

# Projekt i implementacja kompilatora języka JavaScript na platformę .NET.

Przemysław Gawlas

Opiekun pracy: dr inż. Piotr Błaszyński

# Cel pracy

Celem pracy jest implementacja aplikacji wykonującej kompilację języka JavaScript do kodu pośredniego wykonywalnego na platformie .NET oraz porównania zaimplementowanych w języku JavaScript testowych algorytmów uruchamianych na platformie Node.js oraz .NET.

# Zakres pracy

- ▶ Opis pojęć i technologii wykorzystanych w projekcie oraz nawiązujących do tematu pracy.
- ▶ Analiza języka JavaScript
- ▶ Określenie zakresu implementacji
- ▶ Implementacja aplikacji kompilatora
- ▶ Wybór oraz implementacja algorytmów testujących
- ▶ Przeprowadzenie testów i zebranie wyników
- ▶ Podsumowanie wyników

# Wykorzystane narzędzia

## Implementacja kompilatora:

- ▶ Platforma *.NET core 3.1*
  - ▶ Język C#
- ▶ ANTLR 4.8

## Implementacja kodu testującego:

- ▶ Platforma *Node.js 10.15*
  - ▶ Język JavaScript
- ▶ Platforma *.NET Framework v4.0*
  - ▶ Język C#
  - ▶ Język JScript

# Wykorzystane narzędzia cd.

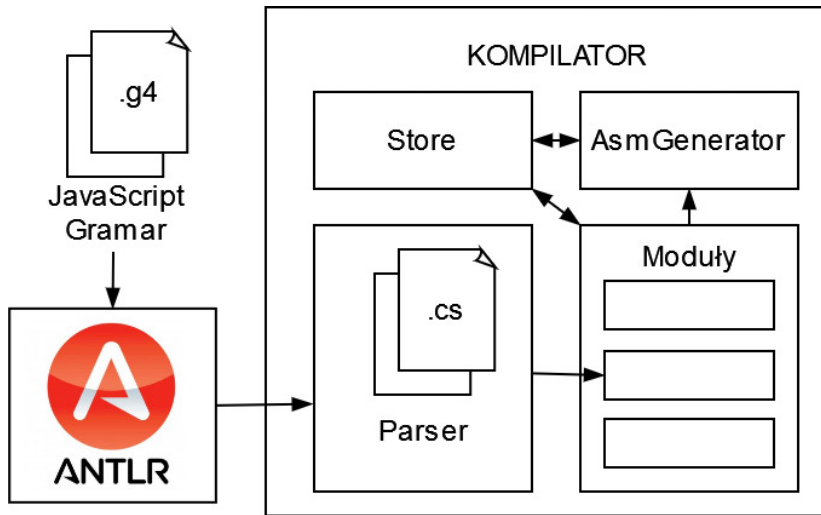
## Narzędzia testujące:

- ▶ Dekompilacja kodu:
  - ▶ Platforma *.NET Framework v4.0*
  - ▶ JetBrains dotPeek 2020.2.1
- ▶ Pomiar czasu i wielkości plików wykonywalnych:
  - ▶ PowerShell 7.1.1
- ▶ Pomiar zużycia pamięci:
  - ▶ JetBrains dotMemory 2020.2.1

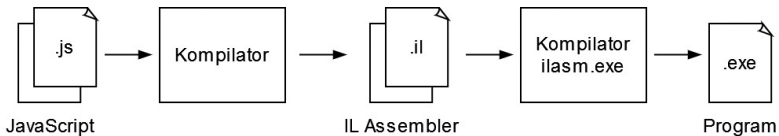
## Narzędzia pomocnicze:

- ▶ Skrypty pomocnicze
  - ▶ PowerShell 7.1.1

# Konstrukcja kompilatora



# Sposób działania



# Przygotowanie testów

## Zakres testów:

- ▶ Testy dla poszczególnych modułów
- ▶ Implementacja oraz przygotowanie algorytmów odnalezionych w Internecie
  - ▶ Algorytm sortowania bąbelkowego
  - ▶ Algorytm liniowego przeszukiwania

## Przygotowanie implementacji algorytmów:

- ▶ Niewielka modyfikacja implementacji.
- ▶ Przygotowanie odpowiedników w języku:
  - ▶ JScript
  - ▶ C#



# Kryteria testów

## Testy modułów

- ▶ Porównanie wyniku wykonania programów
- ▶ Porównanie generowanego kodu assemblera

## Testy algorytmów

- ▶ Pomiar czasu wykonywania programów
- ▶ Pomiar zużycia pamięci
- ▶ Pomiar wielkości plików programów

# Wyniki testów algorytmów

## Pomiar czasu wykonywania programów

# Wyniki testów algorytmów

## Pomiar zużycia pamięci

# Wyniki testów algorytmów

## Pomiar wielkości plików programów

# Podsumowanie

- ▶ Przy analizie wyników poszczególnych testów funkcjonalności zostały wykazane niewielkie różnice przy wyświetlaniu elementów na konsoli.
- ▶ Porównanie generowanego kodu assemblera wykazało słabą optymalizacją stworzonego kompilatora.
- ▶ Testów algorytmów pokazują porównywalne wyniki dla kodu kompilowanego z języka C# oraz lepsze wyniki w porównaniu do kodu JScript.

# Kierunki dalszych prac

## Przykłady:

- ▶ Implementacja pełnej funkcjonalności języka JavaScript
- ▶ Zastosowanie pełnej gamy instrukcji assemblerowych
- ▶ Testy na bardziej złożonych algorytmach lub bibliotekach
- ▶ Optymalizacja generowanego kodu assemblera
- ▶ Dodanie mechanizmów zrównoleglania kodu

# Dziękuję za uwagę