|  |  |  |
| --- | --- | --- |
| Projekt **Java**  Wydział Elektrotechniki Automatyki i Informatyki  Politechnika Świętokrzyska | | |
| Zespół  Piotr Skiba  Krzysztof Wódkowski | Program do kompresji (dekompresji) plików i katalogów (format ZIP) | Grupa: 2ID12B  2016\2017  08.09.2017 |

**1.Opis ogólny**

Program polega na kompresji i dekompresji plików i katalogów (format Zip). Wykorzystywana jest biblioteka java.util.zip, zawierając najważniejsze klasy potrzebne do kompresji i dekompresji zip.

W projekcie zostały użyte takie biblioteki jak :

**- java.util.zip**

-java.awt.event.ActionEvent;

**-** java.awt.event.ActionListener;

**-**java.io.IOException;

**-**javax.swing.JFileChooser;

**-**javax.swing.JFrame;

**-**javax.swing.JMenu;

**-**javax.swing.JMenuBar;

**-**javax.swing.JMenuItem;

**-**javax.swing.JOptionPane;

- java.awt.Font;

Projekt był stworzony w programie w Eclipse Version: Neon.2 w ‘JavaSE - 1.8’ i składa się on z 6 klas. A są to między innymi :

**Main** - Klasa główna

**ZipArchiver -**  Interfejs (kompresja)

**Unarchiver**– Interfejs (dekompresja)

**SimpleZipArchiver –** Kompresja (implementacja ZipArchiver)

**UnZip** – Interfejs (implementacja Unarchiver)

**ZipApp** - Klasa implementująca interfejsy ZipArchiver i Unarchiver, pozwalająca wywoływać metody archive/unarchive z poziomu interfejsu graficznego

**2.Informacje na temat funkcjonalności projektu**

Funkcje programu :

- kompresja plików/katalogów

- dekompresja plików i katalogów

- dodanie komentarza do kompresji

- możliwość wyboru ścieżki kompresji i dekompresji

- prosta i szybka obsługa

**Klasa Main:**

Konstruktor klasy **Main** tworzy nowe okno wraz z interfejsem graficznym oraz ustawienie ActionListenera dla poszczególnych opcji w menu głównym programu ,

**Klasa ZipArchiver:**

Interfejs zawierający metody potrzebne przy kompresji

**Unarchiver:**

Interfejs zawierający metody potrzebne przy dekompresji.

**SimpleZipArchiver:**

Klasa **SimpleZipArchiver**główna klasa odpowiedzialna za kompresje

plików i katalogów (format Zip), wyposażona w funkcjonalność dodania komentarza do kompresowanego pliku i katalogu.

**UnZip :**

Klasa **UnZip** główna klasa odpowiedzialna za dekompresje plików i katalogów (format Zip)

**Klasa ZipApp:**

W Klasie **ZipApp** implementowane są 3 metody odpowiedzialne za prawidłowe funkcjonowanie programu (archive, unarchive, setComment).

**3.Informacje na temat sposobu uruchomienia oraz obsługi projektu**

Projekt można uruchomić poprzez uruchomienie ProjektZip.jar

Obsługa działania programu:

1. Program umożliwia kompresje jak i dekompresji plików lub katalogów.

W celu skorzystania z pierwszej opcji należy wejść w menu główne programu, następnie kliknąć opcje „ZIP” i wybrać „Wybierz plik do kompresji” w celu wybrania odpowiedniego pliku lub katalogu do kompresji.

2. Następnie wyskoczy okno w którym użytkownik musi wybrać ścieżkę do pliku który

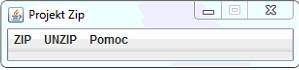
będzie obsługiwany przez program. Kolejno program spyta o możliwość dodania komentarza. Jeśli użytkownik wybierze ‘tak’ , wyskoczy okno w którym będzie możliwość wpisana własnego komentarza. Po czym należy wybrać miejsce docelowe kompresji.

Jeśli działanie ukończy się ona pomyślnie wyskoczy okno dialogowe potwierdzające ukończenie kompresji.

3. Program ma również możliwość dekompresji wybranego przez użytkownika pliku lub katalogu .W tym celu należy w menu głównym programu kliknąć opcje „UNZIP” a następnie wybrać „Wybierz plik do dekompresji” w celu wybrania pliku lub katalogu do dekompresji.

4. Po wybraniu przez użytkownika tej opcji wyskoczy okno w którym użytkownik może wybrać ścieżkę do pliku który będzie dekompresowany. Następnie wyskoczy okno w którym będzie mógł wybrać ścieżkę docelową. Użytkownik ma możliwość utworzyć i nazwać folder w którym znajdą się zdekompresowane pliki. Jeśli dekompresja ukończy się pomyślnie, wyświetli się komunikat o ukończeniu operacji.

5. Użytkownik ma możliwość w każdej chwile skorzystać z Pomocy jak i może zakończyć działanie programu. W celu skorzystania z pomocy należy kliknąć na „Pomoc” następnie „ Wyświetl Pomoc” . A w celu zakończenia działania kliknąć „ZIP” lub „UNZIP” a następnie „Wyjście”.



**4. Informacje na temat stworzonych klas, metod, funkcji**

**ZipArchiver:**

Interfejs**ZipArchiver**zawierametody**archive(String source, String target)**oraz**setComment(String msg)**.

**voidarchive(String source, String target)** – metoda przyjmuje jako parametry ścieżkę do pliku, który ma zostać skompresowany oraz ścieżkę gdzie ma być umieszczony już spakowany.

**voidsetComment(String msg)** – ustawia do pliku kompresowanego komentarz

Interfejs zawiera również **enum COMPRESSION\_LEVEL**, który oznacza poziom kompresji. Domyślnie w programie jest on ustawiony na 5.

**Unarchiver:**

Interfejs**Unarchiver** zawiera metodę**unarchive(String source, String target)**. Przyjmuje ona takie same parametry jak**archive**, ale ma odwrotne działanie.

**SimpleZipArchiver:**

Klasa**SimpleZipArchiver**implementujeinterfejs**ZipArchiver**. Posiada 2 konstruktory. **Domyślny** ustawia poziom kompresji na 5, natomiast za pomocą **drugiego** można ustalić dowolny poziom kompresji.

Klasa zawiera implementację metod interfejsu.

**voidarchive(String source, String target)** – pobiera w parametrach ścieżkę pliku do skompresowania i ścieżkę, gdzie ma zostać umieszczony plik skompresowany. W miejscu „**target**” tworzony jest nowy plik „**target.createNewFile()**” Tworzone są strumienie wyjściowe

**FileOutputStream** (dla target) oraz**ZipOutputStream**. Następnie ustawiany jest poziom kompresji **zipOutputStream.setLever(compressonLevel.lvl)**. Następnie, jeśli ustawiliśmy komentarz, to jest on dodawany do pliku zip, po czym wywołana zostaje metoda **archiveDir** lub **archiveFile** w zależności czy wskazaliśmy katalog czy plik.

Po zakończonej kompresji strumienie zostają zamknięte.

**archiveDir(File file)** – dla każdego pliku znajdującego się w katalogu wywołuje metodę **archiveFile** lub sama siebie (**rekurencja**), jeśli w katalogu są podkatalogi.

**archiveFile(File file)** – kompresuje plik za pomocą **zipOutputStream**

**UnZip:**

Klasa implementująca interfejs **Unarchiver.** Implementuje metodę **unarchive.**

**voidunarchive(String source, String target)** – tworzy folder w miejscu target, następnie pobiera wejście **zipInputStream**. Dla każdego wejścia (jeśli istnieje) dekompresuje plik, tworząc nowy, a następnie za pomocą **fileOutputStream**zapisuje go we wskazanym folderze.

**ZipApp:**

Klasa implementująca interfejsy **ZipArchiver** i **Unarchiver**. Za jej pomocą wywoływane są poszczególne metody tych interfejsów. Jest ona wykorzystywana w interfejsie graficznym.

**Main:**

Klasa główna, zawierająca interfejs graficzny. Za pomocą **JfileChooser** pozawala wybierać pliki lub foldery do skompresowania i dekompresowania. Korzysta ona z obiektu klasy **ZipApp** dla ułatwionego korzystania z metod interfejsów.