

# Inżynieria oprogramowania



Politechnika  
Śląska

## Projekt: Binary file handling tool

Część II - wersja 1

### **Skład sekcji:**

Marcin Przybylski  
Marcin Barczyk  
Mateusz Major  
Hubert Szymiski  
Maciej Walkowiak

---

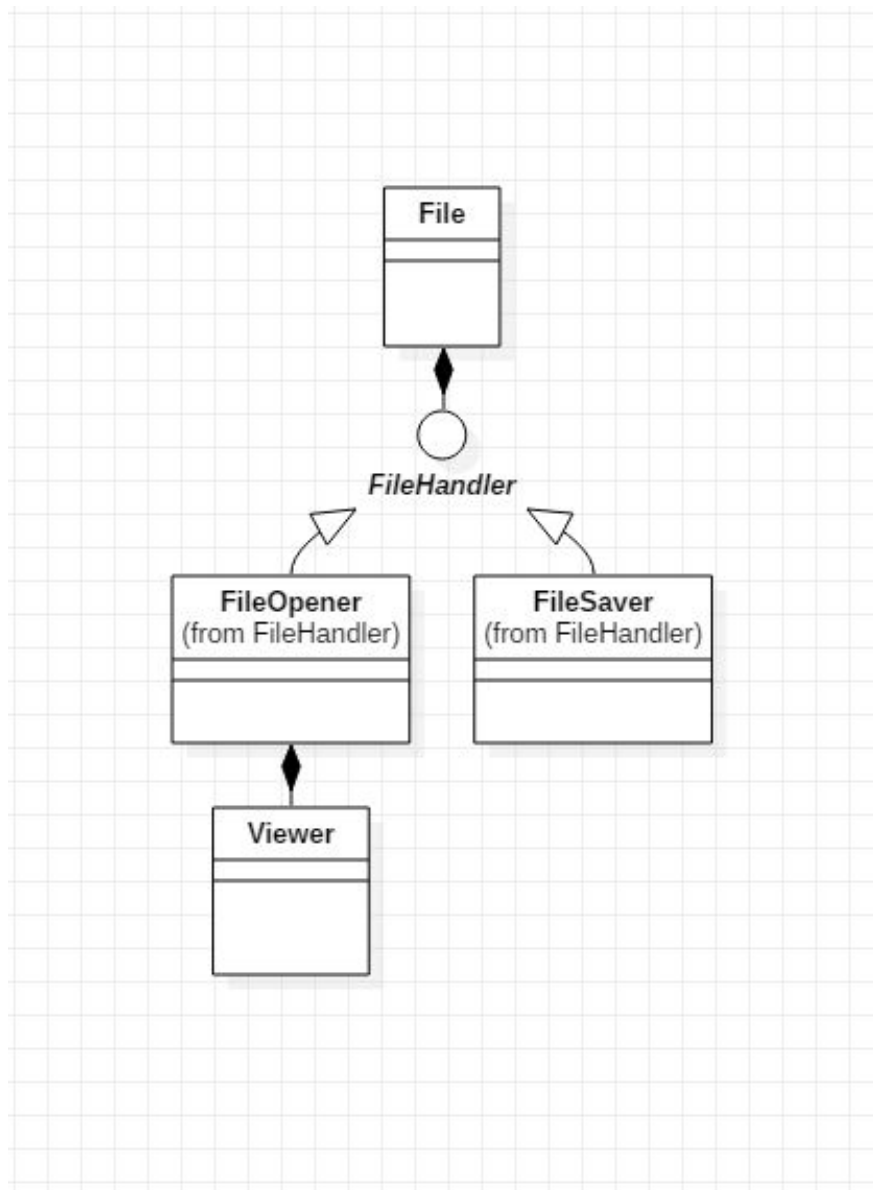
## **SPIS TREŚCI**

<b>Diagram klas</b> (Marcin Przybylski)	<b>3</b>
<b>Szczegółowy opis klas z diagramu</b> (Marcin Przybylski, Hubert Szymiski, Maciej Walkowiak)	<b>4</b>
<b>Model danych</b> (Marcin Barczyk, Mateusz Major)	<b>6</b>

---

## Diagram klas

---



rys 1. Diagram klas

# Szczegółowy opis klas z diagramu

---

## Klasa File

Zgodnie z diagramem klasa ta jest potrzebna do istnienia dalszych klas(kompozycja). Zawierać będzie podstawowe właściwości plików, na jakich będziemy operować, takie jak: nazwa, lokalizacja format pliku.

---

zmienne:

private:

String fileName - przechowuje ciąg znaków określający nazwę pliku.

String fileLocation - określa lokalizację pliku.

String fileFormat - zawiera format pliku (niezbędny do zapisu, np ".txt")

metody:

getterzy oraz setterzy do fileName, fileLocation, fileFormat.

---

## Interfejs FileHandler

Interfejs ten umożliwi schludne napisanie kodu. Będą go implementować dwie klasy.

---

zmienne:

File file - obiekt File j.w.

metody czysto wirtualne:

handle - przeznaczona będzie do implementacji odpowiednio odczytu/zapisu.

---

## **Klasa FileOpener**

Klasa implementująca interfejs FileHandler. Rozszerza go o zaimplementowanie metody handle jako otwierającej plik.

---

zmienne dodatkowe:

String opened - zawiera odczytany metodą handle plik.

metody:

handle - odpowiedzialna za otwarcie pliku. Operuje jako metoda na obiekcie typu File. W razie niepowodzenia wyświetla stosowny komunikat (za pomocą wyjątków).

---

## **Klasa FileSaver**

Druga klasa implementująca interfejs FileHandler. Implementuje metodę handle.

---

metody:

handle - odpowiedzialna za zapis pliku. Działa na obiekcie typu File. Przyjmuje jako argumenty int (0-5), który ma odzwierciedlenie jako format pliku (wektor o rozmiarze 6, np. ".txt", ".rtf", ".doc", ".docx", ".xml")

---

## **Klasa Viewer**

Klasa wymaga odczytanego pliku, aby mogła go wyświetlić(kompozycja).

---

metody:

view - metoda wyświetlająca wczytany do pamięci plik. Nie zmienia jego zawartości (const).

---

# Model danych

---

Przechowywanie danych odnośnie ostatnio używanych lokalizacji do zapisu pliku występowałoby w pliku tekstowym, który po wybraniu nowej lokalizacji byłby aktualizowany. Wybór danej lokalizacji byłby wektorem, który występuje w pamięci podręcznej programu. Dostęp do ostatnio używanej lokalizacji (pliku tekstowego) byłby zakodowany lub zabezpieczony aby tylko program miał do niej dostęp i niepożądane zmiany nie miały miejsca.

Lokalizacja pliku, który ma być poddany konwersji zostaje zapisana w pamięci programu, a po jego ponownym uruchomieniu jest zerowana.

Odczytane dane z pliku binarnego zapisywane są do pamięci programu jako wektor. W późniejszym etapie wykorzystywane są do konwersji na tekst.

Po konwersji dane z pamięci programu zostają zapisane do pliku tekstowego w formacie wybranym przez użytkownika, który podobnie jak ostatnie lokalizacje jest przechowywany w pliku tekstowym.

Wybór konwersji jednego pliku lub zestawu przechowywany jest w pamięci programu. Po ponownym uruchomieniu zmienna zostaje zresetowana.

Dostęp do wszystkich zmiennych, które znajdują się w pamięci programu, użytkownik będzie miał dostęp tylko w wyznaczonych momentach, przez co nie będzie mógł ich modyfikować np. podczas procesu konwersji.