

Specyfikacja projektowa JavaDEKOM

Bartłomiej Janota

Michał Bibrzycki

1 Cel projektu

JavaDEKOM to program, który został stworzony w celu dekompresji plików, które wcześniej zostały skompresowane przy użyciu MiniByte. Program umożliwia również deszyfrowanie tych plików za pomocą podanego hasła. Interfejs użytkownika programu jest w pełni graficzny i umożliwia pobieranie wszystkich niezbędnych danych podczas jego działania.

2 Opis funkcjonalności

Dostępne funkcjonalności to:

- **Dekompresja** - Program dekompresuje podany przez użytkownika plik.
- **Deszyfrowanie** - Program deszyfruje podany przez użytkownika plik za pomocą hasła, które użytkownik wpisał w odpowiednie pole.
- **Wyświetlanie drzewa** - Program wyświetla drzewo Huffmana, za pomocą którego zaszyfrowano plik.

3 Opis modułów

Pełny diagram klas znajduje się w **załączniku 1**.

- **FileHandler** - jest odpowiedzialny za obsługę plików w programie. Zapewnia funkcje wyboru plików wejściowych i wyjściowych poprzez interfejs graficzny.
- **Main** - jest odpowiedzialny za uruchamianie interfejsu graficznego programu. Inicjalizuje główne okno aplikacji, wczytuje pliki FXML, ustawia ikonę, styl i tytuł okna. Udostępnia także metodę "getStage" do uzyskania referencji do głównego okna aplikacji.
- **MainViewController** - jest odpowiedzialny za obsługę interakcji użytkownika z interfejsem graficznym programu. Zawiera pola i metody, które odpowiadają różnym elementom interfejsu, takim jak przyciski, etykiety, pola tekstowe itp.
- **MsgChanger** - zawiera metody do zmiany komunikatów wyświetlanych w interfejsie użytkownika. Umożliwia zmianę tekstu w etykiecie wiadomości.
- **ReadWriteFile** - odpowiada za operacje odczytu i zapisu plików w programie. Wykorzystuje bibliotekę Java NIO do obsługi operacji na plikach.
- **BinaryOperations** - zawiera metody do operacji na danych binarnych w programie.

- **CheckFile** - zawiera metody do sprawdzania poprawności pliku.
- **ConnectIndex** - zawiera metody sprawdzające, które gałęzie należy połączyć.
- **Decode** - realizuje proces dekodowania danych, przemieszczając się po drzewie symboli i odkodowując odpowiednie bity danych.
- **DeepCheck** - zawiera metodę do obliczania maksymalnej głębokości drzewa Huffmana.
- **Dekom** - zawiera metodę "solve", która wykonuje proces dekompresji.
- **HuffmanTreeDisplay** - zawiera metodę "displayHuffmanTree", która jest odpowiedzialna za wyświetlanie drzewa Huffmana w interfejsie użytkownika.
- **Leaves** - jest klasą abstrakcyjną, która definiuje interfejs dla węzłów drzewa Huffmana. Zawiera deklaracje abstrakcyjnych metod, które muszą być zaimplementowane przez klasy dziedziczące po "Leaves".
- **MouseControl** - zawiera logikę obsługi myszy w kontekście skalowania elementów (zoom in / zoom out).
- **PswdTry** - służy do odszyfrowania pliku hasłem
- **ResetValues** - zawiera metodę "reset()", która służy do resetowania wartości zmiennych globalnych używanych w klasie "Dekom".
- **Symbols** - jest wykorzystywana w procesie dekompresji w klasie "Dekom" do przechowywania i manipulowania symbolami. Pozwala na zmianę LinkedList na bitową tablicę (getAllSymbols) oraz umożliwia poruszanie się po drzewie Huffmana.
- **TreeNode** - jest implementacją abstrakcyjnej klasy "Leaves" i reprezentuje węzeł drzewa kodowania Huffman'a. Umożliwia łączenie ze sobą gałęzi.
- **Mode0** - służy do sprawdzania, który mode należy ustawić.
- **Mode1** - służy do łączenia ze sobą liści.
- **Mode2** - służy do tworzenia nowego liścia.
- **Mode3** - uruchamiany, gdy wiemy, że następny znak jest ostatni. Łączy ze sobą wszystkie poprzednie znaki, które nie zostały połączone.
- **ModeAll** - służy do obsługi mode0-3.
- **ModeInterface** - jest interfejsem w pakiecie prog.javadekom.cal.Modes. Definiuje on jedną metodę execute() bez żadnej implementacji. Ten interfejs służy jako kontrakt dla klas, które go implementują, określając, że muszą dostarczyć implementację metody execute(). Interfejs ten zapewnia strukturę dla różnych trybów działania w aplikacji, które mogą być wywoływane w zależności od potrzeb.

4 Dane wejściowe

Danymi wejściowymi są:

- Skompresowany przez MiniByte plik,
- Ścieżka do pliku wyjściowego,
- Hasło.

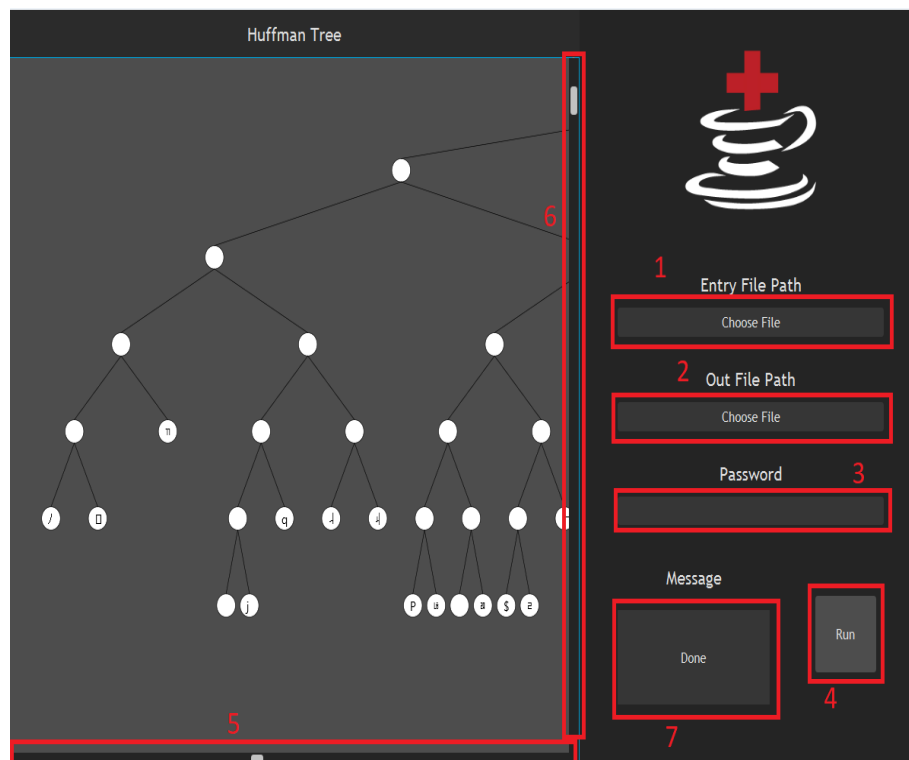
5 Dane wyjściowe

Danymi wyjściowymi są:

- Zdekompresowany plik wejściowy zapisany na ścieżce pliku wyjściowego podanej przez użytkownika.

6 Instrukcja obsługi programu

1. Aby wybrać plik do dekompresji, należy kliknąć przycisk "Choose File" pod napisem Entry File Path (oznaczony numerem 1 na rysunku). Pojawi się okienko, w którym należy wybrać plik wejściowy.
2. Aby wybrać miejsce pliku wyjściowego, należy kliknąć przycisk "Choose File" pod napisem Out File Path (oznaczony numerem 2 na rysunku). Pojawi się okienko, w którym należy wybrać plik wyjściowy.
3. Jeżeli plik został zaszyfrowany hasłem, należy je wpisać w okienko znajdujące się pod napisem "Password (oznaczone numerem 3 na rysunku)". Jeżeli plik nie był zaszyfrowany, należy pozostawić to okienko puste.
4. Aby zdekompresować plik, należy kliknąć przycisk "Run" (oznaczony numerem 4 na rysunku)
5. Po dekompresji pojawi się drzewo Huffmana. Aby się po nim poruszać, należy kliknąć na drzewo i przeciągnąć. Aby powiększyć lub pomniejszyć drzewo należy użyć scrolla. Alternatywnym sposobem na poruszanie się po drzewie są suwaki umieszczone na dole (oznaczony numerem 5 na rysunku) oraz po prawej stronie drzewa (oznaczony numerem 6 na rysunku).
6. Wszystkie komunikaty będą pojawiały się w okienku pod napisem "Message" (oznaczone numerem 7 na rysunku).



7 Scenariusz działania programu

1. Użytkownik wybiera plik do dekompresji.
2. Użytkownik wybiera plik wyjściowy.
3. Jeśli jest taka potrzeba, to użytkownik podaje hasło do odszyfrowania pliku.
4. Użytkownik uruchamia działanie programu klikając przycisk "Run".
5. Program sprawdza poprawność danych wejściowych.
6. Jeśli plik jest zaszyfrowany i użytkownik podał hasło, program deszyfruje plik.
7. Program odczytuje słownik.
8. Program dekompresuje plik.
9. Program powiadamia użytkownika o zakończeniu dekompresji.
10. Program wyświetla drzewo Huffmana.
11. Na żądanie zamknięcia programu - program kończy działanie.

8 Komunikaty błędów i sytuacje wyjątkowe

Tabela komunikatów:

Komunikat błędu	Wyjaśnienie
Cannot open: - Entry File	Program nie może otworzyć pliku wejściowego
Cannot open: - Out File	Program nie może otworzyć pliku wyjściowego
File or password is not correct	Plik lub hasło są niepoprawne
You have to select valid: -Entry File Path -Out File Path	Nie wybrano pliku wejściowego lub pliku wyjściowego

9 Załączniki

Załącznik 1 - diagramKlas.png