Gliwice, 20 stycznia 2019

Programowanie Komputerów

Temat: Baza płyt z oprogramowaniem instalacyjnym

Autor: Radosław Serba
Semestr: trzeci
Grupa: II

Prowadzący: dr inż. Karolina Nurzyńska

1 Treść zadania

Zadaniem programu jest zarządzanie zbiorem plików z oprogramowaniem instalacyjnym. Zarządzanie obejmuje: przeszukiwanie zasobów, dodawanie nowych pozycji, modyfikacje oraz usuwanie istniejących rekordów.

2 Analiza rozwiązania

Głównym zadaniem programu jest przechowywanie informacji o przechowywanych plikach oprogramowania instalacyjnego w postaci wide stringów. Przy pierwszym otwarciu programu ten poinformuje nas o tym oraz o utworzeniu pliku konfiguracyjnego przechowującego repozytorium.

Następnie program wyświetli użytkownikowi wszystkie możliwe opcje do wykonania począwszy od zmiany lokalizacji repozytorium po dodawanie, usuwanie, edytowanie i wyszukiwanie plików wedle wskazanych kryteriów. Użytkownik porusza się w programie za pomocą uproszczonego menu głównego, przy czym aby się po menu poruszać należy naciskać przyciski cyfr odpowiadających funkcjom, które użytkownik chce wykorzystać.

Do wykonania programu zostały wykorzystane klasy **Repository** do obsługi zapisu pliku konfiguracyjnego przechowującego repozytorium, **Menu** do wyświetlania interfejsu użytkownika oraz do zapewnienia dostępu do wszystkich funkcjonalności programu, **RepoFile** do definicji plików instalacyjnych w postaci obiektów, **RepoManager** do obsługi listy wspomnianych plików instalacyjnych wraz ze wszystkimi operacjami wspomnianymi w treści zadania oraz **ValueChecker** jako klasa uzupełniająca do uniknięcia błędów wynikających z naturalnego działania niektórych funkcji w języku C++ dla wcześniej wspominanych klas.

3 Specyfikacja zewnętrzna

Program należy uruchomić poprzez kliknięcie dwukrotnie lewym przyciskiem myszy na ikonę w Eksploratorze Windows lub poprzez linię poleceń (cmd.exe) bez podania argumentów, w następujący sposób:
`<FileFinder.exe>`

Po uruchomieniu programu, w przypadku jego pierwszego uruchomienia wyświetli komunikat:
`This is the first run of FileFinder. The configuration file will be created in:`
`C:\Users\<nazwa użytkownika>\AppData\Roaming\FileFinder.cfg`

Następnie po naciśnięciu dowolnego klawisza zostaniemy przekierowani do menu głównego programu:
`WARNING: repository location is not set`

```
Welcome in Repofinder! Choose desired operation by typing a number:
|
|--(1) Set new repository location
|--(2) Show all files in repository
|--(3) Find specified Repo file
|--(4) Add Repo file
|--(5) Delete Repo file
|--(6) Edit Repo file
|--(7) Save changes into repository
|--(8) Help
|--(9) About the project
|--(0) Exit
```

Wybierając opcję (1) jesteśmy przenoszeni do podmenu pozwalającego na ustawienie nowego miejsca na repozytorium plików poprzez podanie odpowiedniej wartości:

```
The current repository location is: -
Please enter new repository location and press ENTER.
If you want to cancel repository change, please type ćanceląs a value and press ENTER.
```

W przypadku chęci wycofania się ze zmiany lokalizacji repozytorium należy wpisać **cancel** i zatwierdzić. W przypadku podania wartości komunikat o braku zdefiniowanego miejsca repozytorium po pierwszym uruchomieniu programu zniknie, konkretnie te pole:
`WARNING: repository location is not set`

Druga opcja w menu głównym, **(2) Show all files in repository** pozwala na przejrzenie wszystkich plików znajdujących się w repozytorium. W przypadku posiadania pustego repozytorium użytkownik zostanie poinformowany o tym

fakcie:

```
The list is empty.  There is nothing to watch.
Press any key to back to main menu.
```

W innym wypadku zostanie użytkownikowi wyświetlona zawartość repozytorium:

```
This is an actual list of files in the repository:
System file name:  windows.iso
Description:  Instalator Windowsa XP Home Edition SP3
User defined file name:  windows
Location:  NAS


Press any key to back to main menu.
```

Opcja **(3) Find specified Repo file** pozwala na znalezienie istniejącego elementu na liście poprzez jedną z wybranych opcji:

```
How do you want to find files?
|
|--(1) Through system file name - wyszukuje poprzez nazwę pliku w systemie
|--(2) Through description - wyszukuje poprzez opis pliku w systemie
|--(3) Through user defined file name - wyszukuje poprzez zdefiniowaną przez użytkownika nazwę pliku
|--(4) Through file location - wyszukuje poprzez lokalizację pliku
|--(5) Go back to the main menu - wraca do menu głównego
```

Wybierając jedną z interesujących kategorii jesteśmy pytani o podanie wybranej frazy wyszukiwania:

```
Please enter specified file system name to search:
```

Po zdefiniowaniu frazy w przypadku znalezienia wyniku pojawi się wynik jak na przykładzie:

```
The results are:
System file name:  windows.iso
Description:  Instalator Windowsa XP Home Edition SP3
User defined file name:  windows
Location:  NAS


Press any key to back to main menu.
```

W przypadku braku wyników dla wpisanej frazy nie zostanie wyświetlony żaden wynik i użytkownik zostanie poproszony o powrót do poprzedniego pola w menu. W przypadku braku plików do repozytorium użytkownik zostanie o tym poinformowany przed zdefiniowaniem parametru wyszukwania:

```
The actual list is empty.  There is nothing to search.

Press any key to continue.
```

Czwarta opcja, **(4) Add Repo file** pozwala na dodanie nowego pliku do repozytorium. Po wejściu w tę opcję użytkownik jest kolejno proszony o podane nazwę pliku w systemie, jej krótki opis, własną zdefiniowaną nazwę pliku oraz jego lokację. Po wypełnieniu pól pliku użytkownik widzi tego wynik, a nowy plik jest zapisywany w repozytorium:

```
You have added a new file with following data:
System file name:  windows
Description:  Instalator Windowsa XP Home Edition SP3
User defined file name:  windows xp.iso
Location:  NAS


Press anything to continue.
```

(5) Delete Repo file usuwa istniejący plik w repozytorium po wybraniu jednego z kryteriów wyszukiwania pliku, który ma być docelowo usunięty. W przypadku braku plików w repozytorium użytkownik otrzyma o tym komunikat:

```
The actual list is empty.  There is nothing to delete.

Press any key to continue.
```

W przypadku istnienia plików w repozytorium wybór opcji parametrów jest następujący:

```
How do you want to find files which are going to be delete?
|
|--(1) Through system file name - poprzez wyszukanie nazwy pliku w systemie
|--(2) Through description - poprzez wyszkukanie opisu pliku w systemie
|--(3) Through user defined file name - poprzez wyszukanie zdefiniowanej przez użytkownika nazwy pliku
|--(4) Through file location - poprzez wyszukanie lokalizacji pliku
|--(5) Forget about searching, just show me all files - poprzez wybranie pliku z listy wszystkich plików w repozytorium
|--(6) Go back to the main menu - opcja pozwala na powrót do menu głównego
```

W przypadku podopcji 1-4 po zdefiniowaniu parametru wyszukiwania program wyświetla dostępne pliki wraz z ich indeksami, a następnie oczekuje na wybranie przez użytkownika jednego z nich poprzez podanie wartości indeksu:

```
The actual file list is:  System file name:  windows xp.iso
Description:  Instalator Windowsa XP Home Edition SP3
User defined file name:  Windows
Location:  NAS



Please enter desired object index to be deleted:  1
Done, file deleted.
```

Opcja **(6) Edit Repo file** pozwala na edycję istniejących plików w repozytorium. Podobnie jak w przypadku dodawania i usuwania, jeśli repozytorium nie posiada żadnych plików użytkownik zostanie o tym poinformowany z brakiem możliwości wykorzystania dalszych opcji dopóki nie pojawi(ą) się w repozytorium plik(i). Poniżej wspomniany komunikat:

texttttThe actual list is empty. There is nothing to edit.

Please enter any key to continue.

W przypadku istnienia plików w repozytorium pojawi się lista wszystkich obiektów wraz z ich indeksami, a następnie użytkownik zostanie poproszony o wybranie indeksu w celu edycji.

```
The actual file list is:  System file name:  windows xp.iso
Description:  Instalator Windowsa XP Home Edition SP3
User defined file name:  Windows
Location:  NAS

Object index:  2
System file name:  linux.iso
Description:  Instalator Ubuntu 18.04 Desktop amd64
User defined file name:  Linux
Location:  NAS

Please enter desired object index to be edited:  1
```

Następnie należy wybrać właściwość pliku do edycji przy czym 1 to systemowa nazwa pliku, 2 to opis pliku, 3 to zdefiniowana przez użytkownika nazwa pliku, a 4 to lokalizacja pliku. Po wyborze należy wprowadzić zmienianą wartość.

```
Select edited value type:  (1) System filename, (2) File description, (3) User defined filename, (4) file location
1 - zdefiniowano zmianę systemowej nazwy pliku
Please enter the value you want to edit:  nowawartosc
Done, file edited.
```

```
Please enter any key to continue.
```

Po wykonaniu czynności użytkownik może wrócić do menu.

Opcja **(7) Save changes into repository** pozwala na zapis aktualnego stanu repozytorium do pliku. Lokalizacja pliku zawierającego bazę jest stała dla każdego uzytkownika systemu Windows: %appdata%\FileFinder.dat, np dla uzytkownika serba.r to C:\Users\serba.r\AppData\Roaming.
W przypadku jakiejkolwiek modyfikacji repozytorium (dodanie i usuwanie plików, edycja istniejących plików) w menu głównym programu pojawi się komunikat o niezapisanych zmianach:

```
WARNING: there are unsaved changes in the repository
```

Komunikat nie jest wyświetlany po zapisaniu zmian. Po uruchomieniu funkcji wykonywany jest zapis i po zakończeniu użytkownik otrzymuje komunikat (:

Saved successfully!
Press any key to go back to the main menu.

Opcja **(8) Help** wyświetla informacje na temat pomocy, która de facto kieruje do tego sprawozdania, opcja **(9) About the project** wyświetla informacje na temat projektu w języku angielskim jak i treść zadania oraz opcja **(0) Exit** kończy pracę programu.

4 Specyfikacja wewnętrzna

Klasa Repository:
Klasa służy do przechowywania, zapisu oraz odczytu z pliku informacji dotyczących lokalizacji repozytorium.

Repository();
Domyślny konstruktor klasy. Przypisuje wartość -"do RepoLocation oraz uruchamia metodę RepoStartup().

int FirstRun();
Metoda wywoływana przez RepoStartup() w przypadku braku pliku konfiguracyjnego repozytorium w katalogu użytkownika (%appdata%\FileFinder.cfg). Genereruje ona taki plik oraz zapisuje jej wartość domyślną RepoLocation.

void SetRepoLocation(std::wstring NewLocationAddress);
Ustawia nową wartość RepoLocation dla obiektu tej klasy.

std::wstring GetRepoLocation();
Wynikiem metody jest wartość RepoLocation dla obiektu tej klasy.

void RepoStartup();
Metoda sprawdza, czy plik konfiguracyjny istnieje. Jeśli tak - wczytuje wartość lokalizacji repozytorium, jeśli nie - uruchamia FirstRun().

std::wstring RepoLocation;
Pole definiujące miejsce repozytorium w systemie.

Klasa Menu:
Klasa ma za zadanie wygenerowanie użytkownikowi uproszczonego interfejsu (z wykorzystaniem prostych pętl) dzięki ktoremu będzie w stanie wykorzystać wszystkie założone funkcjonalności programu.

Menu(Repository startRepo, RepoManager startManager);
Konstruktor przyjmuje na wejściu referencję na obiekt repozytorium, aby można było względem niego wykonywać operacje. Podobnie wygląda w przypadku referencji na obiekt klasy RepoManager.

static void clearConsole();
Metoda statyczna służąca do wyczyszczenia konsoli w celu wygenerowania podmenu lub wyświetlenia na nowo konkretnej funkcjonalności w przyjazny dla oka sposób. Metoda zastąpiła początkowo wykorzystywanie odwołanie systemowe **cls**, co jest ogólnie przyjętą złą praktyką w budowaniu aplikacji konsolowych.

void ChangeRepoLocation();
Metoda zmienia aktualną wartość repozytorium z poziomu menu.

void IsRepositorySet();
Metoda sprawdza czy lokalizacja repozytorium została zmieniona po pierwszym uruchomieniu. Jeśli nie, wyświetla w menu głównym ostrzeżenie.

void IsChangesUnsaved();
Metoda sprawdza czy dokonano zmian na liście obiektu RepoManager. Jeśli tak, wyświetla w menu głównym ostrzeżenie.

void GenerateMainOptions();
Wyświetla elementy menu głównego.

void GenerateSearchOptions();
Wyświetla elementy podmenu dla wyszukiwania plików.

void GenerateDeleteOptions();
Wyświetla elementy podmenu dla usuwania plików.

void OpenMain();
Metoda otwiera menu główne (wywołuje metodę GenerateMainOptions(), IsRepositorySet(), IsChangeUnsaved()) i obsługuje wejścia do do podmenu.

void ShowAllFiles();
Metoda wywołująca listę wszystkich aktualnie obiektów znajdujących się na liście RepoManager jeśli nie jest pusta.

void FindRepoFiles();
Otwiera submenu służące do wyboru parametru wyszukiwania, wywołuje GenerateSearchOption(). W przypadku, gdy lista jest pusta wyświetla komunikat o tym i pozwala na powrót do menu głównego

void FindThroughSystemName();
Metoda prosi o podanie frazy wyszukiwania, a następnie przeszukuje listę sprawdzając czy fraza wyszukiwania jest taka sama jak pole fileSystemName. Jeśli tak - cały obiekt z pasującą wartością jest wyświetlany.

void FindThroughUserDefinedName();
Metoda prosi o podanie frazy wyszukiwania, a następnie przeszukuje listę sprawdzając czy fraza wyszukiwania jest taka sama jak pole fileUserDefinedName. Jeśli tak - cały obiekt z pasującą wartością jest wyświetlany.

void FindThroughDescription();
Metoda prosi o podanie frazy wyszukiwania, a następnie przeszukuje listę sprawdzając czy fraza wyszukiwania jest taka sama jak pole fileDescription. Jeśli tak - cały obiekt z pasującą wartością jest wyświetlany.

void FindThroughLocation();
Metoda prosi o podanie frazy wyszukiwania, a następnie przeszukuje listę sprawdzając czy fraza wyszukiwania jest taka sama jak pole fileLocation. Jeśli tak - cały obiekt z pasującą wartością jest wyświetlany.

void AddRepoFile();
Metoda dodaje nowy obiekt RepoFile do listy RepoManager przedtem pobierając dane do ich pól od użytkownika.
void DeleteRepoFile();
Otwiera submenu służące do wyboru parametru wyszukiwania do usunięcia pliku, wywołuje GenerateDeleteOption(). W przypadku, gdy lista jest pusta wyświetla komunikat o tym i pozwala na powrót do menu głównego

void SaveChanges();
Zapisuje całą listę typu RepoManager do pliku %appdata%\FileFinder.dat. Poprawne zapisanie pliku wyłącza komunikat o niezapisanych zmianach w menu głównym.
void Help();
Wyświetla krótką informację, aby jeśli użytkownik chciał się dowiedzieć więcej o obsłudze programu to niech by zajrzał do tego sprawozdania.

void About();
Wyświetla treść tego zadania i kilka słów od siebie w języku angielskim.

void DeleteThroughSystemName();
Metoda prosi o podanie frazy wyszukiwania, a następnie przeszukuje listę sprawdzając czy fraza wyszukiwania jest taka sama jak pole fileSystemName. Jeśli tak - cały obiekt z pasującą wartością jest wyświetlany wraz z przypisanym do obiektu indeksem. Następnie użytkownik wskazuje jeden z indeksów. Po wskazaniu dostępnej wartości metoda przesuwa

iterator o wartość z indeksu-1 i usuwa obiekt z listy.

void DeleteThroughDescription();
Metoda prosi o podanie frazy wyszukiwania, a następnie przeszukuje listę sprawdzając czy fraza wyszukiwania jest taka sama jak pole fileDescription. Jeśli tak - cały obiekt z pasującą wartością jest wyświetlany wraz z przypisanym do obiektu indeksem. Następnie użytkownik wskazuje jeden z indeksów. Po wskazaniu dostępnej wartości metoda przesuwa iterator o wartość z indeksu-1 i usuwa obiekt z listy.

void DeleteThroughUserDefinedName();
Metoda prosi o podanie frazy wyszukiwania, a następnie przeszukuje listę sprawdzając czy fraza wyszukiwania jest taka sama jak pole fileUserDefinedName. Jeśli tak - cały obiekt z pasującą wartością jest wyświetlany wraz z przypisanym do obiektu indeksem. Następnie użytkownik wskazuje jeden z indeksów. Po wskazaniu dostępnej wartości metoda przesuwa iterator o wartość z indeksu-1 i usuwa obiekt z listy.

void DeleteThroughLocation();
Metoda prosi o podanie frazy wyszukiwania, a następnie przeszukuje listę sprawdzając czy fraza wyszukiwania jest taka sama jak pole fileLocation. Jeśli tak - cały obiekt z pasującą wartością jest wyświetlany wraz z przypisanym do obiektu indeksem. Następnie użytkownik wskazuje jeden z indeksów. Po wskazaniu dostępnej wartości metoda przesuwa iterator o wartość z indeksu-1 i usuwa obiekt z listy.

void ShowAllFilesToDelete();
Metoda w stosunku do poprzednich wyświetla listę wszystkich obiektów wraz z indeksami. Następnie użytkownik wskazuje jeden z indeksów. Po wskazaniu dostępnej wartości metoda przesuwa iterator o wartość z indeksu-1 i usuwa obiekt z listy.

void EditRepoFile();
Metoda w przypadku niepustej listy wywołuje metodę EditFileFromWholeList() dla obiektu RepoManager mającej na celu faktyczną edycję wybranego obiektu.

Repository actualRepo;
Pole przechowujące referencję na wykorzystywany obiekt repozytorium.

RepoManager actualManager;
Pole przechowujące referencję na wykorzystywany obiekt RepoManager przechowujący listę.

Klasa RepoFile:
Klasa definiująca pliki znajdujące się w repozytorium.

RepoFile();
Domyślny konstruktor klasy.
RepoFile(std::wstring inputSystemName, std::wstring inputUserDefinedName, std::wstring inputFileLocation, std::wstring inputFileDesc);
Konstruktor pozwalający na zdefiniowanie wszystkich pól klasy.

virtual RepoFile();
Destruktor klasy.

void SetSystemName(std::wstring newFileName);
Ustawia wartość fileSystemName poprzez podany parametr.

std::wstring GetSystemName();
Zwraca aktualną wartość pola prywatnego fileSystemName.

void SetUserDefinedName(std::wstring newFileFileName);
Ustawia wartość fileUserDefinedName poprzez podany parametr.

std::wstring GetUserDefinedName();
Zwraca aktualną wartość pola prywatnego fileUserDefinedName.

void SetFileDesc(std::wstring newFiledesc);
Ustawia wartość fileDescription poprzez podany parametr.

std::wstring GetFileDesc();

Zwraca aktualną wartość pola prywatnego fileDescription.

void SetFileLocation(std::wstring newFilelocation);
Ustawia wartość fileLocation poprzez podany parametr.

std::wstring GetFileLocation();
Zwraca aktualną wartość pola prywatnego fileLocation.

RepoFile operator=(RepoFile right);
Operator przypisania dla klasy RepoFile.

friend std::wostream operator«(std::wostream output, RepoFile right);
Operator wypisania dla klasy RepoFile, wykorzystywany tylko do wyświetlania zawartości pól obiektu.

friend std::wistream operator»(std::wistream input, RepoFile right);
Operator wpisania dla klasy RepoFile, docelowo niewykorzystywany.

std::wstring fileSystemName;
Pole definiuje nazwę pliku w systemie.

std::wstring fileUserDefinedName;
Pole definiuje nazwę zdefiniowaną przez użytkownika dla pliku.

std::wstring fileDescription;
Pole definiuje opis pliku.

std::wstring fileLocation;
Pole definiuje lokalizację pliku.

Klasa RepoManager:
Klasa ma na celu dostarczenie listy przechowującej oraz dostarczenie jej pełnej obsługi (takie jak edytowanie, dodawanie, usuwanie i wyszukiwanie elementów).

RepoManager();
Konstruktor klasy. W praktyce wykorzystany raz, w main().

virtual RepoManager();
Destruktor klasy.

void AddToRepo(RepoFile inputRepoFile);
Dodaje obiekt na koniec listy oraz zaznacza zmianę w repozytorium.

void ShowAllFiles();
Wyświetla wszystkie obiekty na liście korzystając z operatora wypisania dla obiektu RepoFile.

bool IsRepoEmpty()
Zwraca wartość true w przypadku, gdy lista nie jest pusta. W przeciwnym wypadku zwraca false.
void FindInRepoBySystemName(std::wstring inputName);
Metoda iteruje po całej liście porównując pole fileSystemName z inputName i w przypadku tych samych wartości wyświetla cały obiekt.

void FindInRepoByDesc(std::wstring inputDesc);
Metoda iteruje po całej liście porównując pole fileDescription z inputDesc i w przypadku tych samych wartości wyświetla cały obiekt.

void FindInRepoByUserDefinedName(std::wstring inputFileName);
Metoda iteruje po całej liście porównując pole fileUserDefinedName z inputFileName i w przypadku tych samych wartości wyświetla cały obiekt.

void FindInRepoByLocation(std::wstring inputLocation);
Metoda iteruje po całej liście porównując pole fileLocation z inputLocation i w przypadku tych samych wartości wyświetla cały obiekt.

void RepoLoadFromFile();
Metoda sprawdza, czy plik w lokalizacji %appdata%\FileFinder.dat istnieje i jeśli tak, wczytuje zawartość do listy. Metoda jest wywoływana w trakcie startu programu.

bool RepoSaveToFile();
Metoda sprawdza, czy plik w lokalizacji %appdata%\FileFinder.dat istnieje i jeśli tak, zapisuje zawartość list do pliku.

void DeleteFromRepoBySystemName(std::wstring inputName);
Metoda iteruje po całej liście porównując pole fileSystemName z inputName i w przypadku tych samych wartości wyświetla cały obiekt z indeksem ze zmiennej iter oraz zapisaniem wartości iter do wektora listIter. Następnie użytkownik ma za zadanie wskazanie jednego z obiektów do usunięcia, a ten jest sprawdzany czy jest rzeczywiscie wartością int oraz czy podana wartość int znajduje się we wspomnianym wektorze. Po spełnieniu warunku advance przesuwa iterator o wartość indeksu-1 oraz usuwa wyświetlony obiekt z listy.

void DeleteFromRepoByDesc(std::wstring inputDesc);
Metoda iteruje po całej liście porównując pole fileDescription z inputDesc i w przypadku tych samych wartości wyświetla cały obiekt z indeksem ze zmiennej iter oraz zapisaniem wartości iter do wektora listIter. Następnie użytkownik ma za zadanie wskazanie jednego z obiektów do usunięcia, a ten jest sprawdzany czy jest rzeczywiscie wartością int oraz czy podana wartość int znajduje się we wspomnianym wektorze. Po spełnieniu warunku advance przesuwa iterator o wartość indeksu-1 oraz usuwa wyświetlony obiekt z listy.

void DeleteFromRepoByUserDefinedName(std::wstring inputFileName);
Metoda iteruje po całej liście porównując pole fileUserDefinedName z inputFileName i w przypadku tych samych wartości wyświetla cały obiekt z indeksem ze zmiennej iter oraz zapisaniem wartości iter do wektora listIter. Następnie użytkownik ma za zadanie wskazanie jednego z obiektów do usunięcia, a ten jest sprawdzany czy jest rzeczywiscie wartością int oraz czy podana wartość int znajduje się we wspomnianym wektorze. Po spełnieniu warunku advance przesuwa iterator o wartość indeksu-1 oraz usuwa wyświetlony obiekt z listy.

void DeleteFromRepoByLocation(std::wstring inputLocation);
Metoda iteruje po całej liście porównując pole fileLocation z inputLocation i w przypadku tych samych wartości wyświetla cały obiekt z indeksem ze zmiennej iter oraz zapisaniem wartości iter do wektora listIter. Następnie użytkownik ma za zadanie wskazanie jednego z obiektów do usunięcia, a ten jest sprawdzany czy jest rzeczywiscie wartością int oraz czy podana wartość int znajduje się we wspomnianym wektorze. Po spełnieniu warunku advance przesuwa iterator o wartość indeksu-1 oraz usuwa wyświetlony obiekt z listy.

void DeleteFromRepoWholeList();
Metoda iteruje po całej liście wyświetlając obiekty z indeksem ze zmiennej iter oraz zapisaniem wartości iter do wektora listIter. Następnie użytkownik ma za zadanie wskazanie jednego z obiektów do usunięcia, a ten jest sprawdzany czy jest rzeczywiscie wartością int oraz czy podana wartość int znajduje się we wspomnianym wektorze. Po spełnieniu warunku advance przesuwa iterator o wartość indeksu-1 oraz usuwa wyświetlony obiekt z listy.

void EditFileFromWholeList();
Metoda iteruje po całej liście wyświetlając obiekty z indeksem ze zmiennej iter oraz zapisaniem wartości iter do wektora listIter. Następnie użytkownik ma za zadanie wskazanie jednego z obiektów do edycji, a ten jest sprawdzany czy jest rzeczywiscie wartością int oraz czy podana wartość int znajduje się we wspomnianym wektorze. Po spełnieniu warunku advance przesuwa iterator o wartość indeksu-1 oraz wysyła iterator w parametrze do metody ChangeSelectedValueInsideList(). Po jej zakończeniu usuwa wskazany obiekt.

void ChangeSelectedValueInsideList(std::list<RepoFile>::iterator inputIterator);
Metoda tworzy kopię obiektu na który wskazuje iterator, a następnie użytkownik ma za zadanie wskazanie pola obiektu RepoFile do edycji za pomocą wartości 1-4 w zmiennej selectedValue, a ta jest sprawdzana czy jest rzeczywiscie wartością int oraz czy podana wartość int znajduje się w wektorze availableValues składającego się z wartości 1-4, przy czym 1 oznacza edycję pola fileSystemName, 2 oznacza edycję pola fileDescription, 3 oznacza edycję pola fileUserDefinedName i 4 oznacza edycję pola fileLocation. Edytowana jest kopia obiektu, a następnie po edycji kopia jest dodawana do listy.

std::list<RepoFile> repoList;
Pole listy przechowującej obiekty RepoFile będącymi plikami repozytorium.

std::wstring listPath;
Pole przechowujące lokalizację pliku przechowującego listę, konkretnie %appdata%\FileFinder.dat.

Klasa ValueChecker:
Klasa pomocnicza mająca na celu drobne poprawki w błędach działania programu.

static void IfInt();
Metoda statyczna sprawdzająca czy zczytywana wartość do zmiennej int jest rzeczywiście tego typu. Funkcja szczególnie przydatna w pętlach.

static void DeleteLastCharacter(std::wstring filePath);
Metoda statyczna mająca na celu usunięcie ostatniego znaku z pliku.

5 Testowanie

Test 1:
Program został uruchomiony w następujący sposób: FileFinder.exe
Wynik działania programu: ponieważ to było pierwsze uruchomienie programu, ten wyświetlił informację o utworzeniu pliku konfiguracyjnego dla repozytorium.

```
This is the first run of FileFinder.  The configuration file will be created in:
C:\Users\serba.r\AppData\Roaming\FileFinder.cfg
```

Test 2:
Program został uruchomiony po raz drugi w następujący sposób: FileFinder.exe
Wynik działania programu: ponieważ to nie było pierwsze uruchomienie programu, ten nie wyświetlił informacji o utworzeniu pliku konfiguracyjnego dla repozytorium, lecz od razu przeszedł do menu głównego programu:

```
WARNING: repository location is not set

Welcome in Repofinder!  Choose desired operation by typing a number:
|
|--(1) Set new repository location
|--(2) Show all files in repository
|--(3) Find specified Repo file
|--(4) Add Repo file
|--(5) Delete Repo file
|--(6) Edit Repo file
|--(7) Save changes into repository
|--(8) Help
|--(9) About the project
|--(0) Exit
```