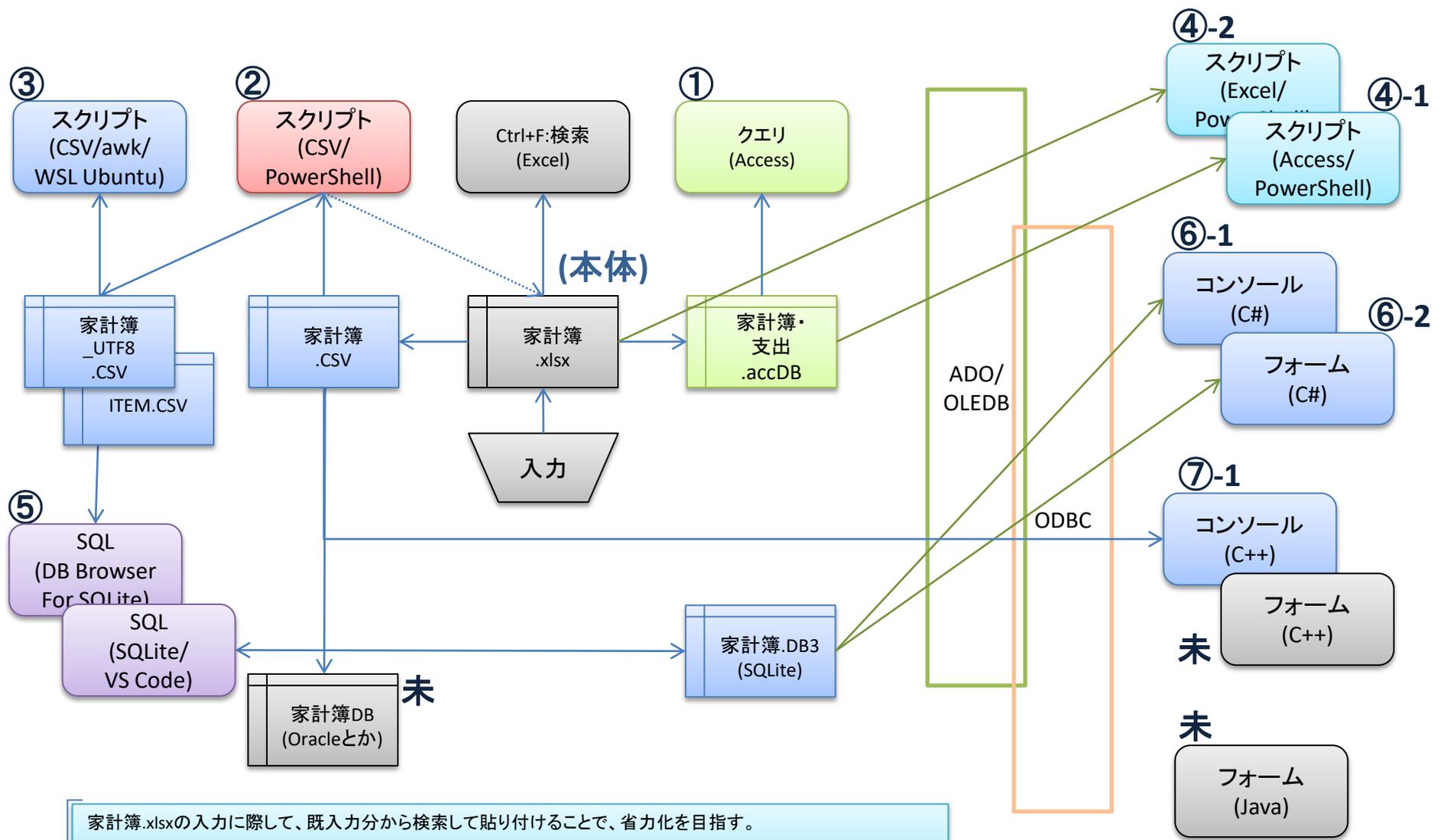


モジュール設計仕様書	システム	家計簿・支出. ACCDB	家計簿入力補助・履歴検索	作成日	2020/5/3
補足	モジュール	クエリ	支出クエリ (家計簿入力補助)	作成者	hosiyamakaze

DFD:全体像



家計簿.xlsxの入力に際して、既入力分から検索して貼り付けることで、省力化を目指す。
 いろいろ兼ねた試行錯誤である。

モジュール設計仕様書	システム	家計簿・支出.ACCDB	家計簿入力補助・履歴検索	作成日	2020/5/3
補足	モジュール	クエリ	支出クエリ(家計簿入力補助)	作成者	hosiyamakaze

DFD:①支出クエリ(家計簿入力補助) Access 実装

No.	日付	大分類	中分類	小分類	項目
17154	17153	2020/5/1(金)	食費	飲料	紅茶
17155	17154	2020/5/1(金)	食費	野菜類	ニラ
17156	17155	2020/5/1(金)	食費	飲料	ミルク
17157	17156	2020/5/1(金)	食費	野菜類	キュウリ
17158	17157	2020/5/1(金)	食費	野菜類	もやし
17159	17158	2020/5/1(金)	食費	野菜類	キノコ
17160	17159	2020/5/1(金)	食費	税金等	消費税
17161	17160	2020/5/1(金)	食費	税金等	消費税

- ① (準備)外部データ⇒保存済みのインポート操作で、家計簿.xlsxの範囲「支出」を支出テーブルにインポートする。
- ② 支出クエリ(家計簿入力補助)をダブルクリックで実行する(「PowerShellからAccessのクエリを起動する」でも可)
- ③ 検索文字列を入力する(「ポカリ*1.5」とかでもよい)
- ④ 該当項目をコピー&ペーストする(見出しが張り付くのでテキストウに処理を)
- ⑤ 2件目以降の検索は、ホームの「表示」をダブルクリックすると③検索文字列入力に戻る
(Excelの範囲「支出」をときどき最新化して取り込み直するとよい。このインポート操作定義はソース管理に保存されない)

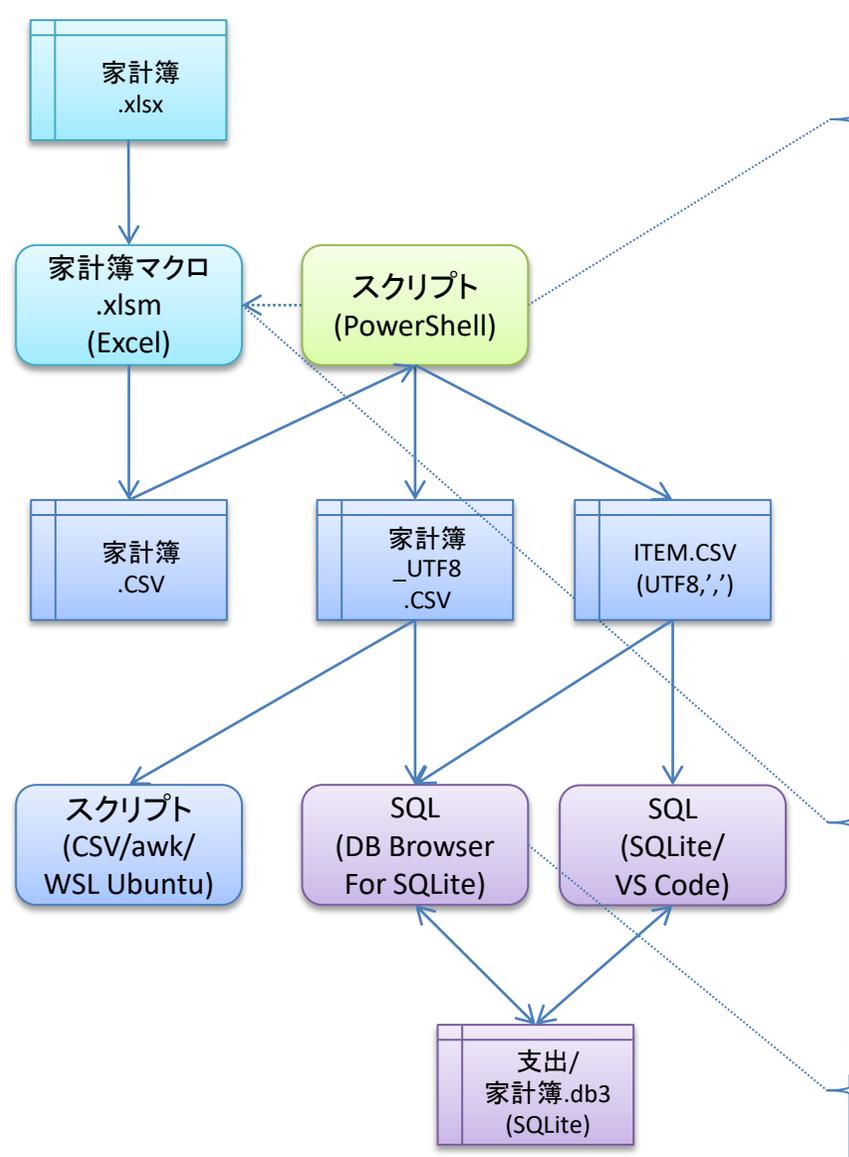
```
#PowerShellからAccessのクエリを起動する
$AccDB = "N:/Access/家計簿・支出.accdb";
$AppAc = New-Object -ComObject Access.Application
$AppAc.OpenCurrentDatabase($AccDB)
$AppAc.Visible = $true
$AppAc.DoCmd.OpenQuery("支出クエリ(家計簿入力補助)")
```

No.	日付	大分類	中分類	小分類	項目
17154	17153	2020/5/1(金)	食費	飲料	紅茶
17155	17154	2020/5/1(金)	食費	野菜類	ニラ
17156	17155	2020/5/1(金)	食費	飲料	ミルク
17157	17156	2020/5/1(金)	食費	野菜類	キュウリ
17158	17157	2020/5/1(金)	食費	野菜類	もやし
17159	17158	2020/5/1(金)	食費	野菜類	キノコ
17160	17159	2020/5/1(金)	食費	税金等	消費税
17161	17160	2020/5/1(金)	食費	税金等	消費税
17162			食費	野菜類	キャベツ
17163					キャベツ 1/2

SELECT DISTINCT 支出.大分類, 支出.中分類, 支出.小分類, 支出.項目
FROM 支出
WHERE (((支出.小分類) Like "*" + [検索文字列] + "*"))
OR (((支出.項目) Like "*" + [検索文字列] + "*"));

モジュール設計仕様書	システム	家計簿マクロ.xlsm	家計簿入力補助・履歴検索	作成日	2020/5/3
補足	モジュール	PowerShell - CSV	家計簿支出検索.ps1スクリプト(家計簿入力補助)(CSV)	作成者	hosiyamakaze

DFD: ②-1, ②-2家計簿支出検索(家計簿入力補助)(CSV) CSVファイル作成 PowerShell 実装



```

#②-1. 家計簿.xlsxの支出をCSV(tab区切り)出力する(家計簿マクロ.xlsmのマクロを実行する)
# ※範囲「支出」としたいところだが家計簿シート全体を出力する
Write-Host -ForegroundColor Yellow "家計簿.csvを更新中..."
$Appxl = New-Object -ComObject Excel.Application
$Appxl.Visible = $false
$Appxl.DisplayAlerts = $false
$xlbk = $Appxl.Workbooks.Open("N:¥Excel¥家計簿マクロ.xlsm")
$Appxl.RUN("Module2.支出をcsv出力する","N:¥Excel¥家計簿.xlsx","G:¥CMD¥家計簿¥家計簿.CSV")
$Appxl.Quit()
$Appxl = $null
Write-Host -ForegroundColor Yellow "家計簿.csvを更新しました。"

#②-2.UTF-8版CSVを更新する(awk,SQLiteの入力)
Write-Host -ForegroundColor Yellow "{家計簿_UTF8,ITEM}.csvを更新中..."
$itemCSV = get-content -encoding Default -path ../家計簿/家計簿.csv
$itemCSV | out-file ../家計簿/家計簿_UTF8.csv -Encoding UTF8
$itemCSV -replace("`t",";") | out-file ../SQLite/ITEM.csv -Encoding UTF8
Write-Host -ForegroundColor Yellow "{家計簿_UTF8,ITEM}.csvを更新しました。"

```

```

Module2:
Sub 支出をCSV出力する(Optional srcXlsx As String = "N:¥Excel¥家計簿.xlsx", _
    Optional dstCSV As String = "N:¥CMD¥家計簿¥家計簿.CSV")
'
' 支出をCSV出力する Macro
'
'
Workbooks.Open Filename:=srcXlsx
ActiveWorkbook.SaveAs Filename:=dstCSV, FileFormat:=xlText, CreateBackup:=False
ActiveWorkbook.Close SaveChanges:=False
End Sub

```

DB Browser for SQLiteの「File⇒Import⇒Table from CSV...」で家計簿_UTF8.csv (UFT8,tab)またはITEM.csv(UTF8,',')を家計簿.db3の[支出]テーブルに取り込む

モジュール設計仕様書	システム	家計簿. CSV	家計簿入力補助・履歴検索	作成日	2020/5/3
補足	モジュール	PowerShell - CSV	家計簿支出検索. ps1スクリプト(家計簿入力補助)(CSV)	作成者	hosiyamakaze

DFD:②-3家計簿支出検索(家計簿入力補助)(CSV) PowerShell 実装

```

Windows PowerShell ISE
ファイル(F) 編集(E) 表示(V) ツール(T) デバッグ(D) アドオン(A) ヘルプ(H)
テンプレート - work.ps1 X
# ----- 家計簿支出検索(家計簿入力補助)
29 $ItemPath = Get-Item -Path "../家計簿/家計簿.csv"
30 $srchText = "__initial_load_csv__"
31 while ($srchText.Length -gt 0){
32     if ($srchText -eq "__initial_load_csv__"){
33         #家計簿.csvを取り込む
34         $itemCSV = ConvertFrom-Csv -InputObject ( Get-Content -Path $ItemPath) -Delimiter `t
35     }else{
36         $itemCSV | Where-Object {
37             ($_.小分類 -match($srchText)) -or ($_.項目 -match($srchText))
38         } | ForEach-Object {
39             $keyStr = $_.小分類
40             if ($itemList.$keyStr) {
41                 $itemList.$keyStr += "`t" + $_.項目
42             } else {
43                 $itemList.$keyStr = $_.項目
44             }
45         }
46     }
47     #検索結果
48     $itemList
49 }
50
51 Write-Host -ForegroundColor green -NoNewline "検索文字列(Enter:終了)="
52 $srchText = Read-Host
53
54
55

```

(準備)家計簿.xlsxの支出データをtab区切りtextに出力する。
 ① 家計簿支出検索スクリプトを選択する
 ② 「選択項目を実行」し、検索文字列を入力する。
 ③ 該当項目をコピー&ペーストでExcelに貼り付ける
 tabが効いているのでセルに区切られて収まる。
 ※ずれる場合は、前ページ④で、区切り位置をtabに戻す
 (家計簿.CSVをときどき最新化するとよい。②-1参照)

#②-3.家計簿.csvから検索する
 \$ItemPath = Get-Item -Path "..¥家計簿¥家計簿.csv"
 \$srchText = "__initial_load_csv__"
 while (\$srchText.Length -gt 0){
 if(\$srchText -eq "__initial_load_csv__"){
 #家計簿.csvを取り込む
 \$itemCSV = Import-Csv -Delimiter `t -Encoding Default -Path \$ItemPath
 }else{
 \$itemCSV | Where-Object {
 (\$_.小分類 -match(\$srchText)) -or (\$_.項目 -match(\$srchText))
 } | ForEach-Object -Process{
 \$_.大分類+"`t"+\$_.中分類+"`t"+\$_.小分類+"`t"+\$_.項目;
 } | Sort-Object -Unique
 }
 Write-Host -ForegroundColor green -NoNewline "検索文字列(Enter:終了)="
 \$srchText = Read-Host
 }
 ※変更済

```

検索文字列(Enter:終了)=キャベツ
食費 野菜類 カット野菜 セレクト キャベツ野菜炒め
食費 野菜類 カット野菜 キャベツ野菜炒め
食費 加工食品 おでん FFおでん ロールキャベツ
食費 野菜類 カット野菜 キャベツ千切り
食費 野菜類 キャベツ キャベツ
食費 野菜類 キャベツ キャベツ 1/2
食費 野菜類 カット野菜 キャベツ&レタス
検索文字列(Enter:終了)=ポカリ
食費 飲料 スポーツ飲料 スポーティ 大塚製薬 ポカリスエット 1500ml×8本
食費 飲料 スポーツ飲料 スポーティ 大塚製薬 ポカリスエット イオンウォーター 900ml×12本
食費 飲料 スポーツ飲料 スポーティ ポカリスエット 1.5L
食費 飲料 スポーツ飲料 スポーティ ポカリ イオンウォーター
食費 飲料 スポーツ飲料 スポーティ ポカリスエット 500ml
食費 飲料 スポーツ飲料 スポーティ ポカリスエット 900ml
食費 飲料 スポーツ飲料 スポーティ ポカリスエット
食費 飲料 スポーツ飲料 スポーティ ポカリスエット イオンウォーター 900ml
食費 飲料 スポーツ飲料 スポーティ ポカリスエット イオンウォーター 500ml
食費 飲料 スポーツ飲料 スポーティ ポカリスエット 500ml
検索文字列(Enter:終了)=ポカリ.*1.5
食費 飲料 スポーツ飲料 スポーティ ポカリスエット 1.5L
検索文字列(Enter:終了)=
PS G:¥CMD¥ps>

```

家計簿.xlsx - Microsoft Excel

1	No.	日付	大分類	中分類	小分類	項目	数量
17154	17153	2020/5/1(金)	食費	飲料	紅茶	デイリークラブ ティーバッグ	
17155	17154	2020/5/1(金)	食費	野菜類	ニラ	ニラ	
17156	17155	2020/5/1(金)	食費	飲料	ミルク	森永 グループ 袋 200g	
17157	17156	2020/5/1(金)	食費	野菜類	キュウリ	キュウリ	
17158	17157	2020/5/1(金)	食費	野菜類	もやし	もやし	
17159	17158	2020/5/1(金)	食費	野菜類	キノコ	舞茸	
17160	17159	2020/5/1(金)	食費	税金等	消費税	消費税	
17161	17160	2020/5/1(金)	食費	税金等	消費税	キャッシュレス還元額	
17162			食費	野菜類	キャベツ	キャベツ 1/2	
17163							
17164							
17165							
17166							
17167							

モジュール設計仕様書	システム	家計簿_UTF8.CSV	家計簿入力補助・履歴検索	作成日	2020/5/3
補足	モジュール	AWK - CSV	家計簿支出検索.awkスクリプト(家計簿入力補助)	作成者	hosiyamakaze

DFD:③家計簿支出検索(家計簿入力補助)(CSV) AWK(WSL Ubuntu) 実装

```

#UTF8,LF $Workfile: 家計簿支出検索.awk $$Revision: 1 $$Date: 20/04/29 12:50 $
#NoKeywords: $
#
# Usage:awk -v srchText='ポカリ.*1.5' -f ../awk/家計簿支出検索.awk ../家計簿/家計簿_UTF8.csv
# UTF8,LF、tab区切り、項目の並びは家計簿.xlsxに準じる(col4:大分類,col5:中分類,col6:小分類,col7:項目)
#
BEGIN{
  FS="\t";
  OFS="\t";
}
$6 ~ srchText || $7 ~ srchText{
  itemData=$4 "\t" $5 "\t" $6 "\t" $7;
  itemList[itemData]=1;
}
END{
  for (items in itemList) {print items;}
}

```

```

hosiya@Pavilion13:/gen/CMD/awk$
hosiya@Pavilion13:/gen/CMD/awk$ awk -v srchText='キャベツ' -f ../awk/家計簿支出検索.awk ../家計簿/家計簿_UTF8.csv
食費 加工食品 おでん FFおでん ロールキャベツ
食費 野菜類 カット野菜 キャベツ千切り
食費 野菜類 キャベツ キャベツ 1/2
食費 野菜類 カット野菜 キャベツ&レタス
食費 野菜類 カット野菜 セレクト キャベツ野菜炒め
食費 野菜類 カット野菜 キャベツ野菜炒め
食費 野菜類 キャベツ キャベツ
hosiya@Pavilion13:/gen/CMD/awk$ awk -v srchText='ポカリ.*1.5' -f ../awk/家計簿支出検索.awk ../家計簿/家計簿_UTF8.csv
食費 飲料 スポーツ飲料 ポカリスエット 1.5L
hosiya@Pavilion13:/gen/CMD/awk$

```

家計簿.xlsx - Microsoft Excel

	A	B	D	E	F	
1	No.	日付	大分類	中分類	小分類	項目
17154	17153	2020/5/1(金)	食費	飲料	紅茶	デイリークラブ ティーバッグ
17155	17154	2020/5/1(金)	食費	野菜類	ニラ	ニラ
17156	17155	2020/5/1(金)	食費	飲料	ミルク	森永 クリープ 袋 200g
17157	17156	2020/5/1(金)	食費	野菜類	キュウリ	キュウリ
17158	17157	2020/5/1(金)	食費	野菜類	もやし	もやし
			食費	野菜類	キノコ	舞茸
			食費	税金等	消費税	消費税
			食費	税金等	消費税	消費税
			食費	税金等	消費税	キャッシュレス還元額
			食費	野菜類	キャベツ	キャベツ

(準備)家計簿.xlsxの支出データをtab区切りtextに出力し、UTF8,LFに変換する。(SJISだと正しくマッチしない文字がある)

- ① awkコマンドライン(Usage:)に検索文字列を設定する('ポカリ.*1.5'とかでもよい)
- ② awkコマンドラインを選択し、「ターミナル」⇒「選択したテキストを実行」する
- ③ 該当項目をコピー&ペーストでExcelに貼り付ける
- ④ tabがスペースになっているのでテキストに処理を(「データ」⇒「区切り位置」などする)
(家計簿_UTF8.csvをときどき最新化するとよい。②-2参照)

モジュール設計仕様書	システム	家計簿・支出.accdb	家計簿入力補助・履歴検索	作成日	2020/5/3
補足	モジュール	PowerShell - ADO - accdb	家計簿支出検索.ps1 (家計簿入力補助) (ADO-accdb)	作成者	hosiyamakaze

DFD:④-1 家計簿支出検索(家計簿入力補助) ADO-accdb PowerShell 実装

```

Windows PowerShell ISE (x86)
ファイル(F) 編集(E) 表示(V) ツール(T) デバッグ(D) アドオン(A) ヘルプ(H)
テンプレート - work.ps1 X
1 # ----- 家計簿支出検索(家計簿入力補助)(Accessから検索(ADO))
2 #※PowerShellISE(x86):32ビットで実行すること、ワイルドカードは%で(*はダメ)。
3 $AccDB = "N:/Access/家計簿・支出.accdb";
4 $CursorType=3; #adOpenStatic(3)/adOpenKeyset(2)
5 $LockType=1; #adLockOptimistic(3)/adLockReadOnly(1)
6 $srchText = "__initial_connection__"
7 while ($srchText.Length -gt 0) {
8     if($srchText -eq "__initial_connection__"){
9         $ADOcon = New-Object -ComObject ADODB.Connection;
10        $connectionString = "Provider = Microsoft.ACE.OLEDB.12.0; Data Source = " + $AccDB;
11        $ADOcon.Open($connectionString);
12        $ADOrcs = New-Object -ComObject ADODB.Recordset;
13    }else{
14        $sql="SELECT DISTINCT 支出.大分類 + chr(9) + 支出.中分類 + chr(9) + 支出.小分類 + chr(9) + 支出.項目 as 対象項目 "+
15        " FROM 支出 "+
16        " WHERE ((支出.小分類 Like '%' + $srchText + '%') OR ((支出.項目 Like '%' + $srchText + '%'))); "
17        $ADOrcs.Open($sql, $ADOcon, $CursorType,$LockType); #, $CursorType,$LockTypeを省くとRecordCountが-1.
18        while( $ADOrcs.EOF -eq $false ) {
19            Write-Host $ADOrcs.Fields.Item("対象項目").Value;
20            $ADOrcs.MoveNext();
21        }
22        $ADOrcs.Close()
23    }
24    Write-Host -ForegroundColor green -NoNewline "検索文字列(Enter:終了)="
25    $srchText = Read-Host
26 }
27 $ADOcon.Close()
28 $ADOrcs = $null
29 $ADOcon = $null
30
31 $ADOrcs = $null
32 $ADOcon = $null
33
34
35

```

(準備)家計簿.xlsxの支出データをaccdbに取り込む。
 ① 家計簿支出検索スクリプトを選択する
 ② 「選択項目を実行」し、検索文字列を入力する。
 ③ 該当項目をコピー&ペーストでExcelに貼り付ける
 tabが効いているのでセルに区切られて収まる。
 ※ワイルドカードは%で(*はダメ)。
 (accdbをときどき最新化するとよい)。
 Microsoft Access データベース エンジン 2010 再頒布可能コンポーネント AccessDatabaseEngine 32bit版のインストールが必要であった。(Excelにも使える)

```

# ----- 家計簿支出検索(家計簿入力補助)(Accessから検索(ADO))
#※PowerShellISE(x86):32ビットで実行すること、ワイルドカードは%で(*はダメ)。
$AccDB = "N:/Access/家計簿・支出.accdb";
$CursorType=3; #adOpenStatic(3)/adOpenKeyset(2)
$LockType=1; #adLockOptimistic(3)/adLockReadOnly(1)
$srchText = "__initial_connection__"
while ($srchText.Length -gt 0) {
    if($srchText -eq "__initial_connection__"){
        $ADOcon = New-Object -ComObject ADODB.Connection;
        $connectionString = "Provider = Microsoft.ACE.OLEDB.12.0; Data Source = " + $AccDB;
        $ADOcon.Open($connectionString);
        $ADOrcs = New-Object -ComObject ADODB.Recordset;
    }else{
        $sql="SELECT DISTINCT 支出.大分類 + chr(9) + 支出.中分類 + chr(9) + 支出.小分類 + chr(9) + 支出.項目 as 対象項目 "+
        " FROM 支出 "+
        " WHERE ((支出.小分類 Like '%' + $srchText + '%') OR ((支出.項目 Like '%' + $srchText + '%'))); "
        $ADOrcs.Open($sql, $ADOcon, $CursorType,$LockType); #, $CursorType,$LockTypeを省くとRecordCountが-1.
        while( $ADOrcs.EOF -eq $false ) {
            Write-Host $ADOrcs.Fields.Item("対象項目").Value;
            $ADOrcs.MoveNext();
        }
        $ADOrcs.Close()
    }
    Write-Host -ForegroundColor green -NoNewline "検索文字列(Enter:終了)="
    $srchText = Read-Host
}
$ADOcon.Close()
$ADOrcs = $null
$ADOcon = $null

```

```

検索文字列(Enter:終了)=ボカリ*1.5
検索文字列(Enter:終了)=ボカリ*1.5
食費 飲料 スポーツ飲料 ボカリスエット 1.5L
検索文字列(Enter:終了)=キャベツ
食費 加工食品 おでん FFおでん ロールキャベツ
食費 野菜類 カット野菜 キャベツ&レタス
食費 野菜類 カット野菜 キャベツ千切り
食費 野菜類 カット野菜 キャベツ野菜炒め
食費 野菜類 カット野菜 セレクト キャベツ野菜炒め
食費 野菜類 キャベツ キャベツ
食費 野菜類 キャベツ キャベツ 1/2
検索文字列(Enter:終了)=
PS G:¥CMD¥ps>
完了

```

1	No.	日付	大分類	中分類	小分類	項目	数量
17154	17153	2020/5/1(金)	食費	飲料	紅茶	デイリークラブ ティーバッグ	
17155	17154	2020/5/1(金)	食費	野菜類	ニラ		
17156	17155	2020/5/1(金)	食費	飲料	ミルク	森永 クリープ 袋 200g	
17157	17156	2020/5/1(金)	食費	野菜類	キュウリ	キュウリ	
17158	17157	2020/5/1(金)	食費	野菜類	もやし	もやし	
17159	17158	2020/5/1(金)	食費	野菜類	キノコ	舞茸	
17160	17159	2020/5/1(金)	食費	税金等	消費税	消費税	
17161	17160	2020/5/1(金)	食費	税金等	消費税	キャッシュレス還元額	
17162			食費	野菜類	キャベツ	キャベツ 1/2	
17163							

③

モジュール設計仕様書	システム	家計簿.xlsx	家計簿入力補助・履歴検索	作成日	2020/5/3
補足	モジュール	PowerShell - ADO - xlsx	家計簿支出検索.ps1 (家計簿入力補助) (ADO-xlsx)	作成者	hosiyamakaze

DFD:④-2 家計簿支出検索(家計簿入力補助) ADO-xlsx PowerShell 実装

Windows PowerShell ISE (x86)

テンプレート - work.ps1 X

```

93
94 # ----- 家計簿支出検索(家計簿入力補助)(Excelから検索(ADO))
95 #※PowerShellISE(x86):32ビットで実行すること、ワイルドカードは%で(*はダメ)。
96 $xlsDB = "N:/Excel/家計簿.xlsx";
97 $CursorType=3; #adOpenStatic(3)/adOpenKeyset(2)
98 $LockType=1; #adLockOptimistic(3)/adLockReadOnly(1)
99 $srchText = "__initial_Connection__"
100 while ($srchText.Length -gt 0){
101     if($srchText -eq "__initial_Connection__"){
102         $ADOcon = New-Object -ComObject ADODB.Connection;
103         $connectionString = "Provider = Microsoft.ACE.OLEDB.12.0; Data Source = " + $xlsDB +
104             "; Extended Properties=Excel 12.0;"
105         $ADOcon.Open($connectionString);
106         $ADOrcs = New-Object -ComObject ADODB.Recordset;
107     }else{
108         $sql ="SELECT DISTINCT 支出.大分類 + chr(9) + 支出.中分類 + chr(9)
109             * FROM 支出 " +
110             " WHERE ((支出.小分類 Like '%" + $srchText + "%')) OR ((支出.項目 Like '%" + $srchText + "%'))";
111         $ADOrcs.Open($sql, $ADOcon, $CursorType,$LockType); #, $CursorType,$LockTypeを省くとRecordCountが-1.
112         while( $ADOrcs.EOF -eq $false ) {
113             Write-Host $ADOrcs.Fields.Item("対象項目").Value;
114             $ADOrcs.MoveNext();
115         }
116         $ADOrcs.Close()
117     }
118     Write-Host -ForegroundColor green -NoNewline "検索文字列(Enter:終了)=" $srchText
119     $srchText = Read-Host
120 }
121 $ADOcon.Close()
122 $ADOrcs = $null
123 $ADOcon = $null
124
125 検索文字列(Enter:終了)=キャベツ
126 食費 加工食品 おでん FFおでん ロールキャベツ
127 食費 野菜類 カット野菜 キャベツ&レタス
128 食費 野菜類 カット野菜 キャベツ千切り
129 食費 野菜類 カット野菜 キャベツ野菜炒め
130 食費 野菜類 カット野菜 セレクト キャベツ野菜炒め
131 食費 野菜類 キャベツ キャベツ
132 食費 野菜類 キャベツ キャベツ 1/2
133
134 検索文字列(Enter:終了)=ポカリ*1.5
135 食費 飲料 スポーツ飲料 ポカリスエット 1.5L
136
137 検索文字列(Enter:終了)=ポカリ*1.5
138 検索文字列(Enter:終了)=
139
140 PS N:>CMD>

```

家計簿.xlsxの範囲「支出」を直接検索する

- ① 家計簿支出検索スクリプトを選択する
- ② 「選択項目を実行」し、検索文字列を入力する。
- ③ 該当項目をコピー&ペーストでExcelに貼り付ける
tabが効いているのでセルに区切られて収まる。
※ワイルドカードは%で(*はダメ)。
(「支出」範囲を時々更新するとよい)

家計簿.xlsxが開いていると、
検索できないケースが
発生する。
排他制御の関係かな。

```

# ----- 家計簿支出検索(家計簿入力補助)(Excelから検索(ADO))
#※PowerShellISE(x86):32ビットで実行すること、ワイルドカードは%で(*はダメ)。
$xmlsDB = "N:/Excel/家計簿.xlsx";
$CursorType=3; #adOpenStatic(3)/adOpenKeyset(2)
$LockType=1; #adLockOptimistic(3)/adLockReadOnly(1)
$srchText = "__initial_Connection__"
while ($srchText.Length -gt 0){
    if($srchText -eq "__initial_Connection__"){
        $ADOcon = New-Object -ComObject ADODB.Connection;
        $connectionString = "Provider = Microsoft.ACE.OLEDB.12.0; Data Source = " + $xlsDB +
            "; Extended Properties=Excel 12.0;"
        $ADOcon.Open($connectionString);
        $ADOrcs = New-Object -ComObject ADODB.Recordset;
    }else{
        $sql ="SELECT DISTINCT 支出.大分類 + chr(9) + 支出.中分類 + chr(9) + 支出.小分類 + chr(9) + 支出.項目 as 対象項目 "+
            " FROM 支出 " +
            " WHERE ((支出.小分類 Like '%" + $srchText + "%')) OR ((支出.項目 Like '%" + $srchText + "%'))";
        $ADOrcs.Open($sql, $ADOcon, $CursorType,$LockType); #, $CursorType,$LockTypeを省くとRecordCountが-1.
        while( $ADOrcs.EOF -eq $false ) {
            Write-Host $ADOrcs.Fields.Item("対象項目").Value;
            $ADOrcs.MoveNext();
        }
        $ADOrcs.Close()
    }
    Write-Host -ForegroundColor green -NoNewline "検索文字列(Enter:終了)=" $srchText
    $srchText = Read-Host
}
$ADOcon.Close()
$ADOrcs = $null
$ADOcon = $null

```

1	No.	日付	大分類	中分類	小分類	項目	数量
17154	17153	2020/5/1(金)	食費	飲料	紅茶	デイリークラブ ティーバッグ	
17155	17154	2020/5/1(金)	食費	野菜類	ニラ	ニラ	
17156	17155	2020/5/1(金)	食費	飲料	ミルク	森永 クリープ 袋 200g	
17157	17156	2020/5/1(金)	食費	野菜類	キュウリ	キュウリ	
17158	17157	2020/5/1(金)	食費	野菜類	もやし	もやし	
17159	17158	2020/5/1(金)	食費	野菜類	キノコ	舞茸	
17160	17159	2020/5/1(金)	食費	税金等	消費税	消費税	
17161	17160	2020/5/1(金)	食費	税金等	消費税	キャッシュレス還元額	
17162	17161	2020/5/1(金)	食費	野菜類	キャベツ	キャベツ 1/2	

モジュール設計仕様書	システム	家計簿.DB3 SQLite3	家計簿入力補助・履歴検索	作成日	2020/5/3
補足	モジュール	Visual Studio Code - SQLite3 - CSV	家計簿支出検索(家計簿入力補助)(SQLite-CSV)	作成者	hosiyamakaze

DFD: ⑤ 家計簿支出検索(家計簿入力補助) SQLite3 - CSV/Visual Studio Code 実装

Visual Studio Code interface showing the SQLite Explorer and a terminal window. The SQLite Explorer displays a table named '支出' with the following columns: #, 大分類, 中分類, 小分類, 項目. The terminal window shows the execution of a SQL query: 'sqlite> SELECT DISTINCT 大分類, 中分類, 小分類, 項目 FROM 支出 WHERE (項目 Like '%キャベツ%');'.

Context menu options for the SQLite Explorer table:

- すべての出現箇所を変更 (Ctrl+F2)
- 切り取り (Ctrl+X)
- コピー (Ctrl+C)
- 貼り付け (Ctrl+V)
- Run Query (Ctrl+Shift+Q)
- Run Selected Query
- Use Database
- コマンドパレット... (Ctrl+Shift+P)

1. ITEM.CSV(UFT8,',';)をデータベースにインポートする
cmdターミナルでSQLite3を実行し、.open ~ .importを「選択したテキストの実行」する。(Drop table 支出;で支出テーブルをクリアできる)
2. 「Use Database」で拡張機能(SQLite 0.8.2)からデータベース接続し、「Run Selected Query」を実行すれば、テーブル形式で結果表示される。このテーブルからコピーすればtab区切りで取得できる。
3. SQL(SELECT文)を「選択したテキストの実行」した場合は、ターミナルに';区切りで表示される。
4. DB Browser for SQLiteの「File⇒Import⇒Table from CSV」を使えば家計簿_UTF8.CSV(UFT8,tab)を取り込みできる。「Execute SQL」タブを使ってSQL検索しテーブル形式で結果を得る。コピーすればtab区切りで取得できる。
({家計簿_UTF8,ITEM}.csvをとときどき最新化するとよい。②-2参照)

DB Browser for SQLite - G:\CMD\SQLite\家計簿.db3

Execute SQL

```
SQL 4
1 SELECT DISTINCT 大分類, 中分類, 小分類, 項目
2 FROM 支出
3 WHERE (項目 Like '%キャベツ%')
4 ;
5
```

#	大分類	中分類	小分類	項目
3	食費	野菜類	カット野菜	キャベツ千切り
4	食費	加工食品	おでん	FFおでん ロールキャベツ
5	食費	野菜類	カット野菜	セレクト キャベツ野菜炒め
6	食費	野菜類	カット野菜	キャベツ&レタス
7	食費	野菜類	キャベツ	キャベツ 1/2

モジュール設計仕様書	システム	家計簿.DB3 SQLite3	家計簿入力補助・履歴検索	作成日	2020/5/14
補足	モジュール	Visual Studio C# - SQLite3 - console	家計簿支出検索(家計簿入力補助)(C#-SQLite-コンソール)	作成者	hosiyamakaze

DFD:⑥-1 家計簿支出検索(家計簿入力補助) Visual Studio 2017 C#-SQLite-コンソールアプリ実装

参照のコンテキストメニューからNuGetパッケージの管理を開いて、System.Data.SQLiteをインストールする

System.Data.SQLite 作成者: SQLite Development Team, 4.72M 件のダウンロード v1.0.112.2
The official SQLite database engine for both x86 and x64 along with the ADO.NET provider. This package includes support for LINQ and Entity Framework 6.

```

using System;
using System.Data.SQLite;
namespace SrchTextCSConDB3
{
    class Program
    {
        static void Main(string[] args)
        {
            string srchText;
            string strQuery;
            SQLiteConnection DB3con = new SQLiteConnection("Data Source=G:/CMD/SQLite/家計簿.db3");
            DB3con.Open();
            while (true)
            {
                Console.WriteLine("検索文字列(Enter:終了)=");
                srchText = Console.ReadLine();
                if (srchText.Length == 0) break;
                strQuery = "SELECT DISTINCT 大分類,中分類,小分類,
                FROM 支出" +
                " WHERE ((小分類 Like '%" + srchText + "%')
                SQLiteCommand DB3cmd = new SQLiteCommand(strQuery, DB3con);
                SQLiteDataReader DB3rcs = DB3cmd.ExecuteReader();
                if (DB3rcs.HasRows)
                {
                    while (DB3rcs.Read())
                    {
                        Console.WriteLine("{0}\t{1}\t{2}\t{3}", DB3rcs["大分類"], DB3rcs["中分類"], DB3rcs["小分類"], DB3rcs["項目"]);
                    }
                }
                if (DB3con.State != System.Data.ConnectionState.Closed)
                {
                    DB3con.Close();
                }
            }
        }
    }
}

```

```

using System;
using System.Data.SQLite;
namespace SrchTextCSConDB3{
class Program{
static void Main(string[] args){
string srchText;
string strQuery;
SQLiteConnection DB3con = new SQLiteConnection("Data Source=G:/CMD/SQLite/家計簿.db3");
DB3con.Open();
while (true){
Console.WriteLine("検索文字列(Enter:終了)=");
srchText = Console.ReadLine();
if (srchText.Length == 0) break;
strQuery = "SELECT DISTINCT 大分類,中分類,小分類,項目" +
" FROM 支出" +
" WHERE ((小分類 Like '%" + srchText + "%') OR (項目 Like '%" + srchText + "%'))";
SQLiteCommand DB3cmd = new SQLiteCommand(strQuery, DB3con);
SQLiteDataReader DB3rcs = DB3cmd.ExecuteReader();
if (DB3rcs.HasRows){
while (DB3rcs.Read()){
Console.WriteLine("{0}\t{1}\t{2}\t{3}", DB3rcs["大分類"], DB3rcs["中分類"], DB3rcs["小分類"], DB3rcs["項目"]);
}
}
if (DB3con.State != System.Data.ConnectionState.Closed){
DB3con.Close();
}
}
}
}

```

※改行位置変更

```

検索文字列(Enter:終了)=キャベツ
食費 野菜類 キャベツ キャベツ
食費 野菜類 カット野菜 キャベツ野菜炒め
食費 野菜類 カット野菜 キャベツ千切り
食費 加工食品 おでん FFおでん ロールキャベツ
食費 野菜類 カット野菜 セレクト キャベツ野菜炒め
食費 野菜類 カット野菜 キャベツ&シタス
食費 野菜類 キャベツ キャベツ 1/2
検索文字列(Enter:終了)=ポカリ%1.5
食費 飲料 スポーツ飲料 ポカリスエット 1.5L
検索文字列(Enter:終了)=

```

モジュール設計仕様書	システム	家計簿.DB3 SQLite3	家計簿入力補助・履歴検索	作成日	2025/1/22
補足	モジュール	Visual Studio C# - SQLite3 - Form	家計簿支出検索(家計簿入力補助)(C#-SQLite-フォーム)	作成者	hosiyamakaze

DFD:⑥-2 家計簿支出検索(家計簿入力補助) Visual Studio 2017 C#-SQLite-フォームアプリ実装

```

using System;
using System.Data;
using System.Windows.Forms;
using System.Data.SQLite;

namespace SrchTextCSWinDB3
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            this.Text = "家計簿支出検索(家計簿.db3)";
        }

        private SQLiteConnection DB3con;
        private SQLiteCommand DB3cmd;
        private SQLiteDataAdapter DB3da;
        private DataSet DS = new DataSet();
        private DataTable DT = new DataTable();

        // loadData
        private void LoadData(string srchText)
        {
            DB3con = new SQLiteConnection("Data Source = G:/CMD/SQLite/家計簿.db3");
            DB3con.Open();
            DB3cmd = DB3con.CreateCommand();
            string strQuery = "SELECT DISTINCT 大分類,中分類,小分類,項目" +
                " FROM 支出" +
                " WHERE ((小分類 Like '%" + srchText + "%') OR (項目 Like '%" + srchText + "%'))" +
                " order by 大分類,中分類,小分類,項目";
            DB3da = new SQLiteDataAdapter(strQuery, DB3con);
        }
    }
}

```

参照のコンテキストメニューからNuGet/パッケージの管理を開いて、System.Data.SQLiteをインストールする

家計簿支出検索(家計簿.db3)

大分類	中分類	小分類	項目
食費	加工食品	おでん	FFおでん ロールキャベツ
食費	野菜類	カット野菜	キャベツ&レタス
食費	野菜類	カット野菜	キャベツ千切り
食費	野菜類	カット野菜	キャベツ野菜炒め
食費	野菜類	カット野菜	セレクト キャベツ野菜炒め
食費	野菜類	キャベツ	キャベツ
食費	野菜類	キャベツ	キャベツ 1/2

検索文字列

- Ctrl+Cでtab区切りでコピーを取得できる
- 検索文字列のTextChangedイベントでグリッドを更新するので、検索ボタンを押さなくても可。

```

using System;
using System.Data;
using System.Windows.Forms;
using System.Data.SQLite;

namespace SrchTextCSWinDB3{
    public partial class Form1 : Form{
        public Form1(){
            InitializeComponent();
            this.Text = "家計簿支出検索(家計簿.db3)";
        }

        private SQLiteConnection DB3con;
        private SQLiteCommand DB3cmd;
        private SQLiteDataAdapter DB3da;
        private DataSet DS = new DataSet();
        private DataTable DT = new DataTable();

        // loadData
        private void LoadData(string srchText){
            DB3con = new SQLiteConnection("Data Source = G:/CMD/SQLite/家計簿.db3");
            DB3con.Open();
            DB3cmd = DB3con.CreateCommand();
            string strQuery = "SELECT DISTINCT 大分類,中分類,小分類,項目" +
                " FROM 支出" +
                " WHERE ((小分類 Like '%" + srchText + "%') OR (項目 Like '%" + srchText + "%'))" +
                " order by 大分類,中分類,小分類,項目";
            DB3da = new SQLiteDataAdapter(strQuery, DB3con);
            DS.Reset();
            DB3da.Fill(DS);
            DT = DS.Tables[0];
            dataGridView1.DataSource = DT;
            dataGridView1.Columns[dataGridView1.Columns.Count - 1].AutoSizeMode =
                DataGridViewAutoSizeColumnMode.Fill;
            DB3con.Close();
        }

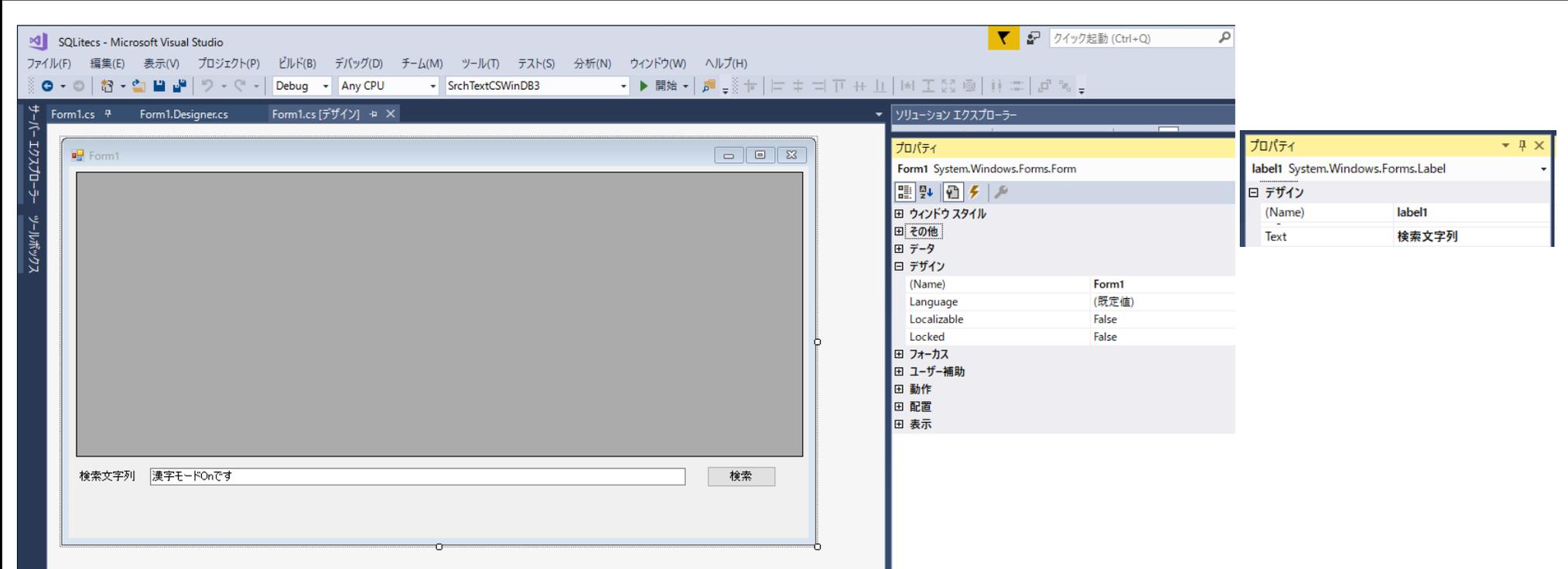
        private void SrchBtn_Click(object sender, EventArgs e) => LoadData(SrchTextTB.Text);
        private void SrchTextTB_TextChanged(object sender, EventArgs e) => LoadData(SrchTextTB.Text);
    }
}

```

※改行位置変更

モジュール設計仕様書	システム	家計簿.DB3 SQLite3	家計簿入力補助・履歴検索	作成日	2025/1/22
補足	モジュール	Visual Studio C# - SQLite3 - Form	家計簿支出検索(家計簿入力補助)(C#-SQLite-フォーム)	作成者	hosiyamakaze

DFD:⑥-2 家計簿支出検索(家計簿入力補助) Visual Studio 2017 C#-SQLite-フォームアプリ実装 FORMの定義



プロパティ

dataGridView1 System.Windows.Forms.DataGridView

デザイン	
(Name)	dataGridView1
動作	
AllowUserToAddRows	False
AllowUserToDeleteRows	False
ClipboardCopyMode	EnableWithoutHeaderText
ColumnHeadersHeightSizeMode	AutoSize
MultiSelect	False
ReadOnly	True
SelectionMode	FullRowSelect

プロパティ

SrchTextTB System.Windows.Forms.TextBox

(Name)	SrchTextTB
動作	
ImeMode	On
表示	
Text	漢字モードOnです

プロパティ

SrchBtn System.Windows.Forms.Button

(Name)	SrchBtn
表示	
Text	検索

※Anchorを設定して、ウインドのサイズ変更に追従させる

モジュール設計仕様書	システム	家計簿.csv	家計簿入力補助・履歴検索	作成日	2020/5/18
補足	モジュール	Visual Studio C++ - CSV - Console	家計簿支出検索(家計簿入力補助)(C++-CSV-コンソール)	作成者	hosiyamakaze

DFD:⑦-1 家計簿支出検索(家計簿入力補助) Visual Studio 2017 C++-CSV コンソールアプリ実装(map, vector, list版)

C++標準ライブラリのmap,vector,listを使ってみた。
※検索文字列によっては異常。「ポ」「ン」とか...

アルゴリズムでソートしようするとコンパイルエラーが発生する。なぜ?
#include <algorithm>
bool strOrder(char *elem1, char *elem2) //文字列の昇順(true)を判定する
{
return strcmp(elem1,elem2) <= 0;
}
sort(DataList.begin(), DataList.end(),strOrder);

```

117 家計簿検索 読み込み...
118 家計簿検索 読み込み... 17433 件(データ項目件数: 5937 件)
119 検索文字列(Enter:終了)=カリス
120 食費 飲料 スポーツ飲料 ポカリスウェット
121 食費 飲料 スポーツ飲料 ポカリスウェット イオンウォーター 500ml
122 食費 飲料 スポーツ飲料 ポカリスウェット 500ml
123 食費 飲料 スポーツ飲料 ポカリスウェット 500ml
124 食費 飲料 スポーツ飲料 ポカリスウェット 900ml
125 食費 飲料 スポーツ飲料 ポカリスウェット 1.5L
126 食費 飲料 スポーツ飲料 ポカリスウェット イオンウォーター 900ml
127 食費 飲料 スポーツ飲料 大塚製薬 ポカリスウェット イオンウォーター 900ml×12本
128 食費 飲料 スポーツ飲料 大塚製薬 ポカリスウェット 1500ml×8本
129 検索文字列(Enter:終了)=キャベツ
130 食費 野菜類 キャベツ
131 食費 野菜類 カット野菜 キャベツ野菜炒め
132 食費 野菜類 カット野菜 キャベツ千切り
133 食費 加工食品 おでん FFおでん ロールキャベツ
134 食費 野菜類 カット野菜 セレクト キャベツ野菜炒め
135 食費 野菜類 カット野菜 キャベツ&シラス
136 食費 野菜類 キャベツ
137 食費 野菜類 キャベツ 1/2
138 検索文字列(Enter:終了)=

```

名前	コード	説明	プロジェクト	ファイル	行	抑制状態
	C2676	二項演算子 '!' : 'const std::list_unchecked_iterator<std::list_val<std::list_simple_types<_Ty>>>' は、この演算子または定義済の演算子に適切な型への変換の定義を行いません。(新しい動作: ヘルプを参照)	srchTextCPPConCSV	algorithm	4034	
	C2672	'_Sort_unchecked': 一致するオーバーロードされた関数が見つかりませんでした。	srchTextCPPConCSV	algorithm	4034	
	C2780	'void std::_Sort_unchecked(_RanIt, _RanIt, iterator_traits<_Iter>::difference_type, _Pr)': 4 引数が必要ですが -3 が設定されます。	srchTextCPPConCSV	algorithm	4034	

モジュール設計仕様書	システム	家計簿.csv	家計簿入力補助・履歴検索	作成日	2020/5/18
補足	モジュール	Visual Studio C++ - CSV - Console	家計簿支出検索(家計簿入力補助)(C++-CSV-コンソール)	作成者	hosiyamakaze

DFD:⑦-1 家計簿支出検索(家計簿入力補助) Visual Studio 2017 C++-CSV コンソールアプリ実装(CSVファイルを読み込む概念図)(map, vector, list版)

家計簿.csv

No.	日付	口座	大分類	中分類	小分類	項目	数量	金額	取引先	備考	周期
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
86	2014/2/3(月)	SUICA	食費	外食	食事	トーストセット	1	399	ジョナサン	NULL	NULL
87	2014/2/3(月)	SUICA	食費	外食	食事	ヨーグルト	1	105	ジョナサン	NULL	NULL

```

63 //項目見出し(名称、位置)をMapに取得する
64 ItemIT_t next(strbuf, strbuf + strlen(strbuf), rxDel);
65 for (next = ItemIT_t(strbuf, strbuf + strlen(strbuf), rxDel, -1), i = 0; next != end; ++next, i++) {
66     ItemMap.insert(ItemMap_t::value_type(_strdup((char *)&next->str()[0]), i));
67 }

```

```

18 typedef std::map<char *, int> ItemMap_t;
19 ItemMap_t ItemMap;

```

ItemMap

キー	値
[comparator]	less
[allocator]	allocator
[0x000001be9fdaeca0 "No. "]	0
[0x000001be9fdaf990 "小分類"]	5
[0x000001be9fdafb70 "大分類"]	3
[0x000001be9fdafb0 "項目"]	6
[0x000001be9fdb0200 "数量"]	7
[0x000001be9fdb0250 "備考"]	10
[0x000001be9fdb02a0 "周期"]	11
[0x000001be9fdb0520 "金額"]	8
[0x000001be9fdb06b0 "口座"]	2
[0x000001be9fdb0700 "中分類"]	4
[0x000001be9fdb07f0 "日付"]	1
[0x000001be9fdb08e0 "取引先"]	9
[未加工ビュ]	{...}

```

72 //データ項目をVectorに取得する
73 ItemIT_t next(strbuf, strbuf + strlen(strbuf), rxDel);
74 for (next = ItemIT_t(strbuf, strbuf + strlen(strbuf), rxDel, -1), i = 0; next != end; ++next, i++) {
75     rowVec.push_back(_strdup((char *)&next->str()[0]));
76 }
77 for (; i < ItemMap.size(); i++) {
78     rowVec.push_back(_strdup("")); //最後の項目値がNULLの分
79 }

```

rowVec

インデックス	値
[0]	0x000001be9fdb3b40 "86"
[1]	0x000001be9fdb0430 "2014/2/3(月)"
[2]	0x000001be9fdb0570 "SUICA"
[3]	0x000001be9fdb05c0 "食費"
[4]	0x000001be9fdb0610 "外食"
[5]	0x000001be9fdafa80 "食事"
[6]	0x000001be9fdafa0 "トーストセット"
[7]	0x000001be9fdb3c80 "1"
[8]	0x000001be9fdb0660 "399"
[9]	0x000001be9fdb0750 "ジョナサン"
[10]	0x000001be9fdb3740 ""
[11]	0x000001be9fdb3540 ""
[未加工ビュ]	{...}

```

13 const char *strDaiBnr = "大分類";
14 const char *strChuBnr = "中分類";
15 const char *strShoBnr = "小分類";
16 const char *strKmk = "項目";

```

```

83 //データ項目を出力対象項目(大分類、中分類、小分類、項目をtab区切りでつなぐ)のみに絞る
84 sprintf_s(strbuf, "%s\t%s\t%s\t%s", strValue(strDaiBnr), strValue(strChuBnr), strValue(strShoBnr), strValue(strKmk));
85 if(isNewValue(strbuf)) DataList.push_back(_strdup(strbuf));

```

DataList

インデックス	値
[33]	0x000001be9fdb6010 "食費\t外食\t食事\tトーストセット"
[34]	0x000001be9fdb6c10 "食費\t外食\t食事\tヨーグルト"

モジュール設計仕様書	システム	家計簿.csv	家計簿入力補助・履歴検索	作成日	2020/5/18
補足	モジュール	Visual Studio C++ - CSV - Console	家計簿支出検索(家計簿入力補助)(C++-CSV-コンソール)	作成者	hosiyamakaze

DFD:⑦-1 家計簿支出検索(家計簿入力補助) Visual Studio 2017 C++-CSV コンソールアプリ実装(map, vector, list版)

```

1 //UTF8 $Workfile: srchTextCPPConCSV.cpp $$Revision: 4 $$Date: 20/05/18 19:30 $
2 //$NoKeywords: $
3
4 #include <iostream>
5 #include <fstream>
6 #include <regex>
7 #include <map>
8 #include <vector>
9 #include <list>
10
11 using namespace std;
12
13 const char *strDaiBnr = "大分類"; //出力(使用)する項目の見出し
14 const char *strChuBnr = "中分類";
15 const char *strShoBnr = "小分類";
16 const char *strKmk = "項目";
17
18 typedef std::map<char *, int> ItemMap_t; //(見出し,出現順位)を保管する(CSVの全ての見出し)
19 ItemMap_t ItemMap;
20 ItemMap_t::iterator ItemMapIT;
21
22 vector<char *> rowVec; //データ項目を出現位置順に保管する(行ごとに一時使用)
23 vector<char *>::iterator rowIT;
24
25 list<char *> DataList; //出力および検索に使用する項目をユニークにして保持する
26 list<char *>::iterator DataListIT;
27
28 char *strValue(const char *itemNm); //rowVec,ItemMapから見出しに対応するデータ値を取り出す
29 bool isNewValue(const char *dataStr); //DataList上でデータ値(出力する大分類~項目)をユニークにするための判定
30
31
32 int main()
33 {
34     const char *csvFlNm = "G:/Projects/srchTextCPPCSV/x64/Debug/家計簿.csv";
35     ifstream* ifCSV = new ifstream;
36     int csvRecCnt = 0;
37
38     typedef regex_token_iterator<const char *> ItemIT_t; //CSVファイルの行ごとにtabで分割する
39     ItemIT_t::regex_type rxDel("¥t");
40     ItemIT_t end;
41     cmatch narrowMatch;
42
43     char srchText[128]; //検索文字列
44     char strbuf[1024]; //汎用文字列
45     int i; //汎用インデックス
46
47     /* ----- procedure */
48
49     //CSVファイルを読み込む
50     cout << "家計簿検索 読み込み...¥n";
51     ifCSV->open(csvFlNm);
52     if (ifCSV->fail()) {
53         cout << "ERROR: Cannot open file :'" << csvFlNm << endl;
54         goto exit_procedure;
55     }
56 ...以下略

```

srchTextCPPConCSV_mapNvector.cpp
Revision: 4 までMap,Vector版

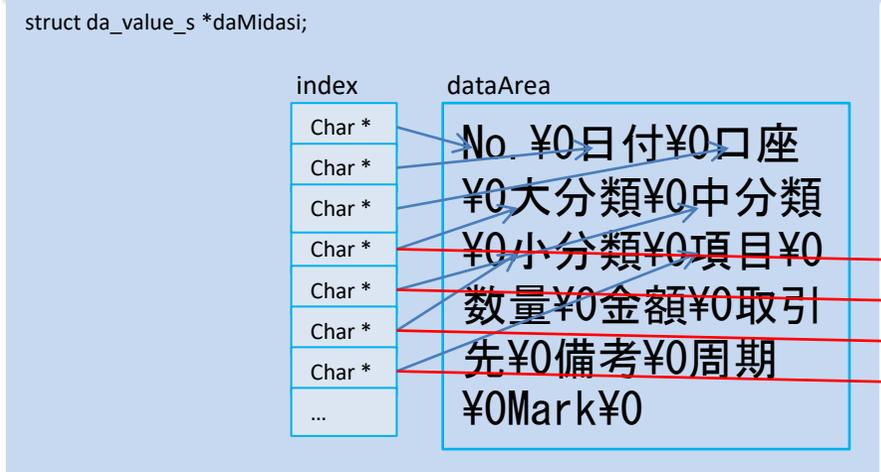
モジュール設計仕様書	システム	家計簿.csv	家計簿入力補助・履歴検索	作成日	2020/5/18
補足	モジュール	Visual Studio C++ - CSV - Console	家計簿支出検索(家計簿入力補助)(C++-CSV-コンソール)	作成者	hosiyamakaze

DFD:⑦-1' 家計簿支出検索(家計簿入力補助) Visual Studio C++-CSV コンソールアプリ実装(CSVファイルを読み込む概念図)(davalue, gdlist版)

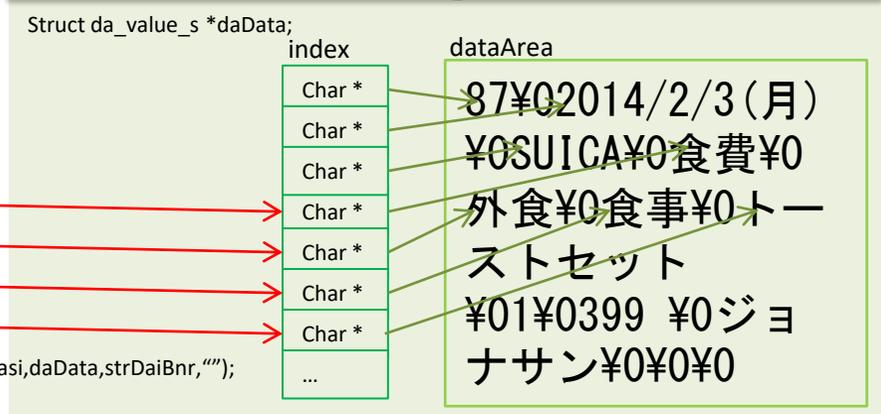
家計簿.csv

No.	日付	口座	大分類	中分類	小分類	項目	数量	金額	取引先	備考	周期
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
86	2014/2/3(月)	SUICA	食費	外食	食事	トーストセット	1	399	ジョナサン	NULL	NULL
87	2014/2/3(月)	SUICA	食費	外食	食事	ヨーグルト	1	105	ジョナサン	NULL	NULL

```
//項目見出し(名称、位置)をdavalueに取得する
ItemIT_t next(strbuf, strbuf + strlen(strbuf), rxDel);
for(next = ItemIT_t(strbuf, strbuf + strlen(strbuf), rxDel, -1); next != end; ++next) {
    da_put(daMidasi, (char *)&next->str()[0]);
}
```

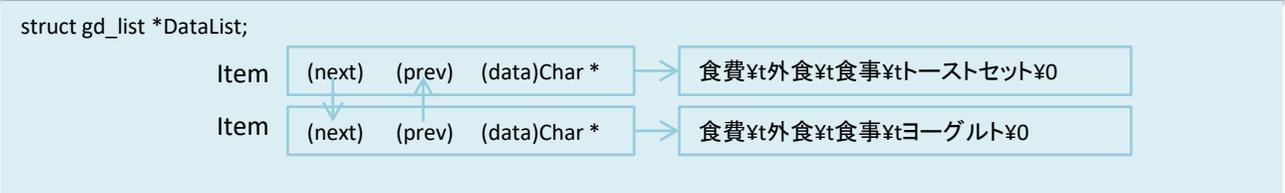


```
//データ項目をdavalueに取得する
da_reset(daData);
ItemIT_t next(strbuf, strbuf + strlen(strbuf), rxDel);
for(next = ItemIT_t(strbuf, strbuf + strlen(strbuf), rxDel, -1); next != end; ++next) {
    da_put(daData, (char *)&next->str()[0]);
}
da_fin(daData);
```



```
13 const char *strDaiBnr = "大分類";
14 const char *strChuBnr = "中分類";
15 const char *strShoBnr = "小分類";
16 const char *strKmk = "項目";
```

```
//データ項目を出力対象項目(大分類、中分類、小分類、項目をtab区切りでつなぐ)のみに絞る
sprintf_s(strbuf, "%s¥t%s¥t%s¥t%s",
    da_getVN(daMidasi, daData, strDaiBnr, ""), da_getVN(daMidasi, daData, strChuBnr, ""),
    da_getVN(daMidasi, daData, strShoBnr, ""), da_getVN(daMidasi, daData, strKmk, ""));
GD_FINDF(findItem, DataList, GD_FIRST(DataList), (strcmp(strbuf, (const char *)item->data) == 0));
if (findItem == NULL) gd_insert(DataList, NULL, (unsigned char *)strbuf, strlen(strbuf) + 1);
```



モジュール設計仕様書	システム	家計簿.csv	家計簿入力補助・履歴検索	作成日	2025/1/22
補足	モジュール	Visual Studio C++ - CSV - Console	家計簿支出検索(家計簿入力補助)(C++-CSV-コンソール)	作成者	hosiyamakaze

DFD:⑦-1' 家計簿支出検索(家計簿入力補助) Visual Studio C++-CSV コンソールアプリ実装(davalue, gdlist版)

```

1 static char sccsid[] =                                     42
2 "@(#)$Workfile: srchTextCPPConCSV.cpp $$Revision: 8 $$Date: 20/06/06 15:11 $$NoKeywords: $"; 43 cmatch narrowMatch;
3 /*****                                                                                               44
4 *                                                                                                   45 unsigned char srchText[128]; //検索文字列
5 *   srchTextCPPConCSV:CSVファイルを取り込み、正規表現で検索する                               46 char strbuf[1024]; //汎用文字列
6 *   (davalue,gdlistをC++化した)                                                                 47
7 *                                                                                                   48 /* ---- procedure */
8 *****/                                                                                               49
9                                                                                                       50 daMidasi = da_init(1024);
10 #include <iostream>                                                                                   51 daData = da_init(1024);
11 #include <fstream>                                                                                   52 DataList = gd_init(U_DUP);
12 #include <regex>                                                                                     53
13 #include "davalue.h"                                                                               54 //CSVファイルを読み込む
14 #include "gdlist.h"                                                                               55 cout << "家計簿検索 読み込み[" << csvFlNm << "]"... ";
15                                                                                                       56 ifstream ifCSV->open(csvFlNm);
16 using namespace std;                                                                               57 if (ifCSV->fail()) {
17                                                                                                       58   cout << "ERROR: Cannot open file :'" << csvFlNm << endl;
18 void regChrEsc(unsigned char *str);                                                                 59   goto exit_procedure;
19                                                                                                       60 }
20                                                                                                       61...以下略
21 int main()
22 {
23   const char *strDaiBnr = "大分類";//出力(使用)する項目の見出し
24   const char *strChuBnr = "中分類";
25   const char *strShoBnr = "小分類";
26   const char *strKmk = "項目";
27
28   const char *csvFlNm = "N:/CMD/家計簿/家計簿.csv";
29   ifstream* ifCSV = new ifstream;
30   int csvRecCnt = 0;
31
32   struct da_value_s *daMidasi; //見出しを出現位置順に保管する(CSVの全ての見出し)
33   struct da_value_s *daData; //データ項目を出現位置順に保管する(行ごとに一時使用)
34
35   struct gd_list *DataList; //出力および検索に使用するデータ項目をユニークにして保持する
36   /* mode:DUP,key:NULL,data:str(大分類、中分類、小分類、項目をtab区切りでつなぐ) */
37   struct gd_list *findItem; //データ項目
38
39   typedef regex_token_iterator<const char*> ItemIT_t; //CSVファイルの行ごとにtabで分割する
40   ItemIT_t::regex_type rxDel(".*t");
41   ItemIT_t end;

```

srchTextCPPConCSV_davalueNgdlist.cpp
Revision: 5 よりdavalue,gdlist版