

# Nuclear Reactor Simulation

Dokumentacja projektu symulacji agentowej

Autorzy:

Kamil Kula (Lider)

Jakub Grzeszek

Kurs: Programowanie Obiektowe

Kierunek: Informatyka Techniczna

18 czerwca 2025

# 1 Wprowadzenie

## 1.1 Cel projektu

Projekt **Nuclear Reactor Simulation** realizuje kompleksową symulację interakcji między reaktorami jądrowymi a otaczającymi je miastami. System modeluje:

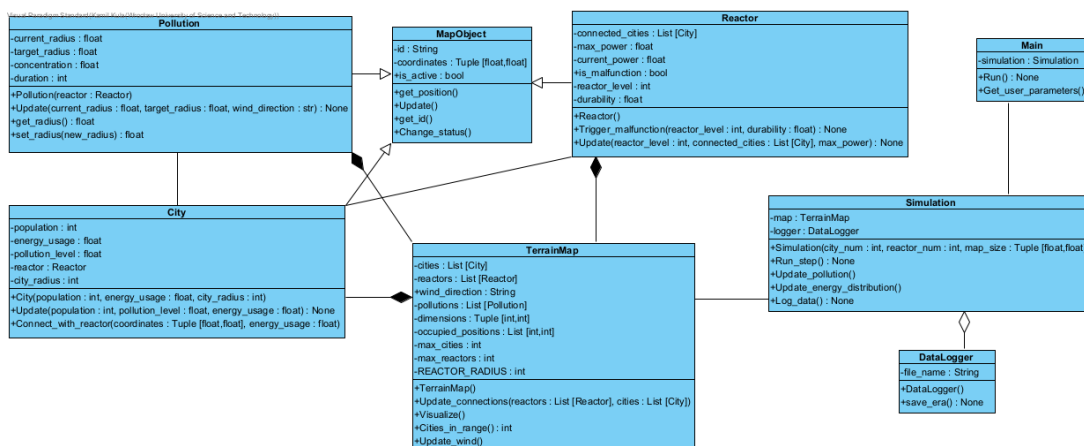
- Dynamiczne zmiany populacji miast w zależności od skażenia
- Mechanizmy awarii reaktorów i rozprzestrzeniania się promieniowania
- System zarządzania energią między reaktorami a miastami
- Wizualizację przestrzenną całego systemu

## 1.2 Technologie

- Język: Java 23
- Budowa: Maven z systemem testów
- Wizualizacja: Java Swing z dynamicznym renderingiem
- Dokumentacja: Javadoc + LaTeX
- Testy: JUnit 5
- Diagramy: PlantText UML Editor (Początkowo visual paradigm)

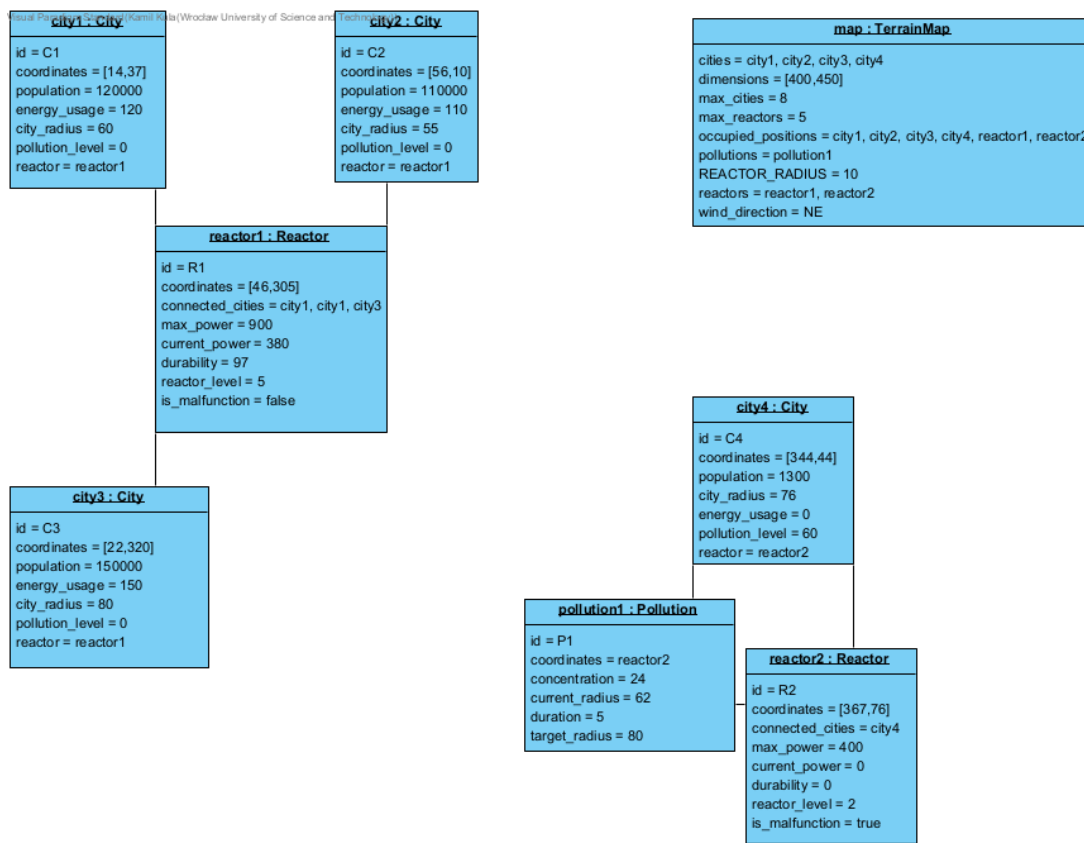
# 2 Etap 1 - Projektowanie systemu

## 2.1 Diagram klas



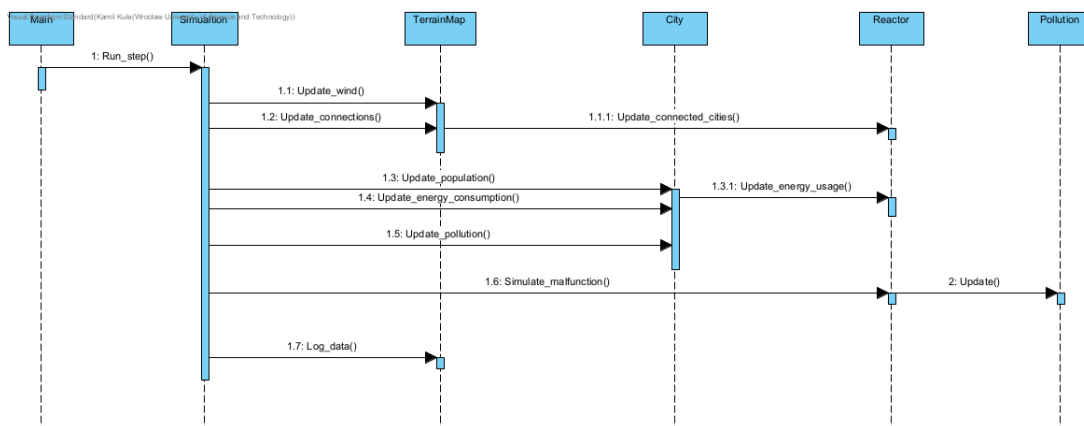
Rysunek 1: Pierwotny diagram klas

## 2.2 Diagram obiektów



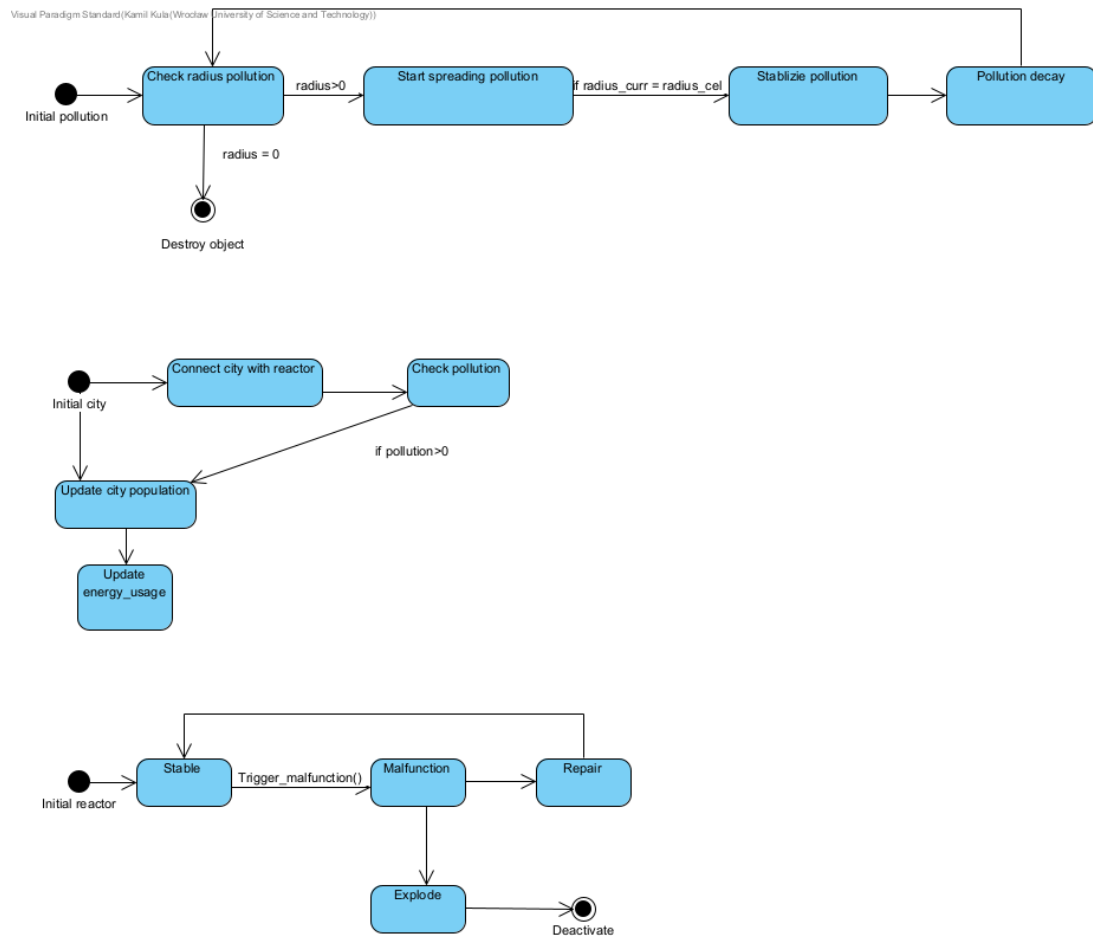
Rysunek 2: Pierwotny diagram obiektów

### 2.3 Diagram sekwencji



### Rysunek 3: Pierwotny diagram sekwencji

## 2.4 Diagram maszyny stanów

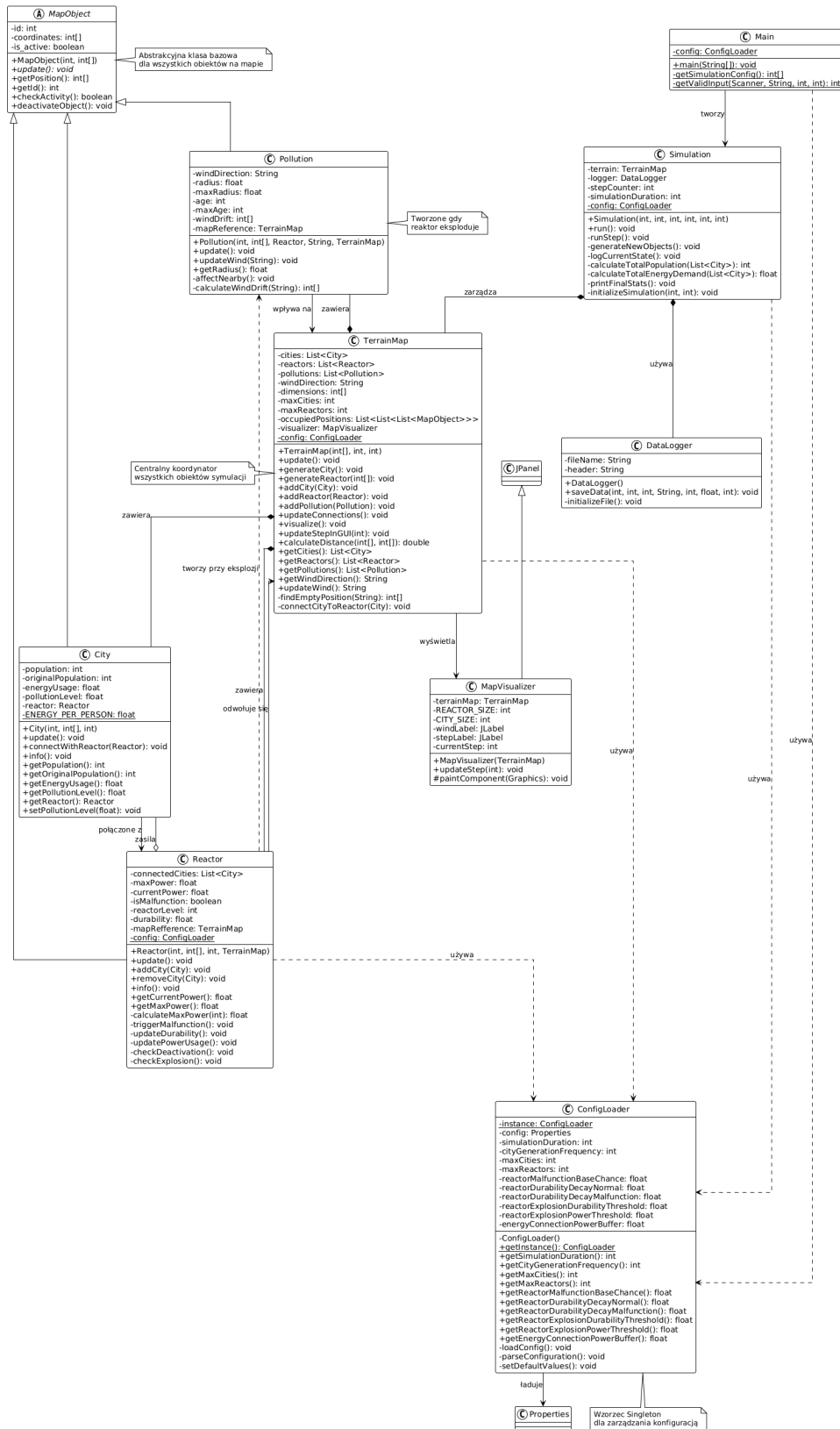


Rysunek 4: Pierwotny diagram maszyny stanów

## 2.5 Diagramy po poprawkach

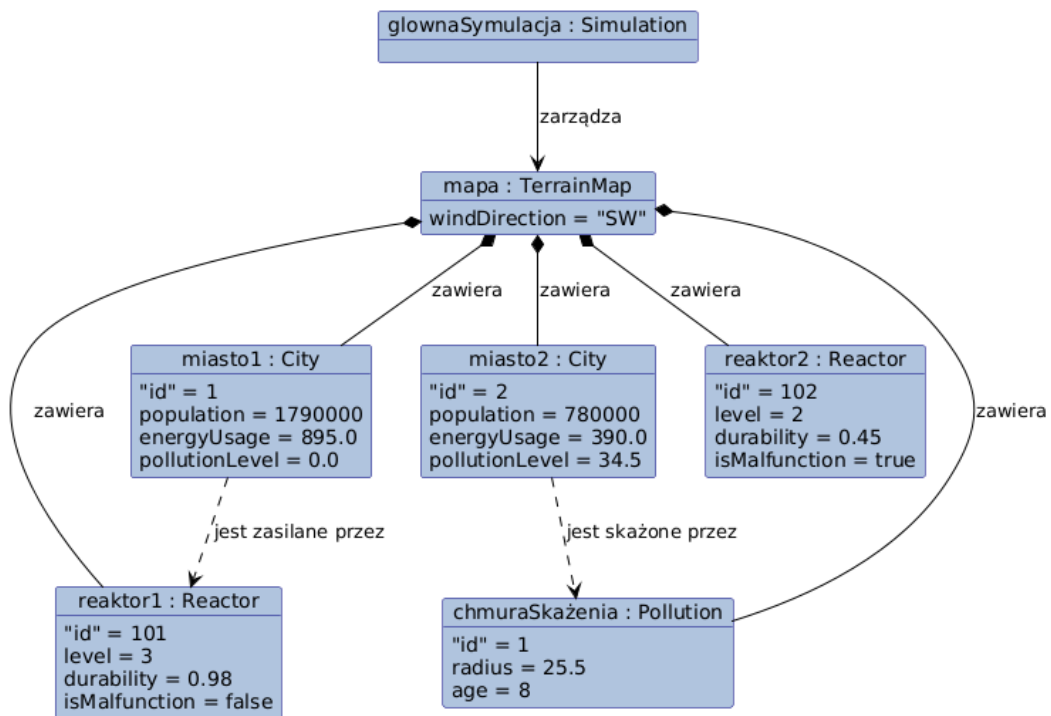
Na podstawie analizy diagramów zostały wprowadzone następujące poprawki:

- W diagramie *terrain map* oraz w diagramie sekwencji poprawiono kierunki i rozmieszczenie strzałek.
- Diagramy zostały podzielone na mniejsze części w celu zwiększenia czytelności i lepszego zobrazowania poszczególnych zależności.
- Publiczne pola w klasach zostały zmienione na prywatne, zgodnie z zasadami programowania obiektowego.



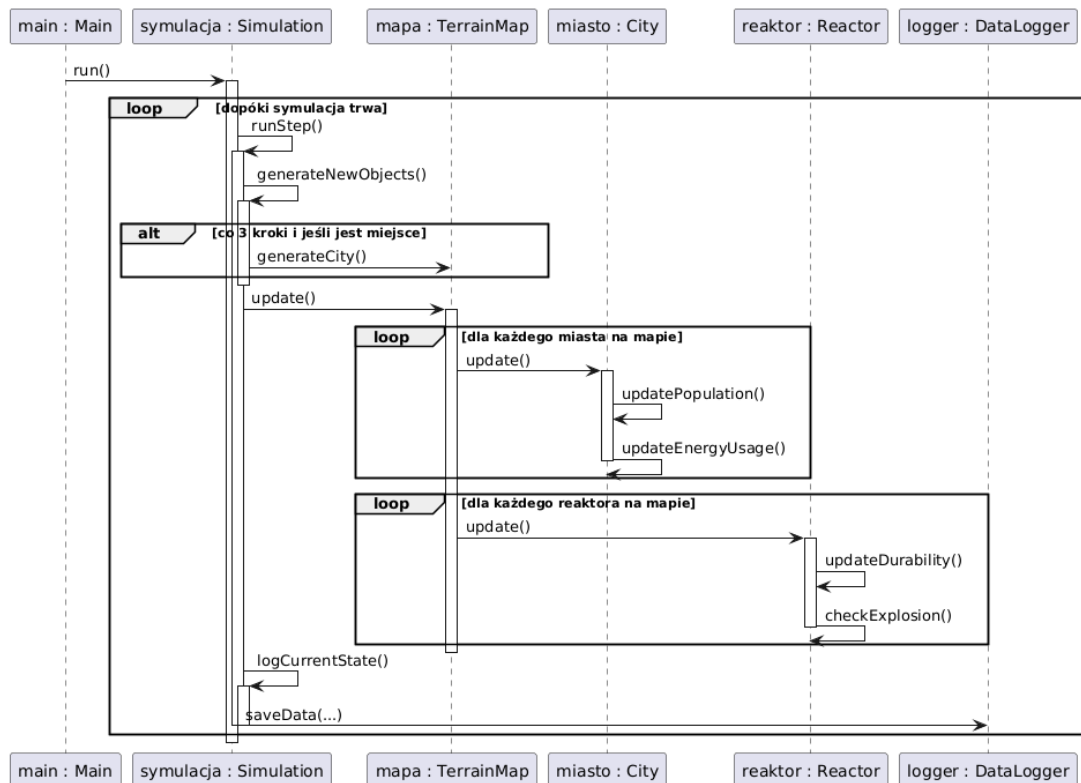
Rysunek 5: Ostateczny diagram klas po wprowadzonych zmianach

Diagram Obiektów: Ostateczna, Poprawiona Wersja



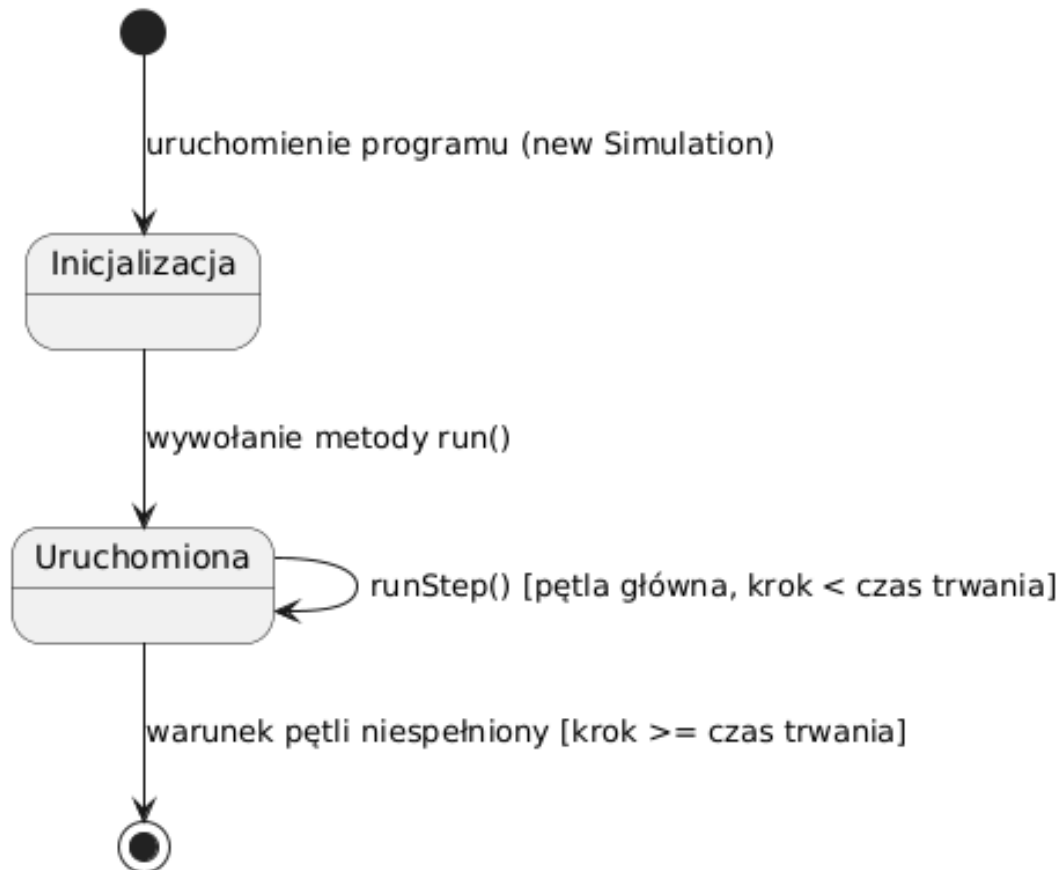
Rysunek 6: Ostateczny diagram obiektów

Diagram Sekwencji: Uruchomienie i krok symulacji



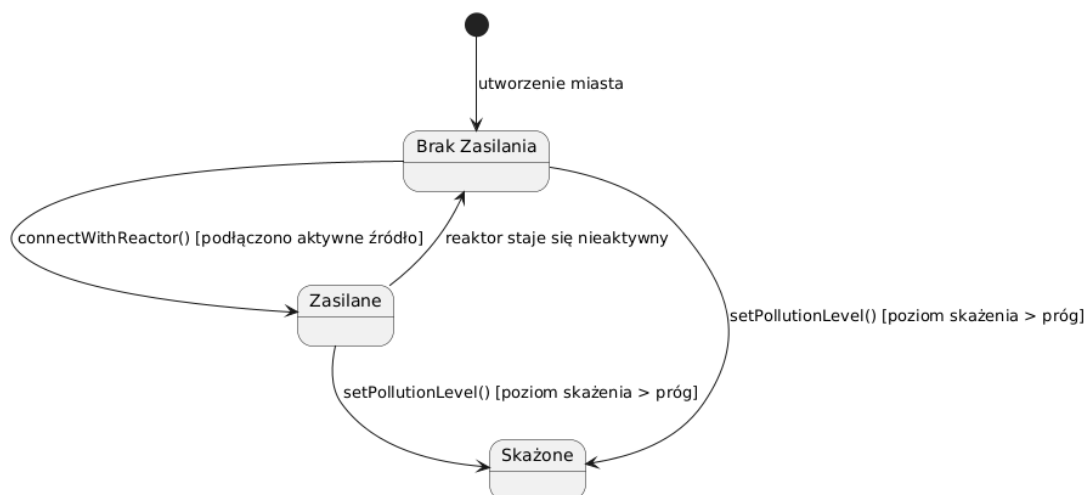
Rysunek 7: Ostateczny diagram sekwencji

## Diagram Stanów dla Aplikacji: Simulation

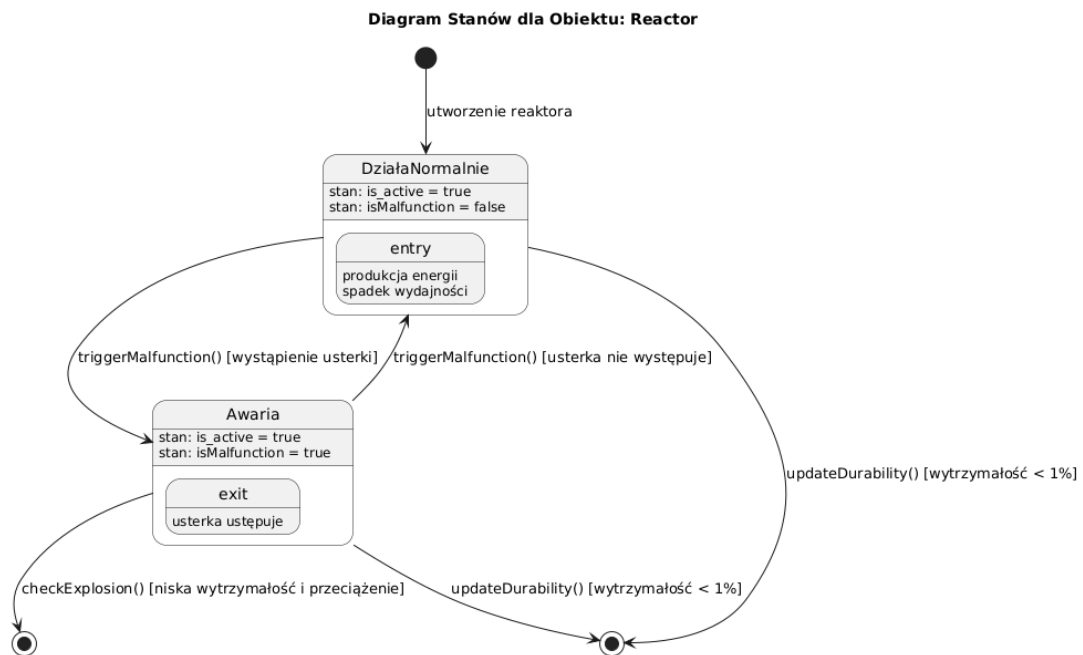


Rysunek 8: Ostateczny diagram maszyny stanów dla Simulation

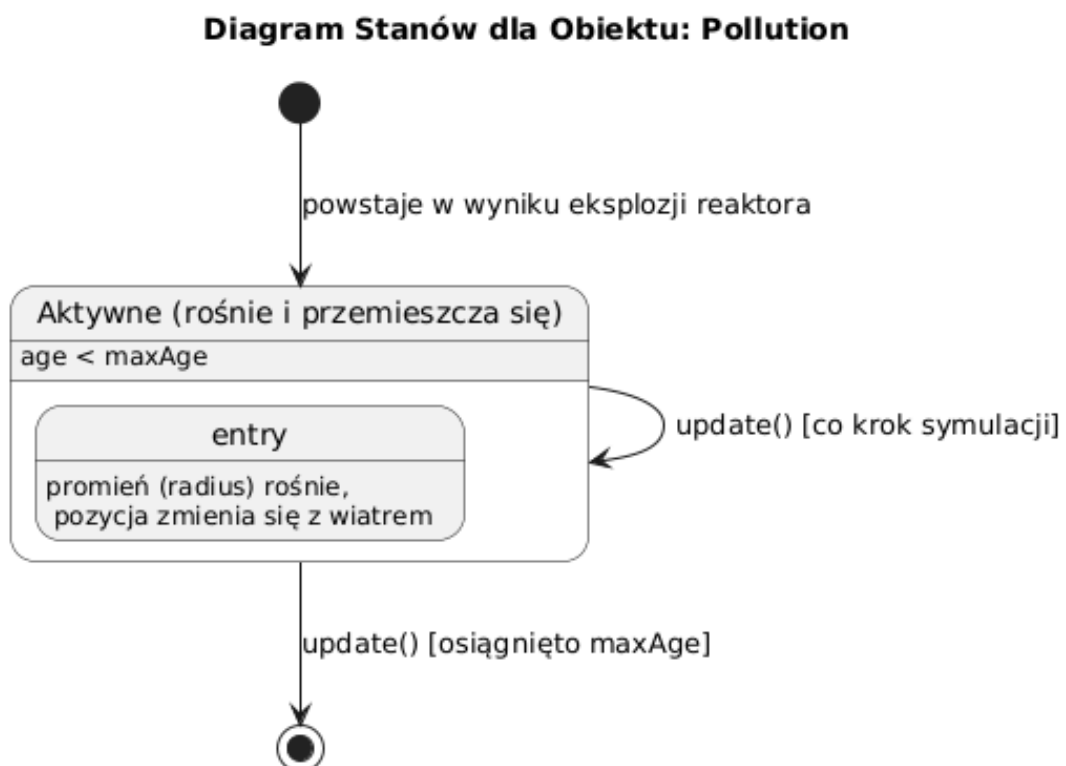
## Diagram Stanów dla Obiektu: City



Rysunek 9: Ostateczny diagram maszyny stanów dla City



Rysunek 10: Ostateczny diagram maszyny stanów dla Reactor



Rysunek 11: Ostateczny diagram maszyny stanów dla Pollution



## **3 Etap 2 - Implementacja podstawowa**

### **3.1 Struktura systemu**

Zaimplementowaliśmy podstawową wersję systemu zgodnie z zaprojektowaną architekturą. Kluczowe komponenty obejmowały:

#### **3.1.1 Moduł symulacji**

- Implementacja logiki głównej pętli symulacji
- Mechanizm aktualizacji stanu wszystkich agentów
- System generowania nowych obiektów

#### **3.1.2 Moduł wizualizacji**

- Interaktywna mapa z kolorowym kodowaniem stanów
- Panel informacyjny z kluczowymi informacjami
- Animacja rozprzestrzeniania się skażenia

#### **3.1.3 Moduł konfiguracyjny**

- Ładowanie parametrów z plików wejściowych
- Walidacja konfiguracji Singleton zapewniający globalny dostęp

## **4 Wnioski i podsumowanie**

### **4.1 Napotkane wyzwania**

- Problem: Kolizje obiektów na mapie
- Rozwiązanie: Implementacja przestrzeni 3D (x,y,z)
- Problem: Niespójności wizualne
- Rozwiązanie: Synchronizacja wątków z `SwingUtilities.invokeLater`

### **4.2 Osiągnięcia**

- Pełna realizacja założeń projektowych
- Przejrzysta wizualizacja ułatwiająca analizę

### **4.3 Dalszy rozwój**

- Rozszerzenie o różne typy elektrowni
- Zaawansowane modele rozprzestrzeniania skażenia