

**Wydział Nauk Stosowanych**

**Kierunek studiów: Informatyka (studia magisterskie)**

**Semestr: I**

**INŻYNIERIA OPROGRAMOWANIA**

**Andrzej Musiałek**

**Nr albumu 66784**

**Testy jednostkowe**

DĄBROWA GÓRNICZA 2025



Spis treści

[1 Cel 5](#_Toc198028220)

[2 SŁOWNIK POJĘĆ ORAZ WYBRANE NARZĘDZIA 6](#_Toc198028221)

[3 ARCHITEKTURA ŚRODOWISKA 7](#_Toc198028222)

[4 WDROŻENIE SYSTEMU 8](#_Toc198028223)

[5 WNIOSKI I PODSUMOWANIE 9](#_Toc198028224)

[6 BIBLIOGRAFIA 10](#_Toc198028225)

[7 ZAŁĄCZNIKI 11](#_Toc198028226)

[Załącznik 1. 11](#_Toc198028227)

# Cel ćwiczenia

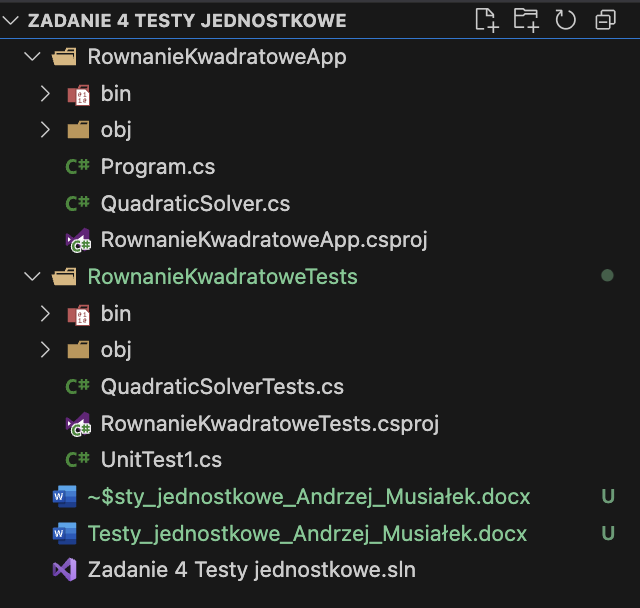
Celem ćwiczenia było zapoznanie się z metodą zapewniania jakości oprogramowania poprzez testy jednostkowe. Szczególny nacisk został położony na testy regresyjne oraz wykorzystanie narzędzia XUnit w języku C#. Zadaniem było stworzenie programu rozwiązującego równanie kwadratowe oraz zestawu testów jednostkowych pokrywających różne przypadki. Dodatkowym elementem było zaznajomienie się z systemem kontroli wersji Git, a także wykonanie zadań na platformie learngitbranching.js.org.

# Opis rozwiązania

## Strutktura projektu

Projekt został podzielony na dwa osobne komponenty:

1. RownanieKwadratoweApp – aplikacja konsolowa zawierająca klasę QuadraticSolver, która oblicza pierwiastki równania kwadratowego.
2. RownanieKwadratoweTests – projekt testowy oparty o framework XUnit, zawierający testy jednostkowe dla różnych przypadków.

Poniżej przedstawiono strukturę katalogów.

Rys. 1. Struktura katalogów (źródło: opracowanie własne)

## Omówienie kodu

W pliku QuadraticSolver.cs zaimplementowano metodę ObliczPierwiastki, która:

* oblicza deltę (b² - 4ac),
* w zależności od jej wartości zwraca odpowiednią liczbę pierwiastków:
  1. brak (0),
  2. jeden (1),
  3. dwa (2),
* metoda obsługuje przypadek błędny, gdy a = 0, rzucając wyjątek ArgumentException.

# Testy jednostkowe

## Omówienie przypadków testowych

Zostały przygotowane cztery grupy testów jednostkowych w QuadraticSolverTests.cs:

1. **Brak pierwiastków rzeczywistych** (delta < 0)

Przykładowe dane: (1, 2, 5)

1. **Jeden pierwiastek rzeczywisty** (delta = 0)

Przykładowe dane: (1, 2, 1)

1. **Dwa pierwiastki rzeczywiste** (delta > 0)

Przykładowe dane: (1, -3, 2)

1. **Niepoprawne dane wejściowe** (a = 0)

Sprawdzenie, czy metoda rzuca wyjątek ArgumentException.

Zastosowano atrybut [Theory] oraz [InlineData] w celu testowania różnych zestawów danych, zgodnie z dobrymi praktykami XUnit. Poniżej przedstawiono zrzuty ekranu z uruchomienia testów.

# 

Rys. 2. Zrzut ekranu po wykonaniu komendy dotnet test (źródło: opracowanie własne)

# Git i GitHub

Poniżej podano link do repozytorium:

<https://github.com/Musiauke/RownanieKwadratoweApp.git>

## Zrzuty ekranów z commitów, brachów i pull requestów

Rys. 3. Zainicjowanie repozytorium (źródło: opracowanie własne)

# WNIOSKI I PODSUMOWANIE

Rys. 4. Pierwszy commit, użycie brach'u oraz wykoanie push'u (źródło: opracowanie własne)

Tutaj należy zamieścić wnioski z realizacji projektu oraz podsumowanie. Rozdział nie powinien być dłuższy niż 2 strony.

# BIBLIOGRAFIA

[1] Redakcja naukowa, M.Torcki, P.Wyrozębski: *Planowanie przebiegu projektów*. Oficyjna wydawnicza, Szkoła Główna Handlowa, Warszawa 2015.

# ZAŁĄCZNIKI

# Załącznik 1.

Tutaj należy zamieścić co ciekawsze fragmenty kodu wraz z opisem słownym. Jeżeli projekt takich nie posiada, rozdział może zostać pominięty.