený (typicky 1) nebo naopak neomezený počet values

...

Subscribers

Subscriber = každý koho zajímají eventy ze streamu

sink(receiveCompletion:receiveValue)

assign(to:on:)

Simple publishers

Empty

žádná hodnota, jen completion, žádný error

Just

okamžitě jedna hodnota a completion, žádný error

Future

někdy jedna hodnota a completion, nebo error

Basic operators

Map

transformace value na jinou value

TryMap

failable map - transformace stringu na první znak
- nelze pro empty string

HandleEvents

side effecty - logování, startování nezávislých
publishers

Kombinace publishers

Merge

“trychtýř” - kombinuje hodnoty z více streamů do jednoho (musí sedět typy hodnot i errorů)

CombineLatest

spojuje poslední hodnoty z více streamů do jednoho (musí sedět typy errorů)

Type erasure

Kombinací streamů vznikají příšerné typy

Výsledný typ závisí na pořadí operátorů

Type erasure z každého publishera udělá
AnyPublisher<Output, Failure>

Vhodné např. do protokolů

Custom publishers

Publisher je protokol - úplně vlastní implementace je možná

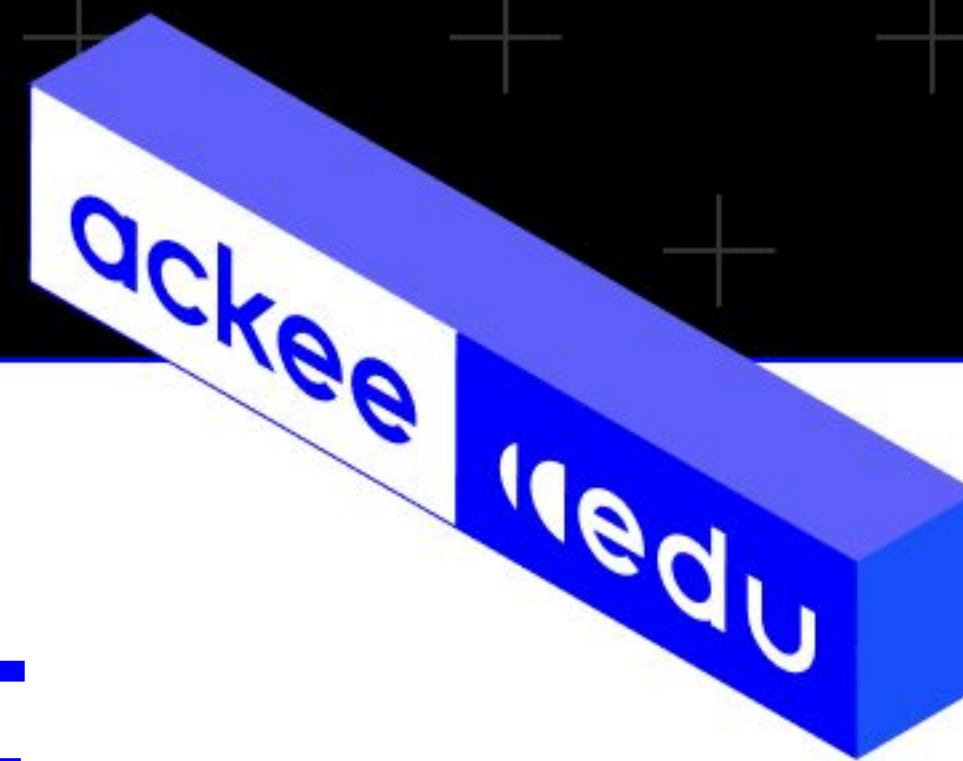
Vlastní implementace by neměla být potřeba

Pomocí *PassthroughSubject*, popř. *CurrentValueSubject* by mělo být možné naimplementovat jakékoliv chování

Swift Concurrency

Až příští cviko 🙄

#sorryjako



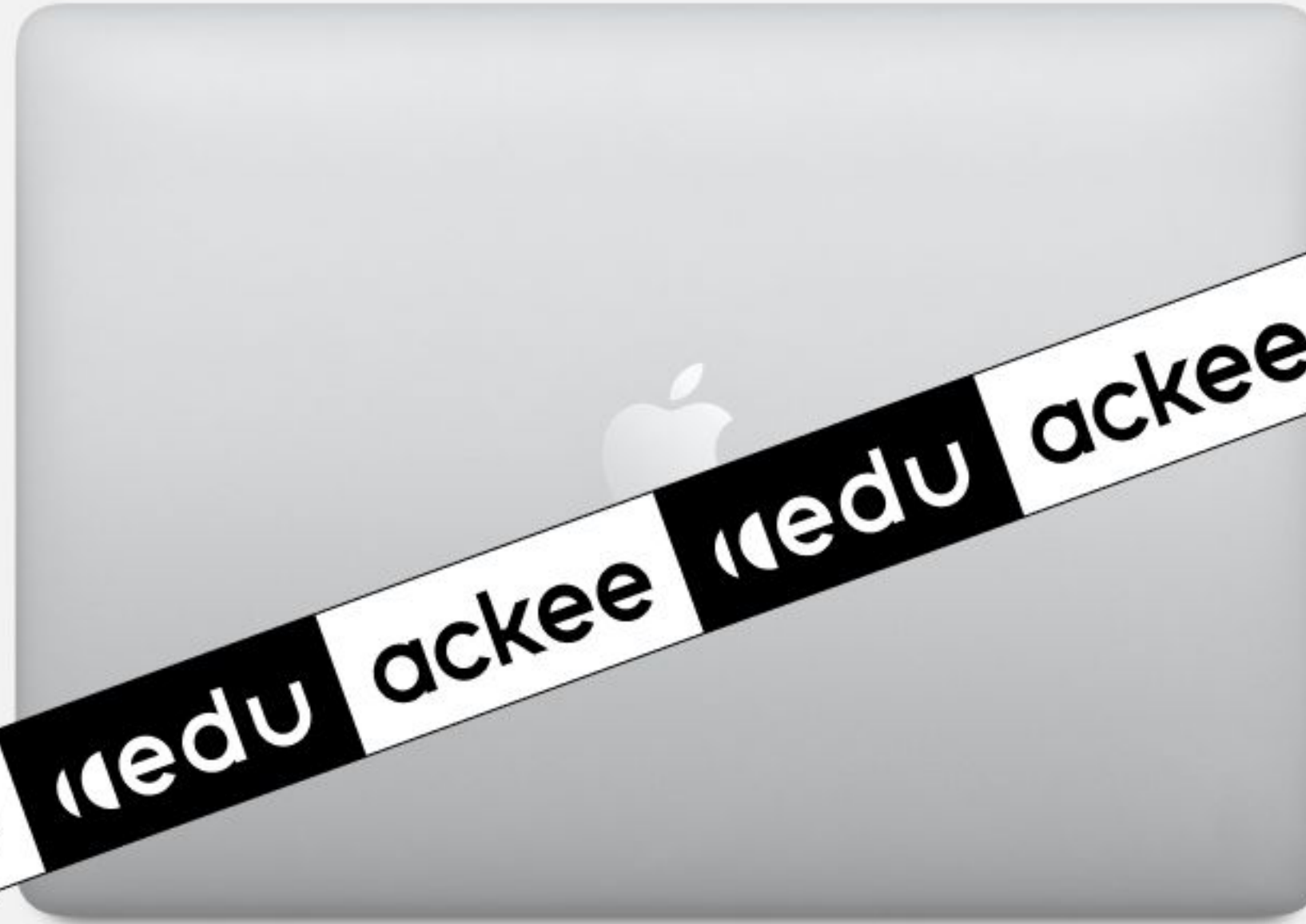
Contact

Jakub Olejník

iOS Dev @ Ackee

jakub.olejnik@ackee.cz

Thank you
Thank you
Thank you



We know how
We know how
We know how

