

Aufgabe 2 - Geschichten vom ...

Lösungsidee:

Zuerst das Replacements file zeile für zeile lesen und jede zeile beim leerzeichen trennen und die beiden wörter aufteilen und diese in einem record repl mit old- und newword speichern.

Danach zeilenweise über das inputfile iterieren und die pos jedes gefunden replacements holen, solange die pos nicht 0 ist das oldWord löschen und an der gleichen stelle das newWord einfügen. Danach die veränderte Zeile ins output file schreiben.

Meine erste Variante habe ich mit funktionen wie SplitString und ReplaceString und dynamic arrays implementiert, aber ein Studienkolleg hat mich darauf hingewiesen das wir nur funktionen aus der Pascal Standardfunktionen Datei verwenden dürfen, also hab ich meinen code nochmal umgeändert. Falls Interesse besteht, habe ich in dem mitabgegebenen zip file die andere Variante auch mit rein gegeben. :D

Zeitaufwand: 45min

Code:

```
program StoryGen;

uses SysUtils;

const
    MAX_REPL_SIZE = 1000;

type
    Repl = record
        OldWord: string;
        NewWord: string;
    end;

var
    repls: array[1..MAX_REPL_SIZE] of Repl;
    replsSize: integer;

procedure CheckIfFileExists(fileName: string);
begin
    if not FileExists(fileName) then
    begin
        WriteLn('Error: file does not exist - ', fileName);
        writeln;
        Halt;
    end;
```

```

end;

procedure CheckFileNamesIdentical(file1, file2: string);
begin
    if (file1 = file2) then
    begin
        WriteLn('Error: file can not be the same - ', file1);
        writeln;
        Halt;
    end;
end;

function GetFilename(paramInt: Integer; msg: string): string;
var
    fileName: string;
begin
    if ParamCount > (paramInt - 1) then
        fileName := ParamStr(paramInt)
    else begin
        write(msg, ' > ');
        ReadLn(fileName);
    end;
    GetFilename := fileName;
end;

procedure getReplFromStr(str: string; var repl: Repl);
var
    whitespacePos: Integer;
begin
    whitespacePos := Pos(' ', str);

    if(whitespacePos = 0) then
    begin
        WriteLn('Error: incorrect format of replacements file');
        writeln;
        Halt;
    end;

    repl.OldWord := Copy(str, 1, whitespacePos - 1);
    repl.NewWord := Copy(str, whitespacePos + 1, Length(str) -
whitespacePos);
end;

procedure readRepls(fileName: string);
var
    replFile: TEXT;

```

```

    line: string;
    foundRepl: Repl;
begin
    assign(replFile, fileName);
    reset(replFile);

    replsSize := 0;

    while not eof(replFile) do
    begin
        readln(replFile, line);

        getReplFromStr(line, foundRepl);
        Inc(replsSize);

        repls[replsSize].OldWord := foundRepl.OldWord;
        repls[replsSize].NewWord := foundRepl.NewWord;
    end;

    close(replFile);
end;

procedure openFiles(var inFile, outFile: TEXT; inFileName, outFileName:
string);
begin
    Assign(inFile, inFileName);
    Reset(inFile);
    Assign(outFile, outFileName);
    Rewrite(outFile);
end;

procedure closeFiles(var inFile, outFile: TEXT);
begin
    Close(inFile);
    Close(outFile);
end;

procedure replaceWords(var line: string);
var
    i, wPos: integer;
begin
    for i := 1 to (replsSize) do
    begin
        wPos := Pos(repls[i].OldWord, line);
        while(wPos <> 0) do
            begin

```

```

        Delete(line, wPos, Length(repls[i].OldWord));
        Insert(repls[i].NewWord, line, wPos);
        wPos := Pos(repls[i].OldWord, line);
    end;
end;
end;

procedure runReplacements(inFileName, outFileName: string);
var
    line: string;
    inFile, outFile: TEXT;
begin
    openFiles(inFile, outFile, inFileName, outFileName);

    while(not Eof(inFile)) do
    begin
        ReadLn(inFile, line);
        replaceWords(line);
        writeln(outFile, line);
    end;

    closeFiles(inFile, outFile);
end;

var
    replsFileName, inFileName, outFileName: string;
begin
    replsFileName := GetFilename(1, 'Enter fileName with the
replacements');
    CheckIfFileExists(replsFileName);

    inFileName := GetFilename(2, 'enter text infilename');
    CheckIfFileExists(inFileName);
    CheckFileNamesIdentical(inFileName, replsFileName);

    outFileName := GetFilename(3, 'enter text outfilename');
    CheckFileNamesIdentical(outFileName, replsFileName);
    CheckFileNamesIdentical(outFileName, inFileName);

    readRepls(replsFileName);
    runReplacements(inFileName, outFileName);
end.

```

Tests:

```
StoryGen.pas M repls.txt M X
UE4 > hu > repls.txt
    You, 58 seconds ago | 1 author (You)
1 | Osterhasen Christkind
2 | Osterhase Christkind
3 | Ostern Weihnachten
4 | Oster Weihnacht
5 |
```

Natürlich könnte man noch mehr Einträge in der Replacement Datei vornehmen, um die Geschichte authentischer wirken zu lassen, aber denke das ist nicht der Sinn der Übung :D

```
C:\_data\fh-repos\2023SS_ADF\UE4\hu>StoryGen repls.txt Ostern.txt Weihnachten.txt
Heap dump by heaptrc unit of C:\_data\fh-repos\2023SS_ADF\UE4\hu\StoryGen.exe
1651 memory blocks allocated : 82649/88200
1651 memory blocks freed    : 82649/88200
0 unfreed memory blocks : 0
True heap size : 294912 (176 used in System startup)
True free heap : 294736
```

```
StoryGen.pas M Ostern.txt U X
UE4 > hu > Ostern.txt
> Osterhase Aa ab * ? of 6
```

```
StoryGen.pas M Ostern.txt U Weihnachten.txt U X
UE4 > hu > Weihnachten.txt
> Christkind Aa ab * 1 of 6 ↑ ↓ ≡ ×
```