

```

unit StateSchoolSampData;

interface

uses SysUtils, LCRGlobals, VCL.FlexCel.Core, FlexCel.XlsAdapter;

type
  TSchoolSampDataRec = record
    numb_second_schools_pub: double;
    numb_elem_schools_pub: double;
    numb_second_schools_priv: double;
    numb_elem_schools_priv: double;
    numb_daycares: double;
    pp_grandfather_mand_child: double;
    pp_grandfather_opt_child: double;
    pp_grandfather_mand_pub_elem: double;
    pp_grandfather_opt_pub_elem: double;
    pp_grandfather_mand_priv_elem: double;
    pp_grandfather_opt_priv_elem: double;
    pp_grandfather_opt1_pub_second: double;
    pp_grandfather_opt2_pub_second: double;
    pp_grandfather_opt1_priv_second: double;
    pp_grandfather_opt2_priv_second: double;
    stateabb: string;
  end;

  {
  State stabb
  numb_second_schools_pub
  numb_elem_schools_pub
  numb_second_schools_priv
  numb_elem_schools_priv
  numb_daycares
  pp_grandfather_mand_child
  pp_grandfather_opt_child
  pp_grandfather_mand_pub_elem
  pp_grandfather_opt_pub_elem
  pp_grandfather_mand_priv_elem
  pp_grandfather_opt_priv_elem
  pp_grandfather_opt1_pub_second
  pp_grandfather_opt2_pub_second
  pp_grandfather_opt1_priv_second
  pp_grandfather_opt2_priv_second
  }

  TStateSchoolSampData = class
  public
    option: string;
    numb_second_schools_pub: double;
    numb_elem_schools_pub: double;

```

```

numb_second_schools_priv: double;
numb_elem_schools_priv: double;
numb_daycares: double;
pp_grandfather_mand_child: double;
pp_grandfather_opt_child: double;
pp_grandfather_mand_pub_elem: double;
pp_grandfather_opt_pub_elem: double;
pp_grandfather_mand_priv_elem: double;
pp_grandfather_opt_priv_elem: double;
pp_grandfather_opt1_pub_second: double;
pp_grandfather_opt2_pub_second: double;
pp_grandfather_opt1_priv_second: double;
pp_grandfather_opt2_priv_second: double;
stateabb: string;

```

```

constructor Create; overload;
constructor Create(aOption: string); overload;
destructor Destroy; override;

```

```

procedure GetSystemData(stateabb: string);
procedure LoadSchoolSampData;
function GetStateSchoolSampData(abb: string): TSchoolSampDataRec;

```

```
private
```

```

Xls: TExcelFile;
SchoolSampData: array of TSchoolSampDataRec;

```

```

function getIndex(s: string): integer;
end;

```

implementation

```
{ TStateSchoolSampData }
```

```
constructor TStateSchoolSampData.Create;
```

```
var
```

```
filename: string;
```

```
begin
```

```
filename := DataPath + 'State_School_Samp_Data.xlsx';
```

```
Xls := TXlsFile.Create(Filename, False);
```

```
//Xls.ActiveSheetByName := 'LCRI';
```

```
Xls.ActiveSheetByName := option;
```

```
{
```

```
State_School_Samp_Data.xlsx
```

```
1 State char(25)
```

```
2 stabb char[2]
```

```
3 numb_second_schools_pub float
```

```
4 numb_elem_schools_pub float
```

```
5 numb_second_schools_priv float
```

```
6 numb_elem_schools_priv float
```

```

    7 numb_daycares float
    8 pp_grandfather_mand_child float
    9 pp_grandfather_opt_child float
10 pp_grandfather_mand_pub_elem float
11 pp_grandfather_opt_pub_elem float
12 pp_grandfather_mand_priv_elem float
13 pp_grandfather_opt_priv_elem float
14 pp_grandfather_opt1_pub_second float
15 pp_grandfather_opt2_pub_second float
16 pp_grandfather_opt1_priv_second float
17 pp_grandfather_opt2_priv_second float
}

SetLength(SchoolSampData, 68 + 1);
end;

constructor TStateSchoolSampData.Create(aOption: string);
var
    filename: string;
begin
    filename := DataPath + 'State_School_Samp_Data.xlsx';

    if aOption = 'LCR' then aOption := 'LCRR';

    Xls := TXlsFile.Create(Filename, False);
    Xls.ActiveSheetByName := aOption;
    {
        State_School_Samp_Data.xlsx
        1 State char(25)
        2 stabb char[2]
        3 numb_second_schools_pub float
        4 numb_elem_schools_pub float
        5 numb_second_schools_priv float
        6 numb_elem_schools_priv float
        7 numb_daycares float
        8 pp_grandfather_mand_child float
        9 pp_grandfather_opt_child float
        10 pp_grandfather_mand_pub_elem float
        11 pp_grandfather_opt_pub_elem float
        12 pp_grandfather_mand_priv_elem float
        13 pp_grandfather_opt_priv_elem float
        14 pp_grandfather_opt1_pub_second float
        15 pp_grandfather_opt2_pub_second float
        16 pp_grandfather_opt1_priv_second float
        17 pp_grandfather_opt2_priv_second float
    }

    SetLength(SchoolSampData, 68 + 1);
end;

```

```

destructor TStateSchoolSampData.Destroy;
begin
    FreeAndNil(Xls);

    inherited;
end;

function TStateSchoolSampData.getIndex(s: string): integer;
var
    r: integer;
begin
    if s = 'AL' then r := 1
    else if s = 'AK' then r := 2
    else if s = 'AZ' then r := 3
    else if s = 'AR' then r := 4
    else if s = 'CA' then r := 5
    else if s = 'CO' then r := 6
    else if s = 'CT' then r := 7
    else if s = 'DE' then r := 8
    else if s = 'DC' then r := 9
    else if s = 'FL' then r := 10
    else if s = 'GA' then r := 11
    else if s = 'HI' then r := 12
    else if s = 'ID' then r := 13
    else if s = 'IL' then r := 14
    else if s = 'IN' then r := 15
    else if s = 'IA' then r := 16
    else if s = 'KS' then r := 17
    else if s = 'KY' then r := 18
    else if s = 'LA' then r := 19
    else if s = 'ME' then r := 20
    else if s = 'MD' then r := 21
    else if s = 'MA' then r := 22
    else if s = 'MI' then r := 23
    else if s = 'MN' then r := 24
    else if s = 'MS' then r := 25
    else if s = 'MO' then r := 26
    else if s = 'MT' then r := 27
    else if s = 'NE' then r := 28
    else if s = 'NV' then r := 29
    else if s = 'NH' then r := 30
    else if s = 'NJ' then r := 31
    else if s = 'NM' then r := 32
    else if s = 'NY' then r := 33
    else if s = 'NC' then r := 34
    else if s = 'ND' then r := 35
    else if s = 'OH' then r := 36
    else if s = 'OK' then r := 37
    else if s = 'OR' then r := 38

```

```

else if s = 'PA' then r := 39
else if s = 'RI' then r := 40
else if s = 'SC' then r := 41
else if s = 'SD' then r := 42
else if s = 'TN' then r := 43
else if s = 'TX' then r := 44
else if s = 'UT' then r := 45
else if s = 'VT' then r := 46
else if s = 'VA' then r := 47
else if s = 'WA' then r := 48
else if s = 'WV' then r := 49
else if s = 'WI' then r := 50
else if s = 'WY' then r := 51
else if s = 'PR' then r := 52
else if s = 'GU' then r := 53
else if s = 'VI' then r := 54
else if s = 'AS' then r := 55
else if s = 'MP' then r := 56
else if s = '01' then r := 57
else if s = '02' then r := 58
else if s = '04' then r := 59
else if s = '05' then r := 60
else if s = '06' then r := 61
else if s = '07' then r := 62
else if s = '08' then r := 63
else if s = '09' then r := 64
else if s = '10' then r := 65
else if s = 'NN' then r := 66
else
    raise Exception.Create('Index not found in TStateSchoolSampData.getIndex for: '
+ s);

    Result := r;
end;

function TStateSchoolSampData.GetStateSchoolSampData(
    abb: string): TSchoolSampDataRec;
var
    idx: integer;
begin
    idx := getIndex(abb);
    Result.numb_second_schools_pub := SchoolSampData[idx].numb_second_schools_pub;
    Result.numb_elem_schools_pub := SchoolSampData[idx].numb_elem_schools_pub;
    Result.numb_second_schools_priv := SchoolSampData[idx].numb_second_schools_priv;
    Result.numb_elem_schools_priv := SchoolSampData[idx].numb_elem_schools_priv;
    Result.numb_daycares := SchoolSampData[idx].numb_daycares;
    Result.pp_grandfather_mand_child := SchoolSampData[idx].pp_grandfather_mand_child;
    Result.pp_grandfather_opt_child := SchoolSampData[idx].pp_grandfather_opt_child;
    Result.pp_grandfather_mand_pub_elem :=
SchoolSampData[idx].pp_grandfather_mand_pub_elem;

```

```

    Result.pp_grandfather_opt_pub_elem :=
SchoolSampData[idx].pp_grandfather_opt_pub_elem;
    Result.pp_grandfather_mand_priv_elem :=
SchoolSampData[idx].pp_grandfather_mand_priv_elem;
    Result.pp_grandfather_opt_priv_elem :=
SchoolSampData[idx].pp_grandfather_opt_priv_elem;
    Result.pp_grandfather_opt1_pub_second :=
SchoolSampData[idx].pp_grandfather_opt1_pub_second;
    Result.pp_grandfather_opt2_pub_second :=
SchoolSampData[idx].pp_grandfather_opt2_pub_second;
    Result.pp_grandfather_opt1_priv_second :=
SchoolSampData[idx].pp_grandfather_opt1_priv_second;
    Result.pp_grandfather_opt2_priv_second :=
SchoolSampData[idx].pp_grandfather_opt2_priv_second;
    Result.stateabb := SchoolSampData[idx].stateabb;

    if Result.stateabb = '' then
        raise Exception.Create('Row not found in State_School_Samp_Data.xlsx for ' +
abb);
    end;

```

```

procedure TStateSchoolSampData.GetSystemData(stateabb: string);
var
    r: integer;
    bFound: boolean;
begin
    bFound := false;

    for r := 3 to Xls.RowCount do
        begin
            if Xls.GetStringFromCell(r,2) = stateabb then
                begin
                    numb_second_schools_pub := Xls.GetCellValue(r, 3).AsVariant;
                    numb_elem_schools_pub := Xls.GetCellValue(r, 4).AsVariant;
                    numb_second_schools_priv := Xls.GetCellValue(r, 5).AsVariant;
                    numb_elem_schools_priv := Xls.GetCellValue(r, 6).AsVariant;
                    numb_daycares := Xls.GetCellValue(r, 7).AsVariant;
                    pp_grandfather_mand_child := Xls.GetCellValue(r, 8).AsVariant;
                    pp_grandfather_opt_child := Xls.GetCellValue(r, 9).AsVariant;
                    pp_grandfather_mand_pub_elem := Xls.GetCellValue(r, 10).AsVariant;
                    pp_grandfather_opt_pub_elem := Xls.GetCellValue(r, 11).AsVariant;
                    pp_grandfather_mand_priv_elem := Xls.GetCellValue(r, 12).AsVariant;
                    pp_grandfather_opt_priv_elem := Xls.GetCellValue(r, 13).AsVariant;
                    pp_grandfather_opt1_pub_second := Xls.GetCellValue(r, 14).AsVariant;
                    pp_grandfather_opt2_pub_second := Xls.GetCellValue(r, 15).AsVariant;
                    pp_grandfather_opt1_priv_second := Xls.GetCellValue(r, 16).AsVariant;
                    pp_grandfather_opt2_priv_second := Xls.GetCellValue(r, 17).AsVariant;

                    bFound := true;
                    break;
                end
            end
        end
    end;

```

```

        end;
    end;

    if not bFound then
        raise Exception.Create('Row not found in State_School_Samp_Data.xlsx for ' +
stateabb);
    end;

procedure TStateSchoolSampData.LoadSchoolSampData;
var
    r, i: integer;
begin
    i := 0;

    for r := 3 to Xls.RowCount do
        begin
            inc(i);

            SchoolSampData[i].numb_second_schools_pub := Xls.GetCellValue(r, 3).AsVariant;
            SchoolSampData[i].numb_elem_schools_pub := Xls.GetCellValue(r, 4).AsVariant;
            SchoolSampData[i].numb_second_schools_priv := Xls.GetCellValue(r, 5).AsVariant;
            SchoolSampData[i].numb_elem_schools_priv := Xls.GetCellValue(r, 6).AsVariant;
            SchoolSampData[i].numb_daycares := Xls.GetCellValue(r, 7).AsVariant;
            SchoolSampData[i].pp_grandfather_mand_child := Xls.GetCellValue(r, 8).AsVariant;
            SchoolSampData[i].pp_grandfather_opt_child := Xls.GetCellValue(r, 9).AsVariant;
            SchoolSampData[i].pp_grandfather_mand_pub_elem := Xls.GetCellValue(r,
10).AsVariant;
            SchoolSampData[i].pp_grandfather_opt_pub_elem := Xls.GetCellValue(r,
11).AsVariant;
            SchoolSampData[i].pp_grandfather_mand_priv_elem := Xls.GetCellValue(r,
12).AsVariant;
            SchoolSampData[i].pp_grandfather_opt_priv_elem := Xls.GetCellValue(r,
13).AsVariant;
            SchoolSampData[i].pp_grandfather_opt1_pub_second := Xls.GetCellValue(r,
14).AsVariant;
            SchoolSampData[i].pp_grandfather_opt2_pub_second := Xls.GetCellValue(r,
15).AsVariant;
            SchoolSampData[i].pp_grandfather_opt1_priv_second :=
Xls.GetCellValue(r,16).AsVariant;
            SchoolSampData[i].pp_grandfather_opt2_priv_second :=
Xls.GetCellValue(r,17).AsVariant;

            SchoolSampData[i].stateabb := Xls.GetStringFromCell(r,2);
        end;
    end;

end.

```