

Politechnika Śląska w Gliwicach
Wydział Automatyki, Elektroniki i Informatyki



Inżynieria Oprogramowania

“Binary file handling tool”

Wymagania funkcjonalne i нефункционалне

Autor:	Maciej Romanek, Klaudia Wójcik Karol Kabulak, Paweł Policha
Prowadzący :	Dr inż. Jakub Nalepa
Rok akademicki:	2018/2019
Kierunek:	Teleinformatyka
Rodzaj studiów:	Stacjonarne
Semestr:	4
Grupa:	2

Spis treści

1) Opis i założenia projektu.....	
2) Wymagania funkcjonalne.....	
a) Dodawanie plików.....	
b) Dodawanie wielu plików.....	
c) Zmiana funkcji dodawania pliku.....	
d) Wybranie ostatnio używanego formatu eksportowanego pliku	
e) Podgląd wybranego pliku.....	
f) Podanie nazwy pliku wyjściowego.....	
g) Eksportowanie.....	
3) Wymagania нефunkcjonalne.....	
4) Diagram sekwencji.....	

1. Opis i założenia projektu

Możliwość wglądu w zawartość plików binarnych, niejednokrotnie bywa problematyczne i wymaga specjalistycznych narzędzi umożliwiających obsługę konkretnego formatu. Pomocne może być więc stworzenie narzędzia dającego użytkownikowi swobodę w otwieraniu i eksportowaniu plików, do formy tekstu, bez względu na jego rozszerzenie.

Celem projektu jest stworzenie narzędzia umożliwiającego odczytywanie plików binarnych, a następnie umożliwienie wyeksportowania jego całości lub części do pliku. Użytkownik ma zdecydować o typie eksportowanego pliku. Przy wykonywaniu projektu należy rozważyć następujące problemy:

1. Struktura pliku powinna być określona przez użytkownika.
Narzędzie pozwalałoby na definiowanie zarówno liniowego jak i iterowanego pliku. Należy wziąć pod uwagę, że liczba iteracji może być zapisana wewnątrz pliku.
2. Po zdefiniowaniu, struktura pliku może zostać zserializowana.
Lista ostatnio używanych formatów eksportu powinna być dostępna z poziomu GUI. To samo dotyczy się ostatnio otwieranych folderów plików.
3. Celem zaoszczędzenia czasu użytkownika przy obsłudze GUI powinno mieć ono intuicyjną konstrukcję oraz być łatwe w obsłudze.
4. Dane odczytane z przekazanego pliku powinny być prezentowane w formie tekstu. Mają być one również zapisywane w tej formie, całkowicie lub częściowo.
5. Powinna zostać zaimplementowana funkcjonalność konwertowania wielu plików jednocześnie.

2. Wymagania funkcjonalne

a) Dodawanie plików

Funkcja:

AddFile()

Opis:

Funkcja uruchamiana przyciskiem "Add" z poziomu GUI programu. Uruchamia ona domyślny eksplorator systemu Windows umożliwiający nawigowanie i wybór pliku, który chcemy odczytać i poddać eksportowi. Jest to domyślna opcja uruchamiana wraz z programem.

Dane wejściowe:

Domyślna lokalizacja taka jak pulpit.

Źródło danych wejściowych:

Typ string.

Warunek poprawnego działania:

Sprawdzenie czy jest możliwe prawidłowe załadowanie pliku.

Warunek końcowy:

Zwrócenie ścieżki do wybranego pliku.

b) Dodawanie wielu plików

Funkcja:

AddMultipleFile()

Opis:

Funkcja uruchamiana przyciskiem "Add" z poziomu GUI programu. Uruchamia ona domyślny eksplorator systemu Windows umożliwiający nawigowanie i wybór wielu plików o tych samych rozszerzeniach. Następnie pliki przekazywane są do odczytu i eksportu.

Dane wejściowe:

Domyślna lokalizacja taka jak pulpit.

Źródło danych wejściowych:

Struktura zawierająca listę stringów.

Warunek poprawnego działania:

Sprawdzenie czy jest możliwe prawidłowe załadowanie plików.

Warunek końcowy:

Zwrócenie ścieżki do wybranego pliku.

c) Zmiana funkcji dodawania pliku**Funkcja:**

SwitchBatchProcessor()

Opis:

Funkcja wyzwalana zaznaczeniem pola "Add more files".

Odblokowuje ona możliwość użycia funkcji "AddMultipleFile()".

Odznaczenie pola spowoduje powrót do domyślnej funkcji obsługującej eksport i odczyt pojedynczego pliku.

Dane wejściowe:

Pole z możliwością zaznaczenia tak/nie.

Źródło danych wejściowych:

Typ bool.

Warunek poprawnego działania:

Sprawdzenie stanu pola.

Warunek końcowy:

Jeśli odznaczone, wykonanie domyślnej funkcji "AddFile()" po naciśnięciu przycisku "Add". Jeśli zaznaczone, wykonanie funkcji "AddMultipleFile()" po naciśnięciu przycisku "Add".

d) Wybranie ostatnio używanego formatu eksportowanego pliku**Funkcja:**

SelectLastExportFormat()

Opis:

Lista z możliwością wyboru ostatnio używanych rozszerzeń pliku do eksportu.

Dane wejściowe:

Lista formatów.

Źródło danych wejściowych:

Typ string.

Warunek poprawnego działania:

Sprawdzenie stanu wybranej pozycji z listy.

Warunek końcowy:

Przekazanie wybranego formatu do funkcji konwertującej.

e) Podgląd wybranego pliku**Funkcja:**

PreviewSelectedFile()

Opis:

Po wybraniu pliku do eksportu, a następnie po naciśnięciu przycisku "Preview", za pomocą domyślnego edytora tekstu systemu windows zostanie wyświetlona zawartość pliku.

Dane wejściowe:

Zawartość bitowa wybranego pliku.

Źródło danych wejściowych:

Plik binarny.

Warunek poprawnego działania:

Sprawdzenie czy jest możliwe otworzenie pliku.

Warunek końcowy:

Otwarcie edytora tekstu z zawartością pliku.

f) Podanie nazwy pliku wyjściowego**Funkcja:**

EndFileName()

Opis:

Pole tekstowe z możliwością wpisania nazwy wyjściowego pliku.

Dane wejściowe:

Nazwa pliku.

Źródło danych wejściowych:

Typ string.

Warunek poprawnego działania:

Sprawdzenie czy w polu tekstowym znajduje się treść. Jeśli nie, użycie domyślnej nazwy "Exported_file".

Warunek końcowy:

Przekazanie do funkcji eksportującej "ExportFile()" nazwy pliku wyjściowego.

g) Eksportowanie**Funkcja:**

ExportFile()

Opis:

Po naciśnięciu przycisku "Export", funkcja zacznie przetwarzać wybrany plik z wybranym rozszerzeniem.

Dane wejściowe:

Zawartość bitowa wybranego pliku.

Źródło danych wejściowych:

Plik binarny.

Warunek poprawnego działania:

Sprawdzenie czy jest możliwe otworzenie pliku.

Warunek końcowy:

Zapisanie wyeksportowanego pliku.

3. Wymagania niefunkcjonalne

Szybkość:

Brak ograniczeń co do czasu przetwarzania danych wejściowych.
Reakcja na sygnał użytkownika: 0,1 sekundy.

Rozmiar:

Małe pliki o kilobajtowych rozmiarach.

Łatwość użycia:

Czas przeszkolenia użytkownika w obsłudze systemu: 5 minut.
Liczba stron systemu pomocy: 3.

Niezawodność:

Prawdopodobieństwo niedostępności: znikome.
Częstotliwość występowania awarii: 1 na 100 użyc programu.

Odporność na błędy:

Czas restartu przy awarii: 10 - 20 sekund.

Przenośność:

Liczba systemów docelowych: 1.

3. Diagram sekwencji

