

31.01.2024 r.

Politechnika Śląska w Gliwicach  
Wydział Automatyki, Elektroniki i Informatyki



**Politechnika  
Śląska**

## **Programowanie Komputerów II**

Sprawozdanie z projektu

Autor	Michał Dyrka
Kierunek	Teleinformatyka
Semestr	3
Grupa	1

## Spis treści

<b>Wprowadzenie i opis aplikacji.....</b>	<b>3</b>
Opis generalny .....	3
Problemy inżynierskie .....	3
Cele pracy.....	4
<b>Użytkowanie i struktura .....</b>	<b>4</b>
Struktura .....	4
Użycie.....	5
Lista klas .....	6
Testy .....	7
Wnioski .....	7

# Wprowadzenie i opis generalny

## Opis generalny

Aplikacja z kompresją Huffmana to proste narzędzie do optymalizacji danych, które wykorzystuje algorytm kompresji Huffmana do efektywnego redukowania rozmiaru plików bez utraty danych. Przez analizę częstotliwości występowania poszczególnych bitów lub znaków w danych, program tworzy drzewo Huffmana, pozwalające na przypisanie krótszych kodów do najczęściej występujących elementów i dłuższych do tych rzadszych. Dzięki temu mechanizmowi, aplikacja zapewnia redukcję rozmiaru plików, co jest niezwykle korzystne w przechowywaniu i przesyłaniu danych, szczególnie w środowiskach z ograniczoną przepustowością lub przestrzenią dyskową.

## Problemy inżynierskie

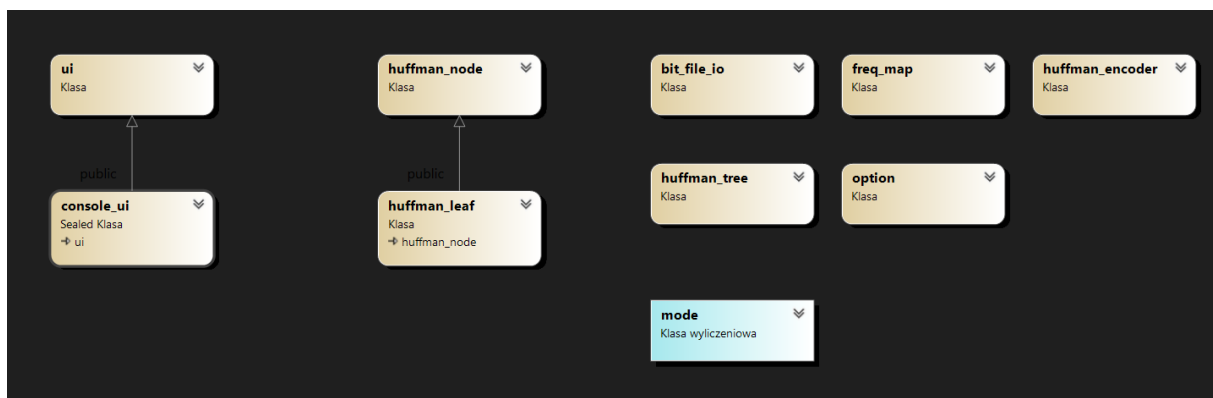
1. **Optymalizacja pamięci i wydajności:** Algorytm kompresji Huffmana wymaga efektywnego zarządzania pamięcią, zwłaszcza przy tworzeniu i przechowywaniu drzewa Huffmana. Optymalizacja struktur danych, aby minimalizować zużycie pamięci i maksymalizować szybkość przetwarzania, jest kluczowa.
2. **Dynamiczne kodowanie i dekodowanie:** Aplikacja musi sprawnie kodować i dekodować dane, adaptując się do różnorodnych zestawów danych. Wymaga to dynamicznego tworzenia drzewa Huffmana na podstawie analizy częstotliwości występowania elementów w danych i skutecznego jego wykorzystania zarówno w procesie kompresji, jak i dekompresji.
3. **Obsługa dużych i zróżnicowanych danych:** Aplikacja powinna być zaprojektowana do pracy z różnymi rodzajami i rozmiarami danych, co może wymagać szczególnych strategii przetwarzania, aby zapewnić efektywność i uniknąć przeciążeń.
4. **Interfejs użytkownika i doświadczenie:** Tworzenie intuicyjnego interfejsu użytkownika, który ułatwia korzystanie z zaawansowanych funkcji kompresji bez konieczności głębokiej wiedzy technicznej, jest wyzwaniem. Ważne jest również zapewnienie, że użytkownik ma jasne informacje o postępie kompresji i opcjach.
5. **Zarządzanie błędami i wyjątkami:** Skuteczne zarządzanie błędami i wyjątkami w procesie kompresji i dekompresji jest niezbędne, aby zapewnić stabilność i niezawodność aplikacji, zwłaszcza przy nieprzewidzianych problemach z danymi wejściowymi.

## Cele pracy

1. **Uniwersalność kompresji i dekompresji:** Stworzenie zaawansowanego mechanizmu kompresji i dekompresji zdolnego do obsługi danych dowolnego formatu i wielkości, co zapewnia szerokie zastosowanie aplikacji w różnorodnych środowiskach i przypadkach użycia.
2. **Intuicyjność interfejsu użytkownika:** Projekt interfejsu użytkownika ma na celu maksymalne uproszczenie procesu kompresji i dekompresji dla użytkowników na każdym poziomie zaawansowania, umożliwiając łatwe i szybkie korzystanie z aplikacji bez konieczności posiadania specjalistycznej wiedzy technicznej.
3. **Zrozumiałe parametry wejściowe:** Zapewnienie, że wszelkie parametry wejściowe, takie jak wybór plików do kompresji/dekompresji czy określenie poziomu kompresji, są jasne i łatwe do zrozumienia przez użytkownika, co minimalizuje ryzyko błędów i poprawia ogólną użyteczność aplikacji.
4. **Optymalizacja wydajności:** Opracowanie algorytmów kompresji i dekompresji w taki sposób, aby zapewnić optymalną wydajność zarówno pod względem szybkości przetwarzania, jak i efektywności redukcji rozmiaru plików, przy jednoczesnym zachowaniu integralności danych.

## Użytkowanie i struktura

### Struktura programu

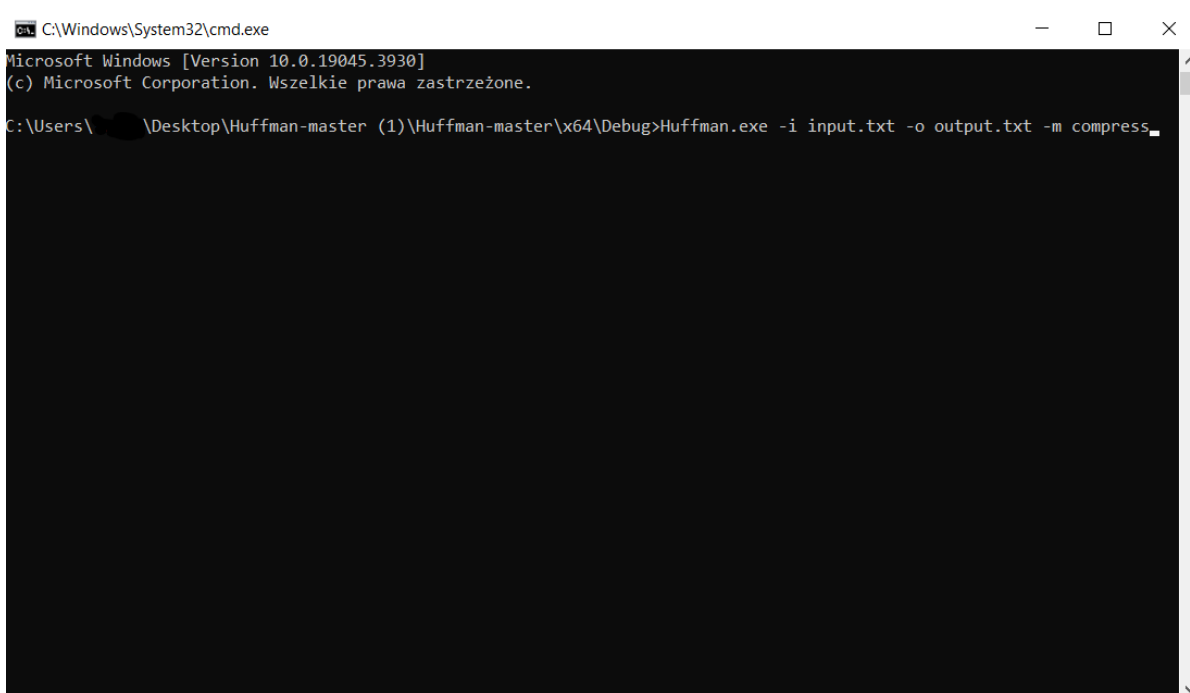


Struktura programu została przedstawiona na zdjęciu powyżej. W programie został zawarty polimorfizm zgodnie z wymaganiami.

## Użycie

Aby skorzystać z programu kompresji Huffmana, użytkownik powinien uruchomić aplikację w wierszu poleceń, dostarczając odpowiednie parametry wejściowe, które sterują działaniem programu. Poniżej znajduje się opis użycia programu na podstawie załączonego zrzutu ekranu:

1. **Wybór pliku wejściowego:** Parametr **-i {inputfile}** służy do określenia pliku, który ma być skompresowany lub zdekompresowany. Na przykład **-i input.txt** wskazuje, że plikiem wejściowym jest **input.txt**.
2. **Wybór pliku wyjściowego:** Parametr **-o {outputfile}** określa nazwę pliku wyjściowego, który zostanie utworzony po kompresji lub dekompresji danych. Na zrzucie ekranu **-o output.txt** określa, że wynikowy plik ma nazwę **output.txt**.
3. **Określenie trybu działania:** Parametr **-m {mode}** pozwala użytkownikowi wybrać tryb działania programu. Dostępne są dwa tryby: **compress** dla kompresji danych i **decompress** dla dekompresji. W przykładzie na obrazku użyto **-m compress**, co oznacza, że dane z **input.txt** zostaną skompresowane i zapisane w pliku **output.txt**.



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.3930]
(c) Microsoft Corporation. Wszelkie prawa zastrzeżone.
C:\Users\... \Desktop\Huffman-master (1)\Huffman-master\x64\Debug>Huffman.exe -i input.txt -o output.txt -m compress
```

### Procedura Użycia:

- Otwórz wiersz poleceń w systemie Windows.
- Przejdź do katalogu, w którym znajduje się plik wykonywalny programu Huffman.
- Wprowadź polecenie zawierające odpowiednie parametry. Na przykład:

Huffman.exe -i input.txt -o output.txt -m compress

- Naciśnij **Enter**, aby uruchomić proces.

Obsługa Błędów:

Program jest zaprojektowany tak, aby obsługiwać przypadki błędnego użycia parametrów:

- Jeśli którykolwiek z parametrów zostanie pominięty lub podany niepoprawnie, program wyświetli stosowny komunikat o błędzie i poinformuje o prawidłowym użyciu.
- Niepoprawne ścieżki do plików (np. nieistniejące pliki) wywołą komunikaty o błędach, informując użytkownika o problemie.
- Wybranie niepoprawnego trybu (inny niż **compress** lub **decompress**) również zwróci błąd, podpowiadając dostępne opcje.

Ważne jest, aby zawsze podawać pełne i poprawne ścieżki dostępu do plików oraz wybrać odpowiedni tryb działania programu.

## Lista klas

### 2.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

<a href="#">bit_file_io</a>	Klasa opakowująca std::fstream. Umożliwia pisanie i czytanie, z i do pliku, pojedynczych bitów .	7
<a href="#">console_ui</a>	Prosta implementacja konsolowego interfejsu użytkownika. Pisze na standardowe wyjście i wyjście błędu . . . . .	10
<a href="#">freq_map</a>	Klasa pomocnicza do przechowywania częstotliwości występowania bajtów . . . . .	11
<a href="#">huffman_encoder</a>	Klasa służąca do kompresji/dekompresji plików przy pomocy kodowania Huffmana . . . . .	13
<a href="#">huffman_leaf</a>	Liść drzewa Huffmana. Zawiera częstotliwość oraz bajt, który reprezentuje. Dziedziczy po <a href="#">huffman_node</a> . . . . .	15
<a href="#">huffman_node</a>	Wierzchołek drzewa Huffmana . . . . .	17
<a href="#">huffman_tree</a>	Reprezentacja drzewa Huffmana . . . . .	18
<a href="#">option</a>	Klasa reprezentująca opcje wiersza linii poleceń . . . . .	20
<a href="#">ui</a>	Prosty interfejs służący do komunikacji z użytkownikiem . . . . .	23

## Testowanie

Aplikacja wykorzystująca algorytm kompresji Huffmana została poddana kompleksowym testom manualnym, mającym na celu zapewnienie jej niezawodności i odporności na błędy. Testy te obejmowały różnorodne scenariusze użytkowania, w tym próby kompresji i dekompresji plików o różnych rozmiarach i formatach, a także próby uruchamiania aplikacji z niekompletnymi lub niepoprawnymi parametrami wejściowymi. Dzięki troskliwej walidacji danych wejściowych i starannemu projektowi obsługi wyjątków, aplikacja wykazuje wysoką tolerancję na błędy użytkownika i inne potencjalne problemy operacyjne. Zastosowano mechanizmy zapobiegające awariom wynikającym z typowych błędów, takich jak nieprawidłowe ścieżki do plików, niewłaściwe formaty danych, czy niepoprawne polecenia działania. Wynikiem tych działań jest solidne narzędzie, które zapewnia stabilność i niezawodność nawet w wymagających warunkach i jest przyjazne dla użytkownika, co czyni je wartościowym dodatkiem do narzędzi do zarządzania danymi.

## Wnioski

Projekt aplikacji wykorzystującej algorytm kompresji Huffmana został zrealizowany z pełnym przestrzeganiem założeń określonych w specyfikacji wymagań. Wykorzystane struktury danych, takie jak drzewa binarne, oraz algorytmy kompresji i dekompresji zostały starannie dobrane i zoptymalizowane w celu zapewnienia efektywnej i bezbłędnej operacji na danych o dowolnym rozmiarze i formacie. Implementacja w pełni korzysta z zalet programowania obiektowego, gdzie enkapsulacja, dziedziczenie i polimorfizm przyczyniają się do modularności, czytelności oraz łatwości w utrzymaniu i rozbudowie kodu.

Wprowadzono szczególną staranność w projektowaniu interfejsów użytkownika i obsługi wyjątków, co skutkuje intuicyjnym i odpornym na błędy doświadczeniem użytkownika. Przejrzysta architektura systemu oraz zastosowanie wzorców projektowych, takich jak Fabryka czy Dekorator, dodatkowo wzmacniają skalowalność i elastyczność rozwiązania. Dodatkowo, systematyczne recenzje kodu oraz zastosowanie testów jednostkowych i integracyjnych przyczyniły się do podniesienia jakości produktu końcowego, zapewniając jego niezawodność i zgodność z najnowszymi standardami w branży oprogramowania.

Projekt ten, dzięki swojej uniwersalności i przemyślanej strukturze, stanowi solidną bazę, którą można łatwo dostosowywać i rozwijać w odpowiedzi na ewoluujące wymagania użytkowników i dynamicznie zmieniające się środowisko technologiczne. Podsumowując, aplikacja ta nie tylko spełnia, ale również przekracza oczekiwania, stanowiąc wydajne i niezawodne narzędzie kompresji danych.