

Aufgabe 2 - Geschichten vom ...

Lösungsidee:

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

Danach zeilenweise über das inputfile iterieren und die funktion StringReplace für jedes repl auf jeder zeile anwenden und die veränderte zeile ins outputfile schreiben.

Ich denke das funktion wie SplitString und ReplaceString wahrscheinlich nicht so gern gesehen sind, aber meiner Meinung nach solange es nicht in der Angabe steht sollte es mir erlaubt sein alle Möglichkeiten der Sprache Pascal zu nutzen. (gleiche gilt für dynamic arrays)

Zeitaufwand: 45min

Code:

```
program StoryGen;

uses SysUtils, StrUtils;

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

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;

var
    repls: array of Repl;

procedure readRepls(fileName: string);
var
    replFile: TEXT;
    line: string;
    words: array of AnsiString;
begin
    assign(replFile, fileName);
    reset(replFile);

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

            words := SplitString(line, ' ');
            if (not (High(words) = 1)) then
                begin
                    WriteLn('Error: incorrect format of replacements file');
                    writeln;
                    Halt;
                end;

            SetLength(repls, Length(repls) + 1);

            repls[Length(repls) - 1].OldWord := words[0];
            repls[Length(repls) - 1].NewWord := words[1];
        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: integer;
begin
    for i := 0 to Length(repls) - 1 do
        line := StringReplace(line, repls[i].OldWord, repls[i].NewWord,
[rfReplaceAll, rfIgnoreCase]);
    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');

```

```

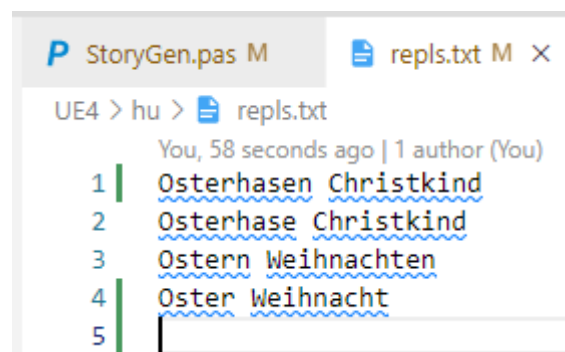
CheckIfFileExists(inFileName);
CheckFileNamesIdentical(inFileName, replsFileName);

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

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

```

Tests:



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

```

