```
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;
    p_grandfather_opt_pub: double;
    p_grandfather_opt_priv: double;
    p_grandfather_mand_pub: double;
    p_grandfather_mand_priv: double;
    p_grandfather_opt_child: double;
    p_grandfather_mand_child: double;
    stateabb: string;
  end;

  TStateSchoolSampData = class
  public
    numb_second_schools_pub: double;
    numb_elem_schools_pub: double;
    numb_second_schools_priv: double;
    numb_elem_schools_priv: double;
    numb_daycares: double;
    p_grandfather_opt_pub: double;
    p_grandfather_opt_priv: double;
    p_grandfather_mand_pub: double;
    p_grandfather_mand_priv: double;
    p_grandfather_opt_child: double;
    p_grandfather_mand_child: double;
    stateabb: string;

    constructor Create;
    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 := 'Sheet1';
  {
    State_School_Samp_Data.xlsx
    1 State char(25)
    2 numb_second_schools_pub float
    3 numb_elem_schools_pub float
    4 numb_second_schools_priv float
    5 numb_elem_schools_priv float
    6 numb_daycares float
    7 p_grandfather_opt_pub (0/1)
    8 p_grandfather_opt_priv (0/1)
    9 p_grandfather_mand_pub (0/1)
   10 p_grandfather_mand_priv (0/1)
   11 p_grandfather_opt_child
   12 p_grandfather_mand_child
   13 stabb char(2)
  }

  SetLength(SchoolSampData, 66 + 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.p_grandfather_opt_pub := SchoolSampData[idx].p_grandfather_opt_pub;
  Result.p_grandfather_opt_priv := SchoolSampData[idx].p_grandfather_opt_priv;
  Result.p_grandfather_mand_pub := SchoolSampData[idx].p_grandfather_mand_pub;
  Result.p_grandfather_mand_priv := SchoolSampData[idx].p_grandfather_mand_priv;
  Result.p_grandfather_opt_child := SchoolSampData[idx].p_grandfather_opt_child;
  Result.p_grandfather_mand_child := SchoolSampData[idx].p_grandfather_mand_child;
  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,13) = stateabb then
    begin
      numb_second_schools_pub := Xls.GetCellValue(r, 2).AsVariant;
      numb_elem_schools_pub := Xls.GetCellValue(r, 3).AsVariant;
      numb_second_schools_priv := Xls.GetCellValue(r, 4).AsVariant;
      numb_elem_schools_priv := Xls.GetCellValue(r, 5).AsVariant;
      numb_daycares := Xls.GetCellValue(r, 6).AsVariant;
      p_grandfather_opt_pub := Xls.GetCellValue(r, 7).AsVariant;
      p_grandfather_opt_priv := Xls.GetCellValue(r, 8).AsVariant;
      p_grandfather_mand_pub := Xls.GetCellValue(r, 9).AsVariant;
      p_grandfather_mand_priv := Xls.GetCellValue(r, 10).AsVariant;
      p_grandfather_opt_child := Xls.GetCellValue(r, 11).AsVariant;
      p_grandfather_mand_child := Xls.GetCellValue(r, 12).AsVariant;

      bFound := true;
      break;
    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, 2).AsVariant;
    SchoolSampData[i].numb_elem_schools_pub := Xls.GetCellValue(r, 3).AsVariant;
    SchoolSampData[i].numb_second_schools_priv := Xls.GetCellValue(r, 4).AsVariant;
    SchoolSampData[i].numb_elem_schools_priv := Xls.GetCellValue(r, 5).AsVariant;
    SchoolSampData[i].numb_daycares := Xls.GetCellValue(r, 6).AsVariant;
    SchoolSampData[i].p_grandfather_opt_pub := Xls.GetCellValue(r, 7).AsVariant;
    SchoolSampData[i].p_grandfather_opt_priv := Xls.GetCellValue(r, 8).AsVariant;
    SchoolSampData[i].p_grandfather_mand_pub := Xls.GetCellValue(r, 9).AsVariant;
    SchoolSampData[i].p_grandfather_mand_priv := Xls.GetCellValue(r, 10).AsVariant;
    SchoolSampData[i].p_grandfather_opt_child := Xls.GetCellValue(r, 11).AsVariant;
    SchoolSampData[i].p_grandfather_mand_child := Xls.GetCellValue(r, 12).AsVariant;
    SchoolSampData[i].stateabb := Xls.GetStringFromCell(r,13);
  end;
end;

end.
```