Delphi Cleaner

Delphi Uygulama Klasöründeki Gereksiz Dosyalari
Silen Bir Program

Table of Contents

1 Symbol Reference

1.1 prjDelphiDirectoryCleaner Namespace

This is namespace prjDelphiDirectoryCleaner.

Functions

	Name	Description
	DeleteTempFiles (see page 1)	bu metot recursive bir sekilde belirtilen klasörü dolasiyor varsa alt klasörüne gidiyor yoksa da dosyanin isimlerini kontrol ediyor dosya uzantisi .dcu olan veya .~ gibi olan bütün dosyalari tek tek siliyor sildigi dosyalarin bilgilerini de console üzerinde gösteriyor

Variables

Name	Description
deleteFileRec (see page 2)	This is variable prjDelphiDirectoryCleaner.deleteFileRec.
directoryPath (see page 2)	This is variable prjDelphiDirectoryCleaner.directoryPath.
result (see page 2)	This is variable prjDelphiDirectoryCleaner.result.

1.1.1 Functions

The following table lists functions in this documentation.

Functions

	Name	Description
	DeleteTempFiles (see page 1)	bu metot recursive bir sekilde belirtilen klasörü dolasiyor varsa alt klasörüne gidiyor yoksa da dosyanin isimlerini kontrol ediyor dosya uzantisi .dcu olan veya .~ gibi olan bütün dosyalari tek tek siliyor sildigi dosyalarin bilgilerini de console üzerinde gösteriyor

1.1.1.1 prjDelphiDirectoryCleaner.DeleteTempFiles Function

Pascal

```
procedure DeleteTempFiles(directoryPath: string);
```

File

prjDelphiDirectoryCleaner (see page 3)

Description

bu metot recursive bir sekilde belirtilen klasörü dolasiyor varsa alt klasörüne gidiyor yoksa da dosyanin isimlerini kontrol ediyor dosya uzantisi .dcu olan veya .~ gibi olan bütün dosyalari tek tek siliyor sildigi dosyalarin bilgilerini de console üzerinde gösteriyor

1.1.2 Variables

The following table lists variables in this documentation.

Variables

Name	Description
deleteFileRec (⊡ see page 2)	This is variable prjDelphiDirectoryCleaner.deleteFileRec.
directoryPath (⊡ see page 2)	This is variable prjDelphiDirectoryCleaner.directoryPath.
result (⊡ see page 2)	This is variable prjDelphiDirectoryCleaner.result.

1.1.2.1 prjDelphiDirectoryCleaner.deleteFileRec Variable

Pascal

```
deleteFileRec: TSearchRec;
```

File

prjDelphiDirectoryCleaner (⊡ see page 3)

Description

This is variable prjDelphiDirectoryCleaner.deleteFileRec.

1.1.2.2 prjDelphiDirectoryCleaner.directoryPath Variable

Pascal

```
directoryPath: string;
```

File

prjDelphiDirectoryCleaner (⊡ see page 3)

Description

This is variable prjDelphiDirectoryCleaner.directoryPath.

1.1.2.3 prjDelphiDirectoryCleaner.result Variable

Pascal

```
result: Integer;
```

File

prjDelphiDirectoryCleaner (⊡ see page 3)

Description

This is variable prjDelphiDirectoryCleaner.result.

1.2 **Files**

The following table lists files in this documentation.

Units

Name	Description
prjDelphiDirectoryCleaner.dpr (⊡ see page 3)	This is file prjDelphiDirectoryCleaner.dpr.

1.2.1 **prjDelphiDirectoryCleaner.dpr**

This is file prjDelphiDirectoryCleaner.dpr.

Body Source

```delphi
program prjDelphiDirectoryCleaner;

{$APPTYPE CONSOLE}

// DeleteFile ve FindClose metotlarinin düzgün çalisabilmesi için
// uses kisminda WinTypes uniti, SysUtils'in önünde olmalidir
// Yoksa FindClose metodunda asagidaki koddaki gibi TSearchRec
// yerine Cardinal türünde parametre aliyor
// bu nokta önemli dikkat edilmeli
uses
  Windows, WinTypes, SysUtils;

var
  directoryPath: string;


// bu metot recursive bir sekilde belirtilen klasörü dolasiyor
// varsa alt klasörüne gidiyor
// yoksa da dosyanin isimlerini kontrol ediyor
// dosya uzantisi .dcu olan veya .~ gibi olan bütün dosyalari
// tek tek siliyor
// sildigi dosyalarin bilgilerini de console üzerinde gösteriyor

procedure DeleteTempFiles(directoryPath: string);
var
  deleteFileRec: TSearchRec;
  result: Integer;
begin
  result := FindFirst(directoryPath + '\*.*', faDirectory, deleteFileRec);
  while result = 0 do
  begin
    if (deleteFileRec.Name <> '.') and (deleteFileRec.Name <> '..') then
    begin
      DeleteTempFiles(directoryPath + '\' + deleteFileRec.Name);
    end;
    result := FindNext(deleteFileRec);
  end;
  FindClose(deleteFileRec);

  result := FindFirst(directoryPath + '\*.~*', faAnyFile, deleteFileRec);
  while result = 0 do
  begin
    if not DeleteFile(PAnsiChar(directoryPath + deleteFileRec.Name)) then
    begin
```

```
        FileSetAttr(directoryPath + deleteFileRec.Name, 0);
        DeleteFile(PAnsiChar(directoryPath + '\' + deleteFileRec.Name));
        Writeln('The file ' + directoryPath + '\' + deleteFileRec.Name + ' has been deleted');
      end;
      result := FindNext(deleteFileRec);
    end;
  FindClose(deleteFileRec);

  result := FindFirst(directoryPath + '\*.dcu', faAnyFile, deleteFileRec);
  while result = 0 do
  begin
    if not DeleteFile(PAnsiChar(directoryPath + deleteFileRec.Name)) then
    begin
      FileSetAttr(directoryPath + deleteFileRec.Name, 0);
      DeleteFile(PAnsiChar(directoryPath + '\' + deleteFileRec.Name));
      Writeln('The file ' + directoryPath + '\' + deleteFileRec.Name + ' has been deleted');
    end;
    result := FindNext(deleteFileRec);
  end;
  FindClose(deleteFileRec);
end;

// program bir console application
// burada önemli olan nokta ise
// eger bu exe dosyasi çalistirilirken parametre almissa
// aldigi parametre dosyalarin silinecegi dizin oluyor ve
// ilgili metoda yönlendiriyor
// parametre almazsa bulundugu klasördeki dosyalari silmeye basliyor
begin
  if ParamCount = 1 then
  begin
    directoryPath := ParamStr(1);
    if DirectoryExists(directoryPath) then
    begin
      DeleteTempFiles(directoryPath);
    end;
  end
  else
  begin
    directoryPath := GetCurrentDir;
    DeleteTempFiles(directoryPath);
  end;
// Writeln(directoryPath);
// Writeln('Press a key to terminate program');
// Readln(a);
end.
```

Namespaces

Name	Description
prjDelphiDirectoryCleaner (⊡ see page 1)	This is namespace prjDelphiDirectoryCleaner.

Index