

# Seminarium dyplomowe magisterskie

Prezentacja nr 1

Wojciech Pasternak

# **Wyznaczanie zer wielomianów**

Finding of roots of polynomial

Promotor: dr hab. inż. Robert Janczewski

# Przyjęte rozwiązanie

- Implementacja w C++
- Konsola jako interfejs aplikacji, umożliwiający wprowadzania danych wejściowych
- Testy jednostkowe, napisane z użyciem frameworka wbudowanego w Visual Studio 2015
- Podział funkcjonalności na klasy
- Struktury wielomianu posiadający wspólną klasę bazową i implementujące jej metody abstrakcyjne

# Podział na klasy

- CharConstant i StringManager - klasy zarządzające znakami i łańcuchami znaków
- Number – klasa zarządzająca długimi liczbami
- Parser – klasa tworząca strukturę wielomianu ze strumienia wejściowego
- Polynomial – klasa abstrakcyjna reprezentująca wielomian
- PolynomialMap – klasa reprezentująca wielomian w postaci mapy - trzymanie tylko niezerowych współczynników
- PolynomialVector – klasa reprezentująca wielomian w postaci tablicy - trzymanie wszystkich współczynników

# Główne funkcje

- Podstawowe operacje na wielomianach: dodawanie, odejmowanie, mnożenie, dzielenie, modulo, potęgowanie
- Porównywanie wielomianu
- Obliczanie wartości wielomianu
- Obliczanie pochodnej wielomianu
- Oliczanie NWD wielomianów
- Obliczanie liczby pierwiastków w zadanym przedziale
- Eliminacja pierwiastków wielokrotnych
- Obliczanie wyrazów ciągu Sturma
- Obliczanie pierwiastków w zadanym przedziale

# Problemy implementacyjne

- Konstrukcja intuicyjnego interfejsu, pozwalającego na łatwe wprowadzanie danych wejściowych
- Brak stabilności - obliczenia na wielomianach, zwłaszcza wysokich stopni, wymagają dużej precyzji obliczeń – drobna zmiana może zupełnie zmienić wartość wielomianu i jego pierwiastki
- Wysoka precyzja obliczeń – eliminacja błędów zaokrągleń
- Konieczność użycia biblioteki dla dużych liczb (GNU MP library)
- Optymalizacja podstawowych działań na wielomianach

# Demo

Prezentacja aktualnego interfejsu i działania programu

# Testy jednostkowe

- Unifikacja wielomianu w postaci strumienia wejściowego
- Konwersja strumienia wejściowego na wielomian
- Podstawowe operacje na wielomianach: dodawanie, odejmowanie, mnożenie, dzielenie, modulo
- Obliczanie wartości wielomianu
- Obliczanie pochodnej wielomianu
- Obliczanie liczby pierwiastków w zadanym przedziale
- Eliminacja pierwiastków wielokrotnych
- Obliczanie pierwiastków w zadanym przedziale



# Kryterium oceny problemu

- Poprawne parsowanie wielomianu w oparciu o strumień wejściowy podawany na kilka sposobów
- Porównanie struktur reprezentujących wielomiany
- Wszystkie testy jednostkowe zakończone powodzeniem dla obu struktur
- Porównanie wydajności obu struktur pod względem czasowym dla różnych przypadków wielomianu wejściowego:
  - Wysoki stopień wielomianu
  - Wielomian gęsty
  - Wielomian rzadki
  - Wielomian nieposiadający pierwiastków
  - Wielomian posiadający pierwiastki wielokrotne

# Wykorzystane środowiska i technologie

- Język programowania: C++
- Visual Studio 2015:
- Podział na trzy projekty (moduły) w ramach jednej solucji (biblioteka + aplikacja + unit testy)
- Git (git extension)
- Kod na githubie

# Podsumowanie

- Implementacja działań na wielomianach
- Obliczanie pierwiastków w zadanym przedziale w oparciu o ciąg Sturma
- Tworzenie wielomianu na podstawie standardowego wejścia
- Ponad 200 testów jednostkowych, testujących kilkanaście funkcjonalności
- Porównanie struktur wielomianu pod względem czasowym zarówno dla standardowych wielomianów, jak i dla specyficznych przypadków