**Dział III: Testy, diagramy, dokumentacja**

**Uporządkowanie aplikacji RhymersDemo**

1. Utworzono nową gałąź docs:

git checkout -b docs

{miejsce na zdjęcie }

1. Zrefaktoryzowano metodę main w RhymersDemo:

- Użyto Refactor → Extract Method do wyodrębnienia metody testRhymers

- Przeniesiono logikę testową do nowej metody statycznej

{miejsce na zdjęcie refaktoryzacji}

**Reorganizacja projektu**

1. Przeanalizowano klasy pod kątem modyfikatorów dostępu
2. Zidentyfikowano klasy, które nie muszą być publiczne
3. Usunięto zbędne pliki i zoptymalizowano strukturę projektu

{miejsce na zdjęcie struktury projektu}

**Testy jednostkowe**

1. Przeprowadzono walidację projektu testami:

- Wykonano istniejące testy jednostkowe

- Przeanalizowano wyniki testów

- Wprowadzono niezbędne poprawki

{miejsce na zdjęcie wyników testów}

1. Dodano nowe testy jednostkowe:

- Wybrano klasę do testowania

- Napisano komplet testów jednostkowych

- Sprawdzono pokrycie kodu testami

{miejsce na zdjęcie pokrycia testów}

**Dokumentacja**

**1. Wygenerowano szkielet dokumentacji:**

- Użyto Source → Generate Element Comment

- Uzupełniono dokumentację JavaDoc

- Dodano opisy klas i metod

{miejsce na zdjęcie wygenerowanej dokumentacji}

**2. Przegląd jakości kodu:**

- Dodano komentarze TODO w miejscach wymagających refaktoryzacji

- Zidentyfikowano potencjalne problemy

{miejsce na zdjęcie komentarzy TODO}

**Integracja zmian**

1. Scalono gałąź docs z master:

git checkout master

git merge docs

git push origin master

**Zgłoszenie rozwiązania**

1. Utworzono gałąź review:

- Wybrano odpowiedni commit bazowy

- Utworzono nową gałąź bez własnych commitów

{miejsce na zdjęcie utworzenia gałęzi}

1. Utworzono Pull Request:

- Z gałęzi master do review

- Dodano prowadzącego jako reviewera

{miejsce na zdjęcie Pull Request}