Labirynt - Dokumentacja Funkcjonalna

Miłosz Wasiniak, Bartłomiej Ciupak

May 30, 2024

1 Wprowadzenie

Dokument ten opisuje funkcjonalności programu służacego do rozwiazywania labiryntów. Program składa sie z kilku klas, które wspólnie realizuja zadanie wczytywania, rozwiazywania oraz zapisywania labiryntów. Główne klasy to `Maze`, `Pair`, `Notification`, `MazeGUI` oraz klasa `Export`.

2 Klasa Pair

Klasa `Pair` jest prosta struktura danych przechowujaca dwie wartości całkowite, reprezentujace współrzedne w labiryncie.

```
public class Pair {
    private int first;
    private int second;

    public Pair(int first, int second) {
        this.first = first;
        this.second = second;
    }

    public int first() {
        return first;
    }

    public int second() {
        return second;
    }
}
```

3 Klasa Notification

Klasa `Notification` umożliwia dodawanie tekstu do panelu powiadomień w interfejsie użytkownika. Tekst może być wyświetlany w kolorze zielonym (pozytywny) lub czerwonym (negatywny).

```
public class Notification {
    public static void addText(String text, boolean positive, JTextPane eventsTextPane) {
        Color kolor = new Color(0, 128, 0);
        if (!positive) kolor = Color.RED;
        StyledDocument doc = eventsTextPane.getStyledDocument();
        SimpleAttributeSet attributes = new SimpleAttributeSet();
        StyleConstants.setForeground(attributes, kolor);
        StyleConstants.setFontSize(attributes, 14);
        try {
            doc.insertString(doc.getLength(), text + '\n', attributes);
        } catch (Exception e) {
            e.printStackTrace();
```

```
        }
    }
}
```

4 Klasa Maze

Klasa `Maze` odpowiada za wczytywanie, przechowywanie oraz rozwiazywanie labiryntu. Nowo dodane funkcje dotycza wczytywania pliku binarnego.

4.1 Wczytywanie pliku tekstowego

```
public void wczytaj(File plik) {
    try {
        // Implementacja wczytywania pliku tekstowego
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }
}
```

4.2 Wczytywanie pliku binarnego

```
public void wczytajBinarny(File plik) {
    try {
        // Implementacja wczytywania pliku binarnego
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

5 Klasa Export

Klasa `Export` odpowiada za zapisywanie rozwiazania labiryntu w formie tekstowej lub graficznej.

5.1 Zapis rozwiazania w formie tekstowej

```
public class Export {
    public static void zapiszTekstowo(Maze lab, File plik) {
        try {
            // Implementacja zapisywania labiryntu w formie tekstowej
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

5.2 Zapis rozwiazania w formie graficznej

```
public static void zapiszGraficznie(Maze lab, File plik) {
    try {
        // Implementacja zapisywania labiryntu w formie graficznej
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

6 Klasa MazeGUI

Klasa `MazeGUI` zapewnia interfejs graficzny użytkownika do wczytywania, rozwiazywania i zapisywania labiryntów.

6.1 Wczytywanie labiryntu

```
loadMenuItem.addActionListener(e -> {
    JFileChooser fileChooser = new JFileChooser();
    FileNameExtensionFilter filter = new FileNameExtensionFilter(
        "Pliki tekstowe i binarne", "txt", "bin");
    fileChooser.setFileFilter(filter);
    int returnValue = fileChooser.showOpenDialog(null);
    if (returnValue == JFileChooser.APPROVE_OPTION) {
        File selectedFile = fileChooser.getSelectedFile();
        if (selectedFile.getName().endsWith(".bin")) {
            lab.wczytajBinarny(selectedFile);
        } else {
            lab.wczytaj(selectedFile);
        }
        wypelnij();
        Notification.addText("-Wczytano plik " + selectedFile.getName() + "!", true, eventsTextPane)
    }
});
```

6.2 Zapis labiryntu

Dostepne sa opcje zapisu labiryntu w formie graficznej oraz tekstowej.