Specyfikacja implementacyjna i funkcjonalna programu rozwiązującego labirynt w Javie

Oliwia Woźniak, Karolina Wiśniewska

1 Opis problemu

Celem naszego programu było rozwiązanie labiryntu wgranego przez użytkownika przy wywołaniu programu. Dozwolony format pliku z labiryntem to tekstowy (.txt) lub binarny (.bin). Przy wywołaniu program program wypisuje do standard output ścieżkę z rozwiązaniem. Przy użyciu odpowiednich flag program również posiada również opcje zapisania rozwiązania do pliku w formacie binarnym. Powinien również posiadać funkcjonalne GUI (Graficzny Interfejs Użytkownika). Program powinien być napisany w języku Java.

2 Algorytm

Do rozwiązania problemu wykorzystałyśmy algorytm Dijkstry. Algorytm ten służy głównie do znajdywania najkrótszej ścieżki w grafie skierowanym, gdzie wszystkie wierzchołki są ze sobą powiązane krawędziami.

W tym wypadku, grafem jest cały labirynt:

- wierzchołkami są wszystkie pola puste, pole początku i końca oraz ściany
- krawędziami są wszystkie przejścia pomiędzy polami, nie będące ścianą. Oznacza to, że ściany są wierzchołkami poza grafem.

Algorytm zaczyna przeszukiwanie labiryntu od punktu określonego jako start (domyślnie lub przez użytkownika),

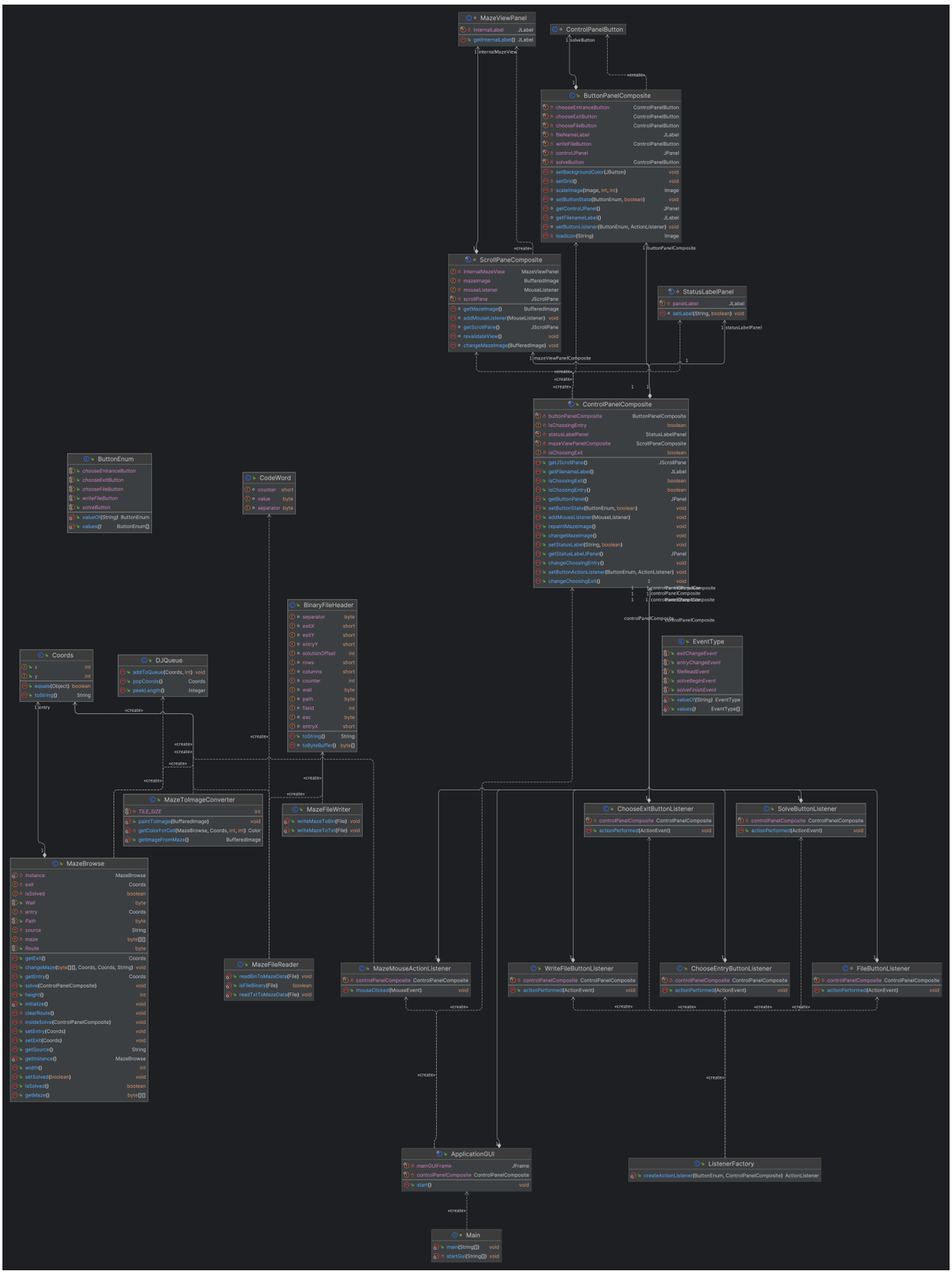
3 Struktura Katalogów

3.0.1 Pakiety - *src/main/java/Main*

- **MazeData** - Pakiet odpowiedzialny za obsługę danych labiryntu, takich jak struktura i właściwości labiryntów.
  - **Coords.java**
  - **DJQueue.java**
  - **MazeBrowse.java**
- **File** - Pakiet odpowiedzialny za operacje na plikach, takie jak wczytywanie i zapisywanie danych.
  - **BinaryFileHeader.java**
  - **CodeWord.java**
  - **MazeFileReader.java**
  - **MAzeFileWriter.java**
- **GUI** - Pakiet zawierający komponenty graficznego interfejsu użytkownika.
  - **ButtonEnum.java** - definiuje typy przycisków używanych w interfejsie graficznym
  - **ButtonPanelComposite.java** - kompozyt zarządzający panelami przycisków "Wybierz labirynt", "Wypisz labirynt", "Wybierz wejście", "Wybierz wyjście", "Rozwiąż"
  - **ControlPanelButton.java** - implementuje przyciski znajdujące się w panelu kontrolnym
  - **ControlPanelComposite.java** - zarządza kompozytem panelu kontrolnego w interfejsie graficznym
  - **MazeToImageConverter.java** - konwertuje strukturę labiryntu na obraz wyświetlany w interfejsie graficznym
  - **MazeView.java** - wyświetla i zarządza widokiem labiryntu
  - **ScrollPaneComposite.java** - kompozyt zarządzający panelami przewijania w interfejsie graficznym.
  - **StatusLabelPanel.java** - mplementuje panel statusowy wyświetlający informacje o stanie pliku z labiryntem
- **Listeners** - Pakiet zawierający klasy nasłuchujące zdarzeń generowanych przez użytkownika. Klasy są odpowiedzialne odpowiednio za przycinki w panelu kontrolnym oraz działania myszki oraz panel przewijania odpowiedzialne za prace w labiryncie
  - **ChooseEntryButton.java**
  - **ChooseExitButton.java**
  - **FileButtonListener.java**
  - **ListenerFactory.java**
  - **MazeMouseActionListener.java**
  - **SolveButtonListener.java**
  - **WriteFileButtonListener.java**

3.0.2 Klasy- *src/main/java/Main*

- **Main.java** - Główna klasa programu uruchamiająca aplikację.
- **ApplicationGUI.java** - Klasa zarządzająca głównym interfejsem graficznym użytkownika.
- **CustomEventManager.java** - Klasa zarządzająca niestandardowymi zdarzeniami w aplikacji.
- **EventType.java** - Klasa definiująca różne typy zdarzeń używane w aplikacji.



Rysunek 1: Diagram Modułów

4 Funkcjonalności Programu

4.1 Zarządzanie Labiryntami

Program umożliwia wgrywanie, edytowanie lub wprowadzanie wejść/ wyjść.

4.2 Wizualizacja Labiryntu

Za pomocą graficznego interfejsu użytkownika (GUI) program wizualizuje strukturę labiryntu, pozwalając użytkownikowi na interaktywną nawigację i manipulację wejść/wyjść. Wizualizuje ścieżke z rozwiązaniem.