

# PORÓWNIANIE WŁAŚCIWOŚCI I WYDAJNOŚCI SWIFT I OBJECTIVE-C

# CEL

Prezentacja języka Swift i porównanie go z Objective-C

# CEL

Czy Swift jest szybszy od Objective-C? Dlaczego?



WŁAŚCIWOŚCI

# SWIFT VS. INNE JĘZYKI

Podstawowe cechy

Typy generyczne

Typy wartościowe i referencyjne

Domknięcia

Leniwość

Programowanie funkcyjne

Składnia

Rozszerzenia typów

# SWIFT VS. OBJECTIVE-C

Bezpieczeństwo

Zarządzanie pamięcią

Składnia

Programowanie obiektowe

Sposoby na wywołanie metod

Biblioteki



# TESTY WYDAJNOŚCIOWE

# TESTY

- Przygotowano 12 testów wydajnościowych



# TESTY

- Przygotowano 12 testów wydajnościowych
- Większość z nich testuje wyodrębnione funkcje języka

Przetwarzanie typów  
napisowych

Rekurencja

Zaawansowane typy  
wyliczeniowe

Operacje tablicowe

Wywołania metod

# TESTY

- Przygotowano 12 testów wydajnościowych
- Większość z nich testuje wyodrębnione części języka
- Część z nich otrzymało alternatywne implementacje dla kilku testów

# TESTY

- Przygotowano 12 testów wydajnościowych
- Większość z nich testuje wyodrębnione części języka
- Część z nich otrzymała alternatywne implementacje dla kilku testów
- 7 testów zostało przeanalizowanych za pomocą narzędzia *Instruments*



# WYNIKI

Test	Swift/Objective-C
ArrayInsertion	-78,32%
Fibonacci	-45,84%
BubbleSort	-87,69%
BinarySearchTree - Classic/Enums/Optimized	+129,29% / +103,20% / -72,09%
DispatchMethod - Static vs Message	-75,00%
DispatchMethod - Dynamic vs Message	+112,79%
Dijkstra - Basic/Optimized	-65,33% / -5,14%
WordFrequency	+157,48%
SieveOfEratosthenes	+70,94%
CountLinesWordsChars	+7718,38%
StringConcatenation	-22,98%
RGBHistogram	-90,21%
RC4	+116,41%

# WYNIKI

Test	Swift/Objective-C
ArrayInsertion	-78,32%
Fibonacci	-45,84%
BubbleSort	-87,69%
BinarySearchTree - Classic/Enums/Optimized	+129,29% / +103,20% / -72,09%
DispatchMethod - Static vs Message	-75,00%
DispatchMethod - Dynamic vs Message	+112,79%
Dijkstra - Basic/Optimized	-65,33% / -5,14%
WordFrequency	+157,48%
SieveOfEratosthenes	+70,94%
CountLinesWordsChars	+7718,38%
StringConcatenation	-22,98%
RGBHistogram	-90,21%
RC4	+116,41%

# WYNIKI

Test	Swift/Objective-C
ArrayInsertion	-78,32%
Fibonacci	-45,84%
BubbleSort	-87,69%
BinarySearchTree - Classic/Enums/Optimized	+129,29% / +103,20% / -72,09%
DispatchMethod - Static vs Message	-75,00%
DispatchMethod - Dynamic vs Message	+112,79%
Dijkstra - Basic/Optimized	-65,33% / -5,14%
WordFrequency	+157,48%
SieveOfEratosthenes	+70,94%
CountLinesWordsChars	+7718,38%
StringConcatenation	-22,98%
RGBHistogram	-90,21%
RC4	+116,41%



# WYNIKI

Test	Swift/Objective-C
ArrayInsertion	-78,32%
Fibonacci	-45,84%
BubbleSort	-87,69%
BinarySearchTree - Classic/Enums/Optimized	+129,29% / +103,20% / -72,09%
DispatchMethod - Static vs Message	-75,00%
DispatchMethod - Dynamic vs Message	+112,79%
Dijkstra - Basic/Optimized	-65,33% / -5,14%
WordFrequency	+157,48%
SieveOfEratosthenes	+70,94%
CountLinesWordsChars	+7718,38%
StringConcatenation	-22,98%
RGBHistogram	-90,21%
RC4	+116,41%

# CZY SWIFT JEST SZYBSZY OD OBJECTIVE-C?

W większości przypadków - tak.

# KIEDY SWIFT JEST SZYBSZY?

- Lepiej zoptymalizowane struktury danych



# KIEDY SWIFT JEST SZYBSZY?

- Lepiej zoptymalizowane struktury danych
- Statyczne wywoływanie metod dużo szybsze od wywoływania przez wiadomości

# KIEDY SWIFT JEST SZYBSZY?

- Lepiej zoptymalizowane struktury danych
- Statyczne wywoływanie metod dużo szybsze od wywoływania przez wiadomości
- Wydajniejsza obsługa Automatic Reference Counting

# KIEDY SWIFT JEST SZYBSZY?

- Lepiej zoptymalizowane struktury danych
- Statyczne wywoływanie metod dużo szybsze od wywoływania przez wiadomości
- Wydajniejsza obsługa Automatic Reference Counting
- Lepsze wsparcie dla typów prostych



# KIEDY SWIFT JEST WOLNIEJSZY?

- Obsługa typu napisowego
- Możliwość łatwego wstawiania kodu w języku C do Objective-C

# WNIOSKI

- Kod napisany w Swift jest w większości przypadków szybszy

# WNIOSKI

- Kod napisany w Swift jest w większości przypadków szybszy
- Głównym powodem takiego stanu jest wyeliminowanie słabych punktów Objective-C



# WNIOSKI

- Kod napisany w Swift jest w większości przypadków szybszy
- Głównym powodem takiego stanu jest wyeliminowanie słabych punktów Objective-C
- Nadal dużo zależy od programisty

# WNIOSKI

- Kod napisany w Swift jest w większości przypadków szybszy
- Głównym powodem takiego stanu jest wyeliminowanie słabych punktów Objective-C
- Nadal dużo zależy od programisty
- Praca nie wyczerpuje tematu

DZIĘKUJĘ ZA UWAGĘ