// Avoti:

// https://stackoverflow.com/questions/34831963/how-to-delete-file-in-c-sharp

// https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/

// https://www.geeksforgeeks.org/file-handling-c-sharp/

// daži youtube video interneta

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

abstract class FileSystemObject //uzstada pamata "blueprint" prieks failu sistemas

{

public string Name { get; set; } // uzstada failu name un parent piederibu un iespejas tikt mainitam (set) un skatitam (get)

public FolderObject Parent { get; set; }

public abstract string Type { get; } //liek nakotnes klasem noteikt kas par tipu tie ir (file vai folder)

public FileSystemObject(string name, FolderObject parent)

{

Name = name;

Parent = parent; //konstruktors kas dod noteiktu vardu un piederibu folderam.

}

public abstract void Show();

}

class FileObject : FileSystemObject

{

public override string Type => "file"; //nosaka tipu (file)

public FileObject(string name, FolderObject parent) : base(name, parent) { } //nosaka parent folder name

~FileObject()

{

Console.WriteLine($"Destruktors izsaukts datnei: {Name}"); //izveido datnu destruktoru

}

public override void Show()

{

Console.WriteLine($"/~{Name}~");

}

}

class FolderObject : FileSystemObject

{

public override string Type => "folder"; //nosaka tipu (folder)

public List<FileSystemObject> Children = new List<FileSystemObject>();

public FolderObject(string name, FolderObject parent = null) : base(name, parent) { }

// funkcija lai paradit bernus

public override void Show()

{

foreach (var child in Children)

{

if (child.Type == "folder")

Console.WriteLine($"/[{child.Name}]");

else

Console.WriteLine($"/~{child.Name}~");

}

}

// izveido "izveidot" metodi

public void Izveidot(string name, string veids)

{

if (veids == "mape")

{

AddChild(new FolderObject(name, this));

}

else if (veids == "datne")

{

AddChild(new FileObject(name, this));

try

{

File.WriteAllText(name + ".txt", "Jauna datne");

}

catch (Exception)

{

Console.WriteLine("Kļūda: nevar izveidot datni.");

}

}

}

public void AddChild(FileSystemObject obj)

{

foreach (var c in Children)

{

if (c.Name == obj.Name)

{

Console.WriteLine($"Objekts '{obj.Name}' jau eksiste."); //izveido add child funkciju lai pievienotu bernus folderiem

return;

}

}

Children.Add(obj);

}

//izveido "dzest" metodi

public void dzest(string name)

{

var obj = GetChild(name);

if (obj is FolderObject folder)

{

if (folder.Children.Count > 0)

{

Console.WriteLine($"Mape '{name}' nav tuksa. Vai velaties dzest? (j/n)"); //ja mape atrodas faili, tad apstiprina vai tiesam velas izdest failu.

var atb = Console.ReadLine();

if (atb != "j") return;

}

}

RemoveChild(name);

}

public void RemoveChild(string name) //tiek dzests berns

{

FileSystemObject obj = null;

foreach (var c in Children)

{

if (c.Name == name)

{

obj = c;

break;

}

}

if (obj != null)

{

Children.Remove(obj);

Console.WriteLine($"{obj.Type} '{name}' tika dzests."); //izvada komandu kad izdzes failu

}

else

{

Console.WriteLine($"{name} nav atrasts"); //gadijuma ja bernu nevar atrast - izvada to

}

}

public FileSystemObject GetChild(string name)

{

foreach (var c in Children)

{

if (c.Name == name) return c; //mekle izveleto bernu pec ta varda folderi, kad atrod - to izvada

}

return null; //ja neatrod vajadzigo bernu - izvada null

}

public FolderObject Parvietoties(string name)

{

var obj = GetChild(name); //mekle izveleto bernu un parbauda vai tas ir fails. ja ir tad to izvada

if (obj is FolderObject folder)

return folder;

return null;//ja neatrod vajadzigo - izvada null

}

~FolderObject()

{

Console.WriteLine($"Mapei {Name} izsaukts destruktors"); //pazino ka destruktors tiek izvadiits mapei

}

}

class FileSystem

{

FolderObject root = new FolderObject("c:"); // nosaka "saknes(root)" mapi (c:) un nosaka pasreizojo sekojoso mapi

FolderObject current;

public FileSystem()

{

current = root;

}

public string GetPath() //funkcija kas veido un izsaka pilno pasreizejas mapes atrasnas vietu, piem, c:/map1/map2/>.

{

string path = "";

FolderObject temp = current;

while (temp != null)

{

path = temp.Name + "/" + path;

temp = temp.Parent;

}

return path + ">";

}

public void Run()

{

while (true)

{

Console.Write(GetPath()); //izvada pasreizejo mnapes celu un gaida komandu ievadi

var line = Console.ReadLine();

if (string.IsNullOrWhiteSpace(line))

continue;

var args = line.Trim().Split(' ', StringSplitOptions.RemoveEmptyEntries);

var cmd = args[0].ToLower();

var rest = args.Skip(1).ToArray();

if (cmd == "exit" || cmd == "by") return;//ievadot exit/by programma parstaj darbibu

if (cmd == "mkdir") //ievadot mkdir, programma izveido jaunu mapi ar noteiktu nosaukumu

{

foreach (var p in rest)

{

current.Izveidot(p, "mape");//izveidos vairakas mapes attiecigi ievaditajiem vardiem

}

}

else if (cmd == "create")//ievadot komandu create, programma izveidos vienu vai vairakas datnes attieicigi ievaditajiem vardiem

{

foreach (var p in rest)//prieks katra ievadita varda izveido attiecigu datni

{

current.Izveidot(p, "datne");

}

}

else if (cmd == "rm")//ievadot tiek izdzesta pasreizejas atrasanas vietas mape

{

foreach (var p in rest)

{

current.dzest(p); //dzes mapi - ja vajadzigs tad pieprasa apstiprinajumu

}

}

else if (cmd == "del")

{

foreach (var p in rest)

{

var dat = current.GetChild(p) as FileObject; //mekle izveleto failu nosaukumu - ja tiek atrasts, tiek izdzests - ja ne - tiek atgriezts null

if (dat != null)

{ //TRY...CATCH ERROR

try

{

File.Delete(p + ".txt");//censas izdzest izveleto failu

}

catch

{

Console.WriteLine("Fails nav dzēšams."); //ja neizdodas - noker kludu un atbild atbilstosi

}

current.RemoveChild(p);//ja tiek atrasts fails - tiek izdzest un nonemts no berna saraksta

}

else

Console.WriteLine("Datne nav atrasta.");//ja neizdodas - atbild atbilstosi

}

}

else if (cmd == "cd")//parvietosanas funkcija

{

if (rest.Length == 0) continue; //ja netiek ievadita vajadziga info - programma vnks turpina stradat

if (rest[0] == "..")//ja tiek noradits, tad kods parvietojas uz vecaku mapi

{

if (current.Parent != null)//ja pasreizeja mape ir parent, tad programma parvietojas uz vecāko mapi. Lauj lietotajiem atgriezties un ieprieksejo dalu.

current = current.Parent;

}

else if (rest[0] == "/")//ja tiek ievadita komanda cd / tad programma parvietojas uz saknes mapi jeb c:

{

current = root;

}

else

{

var jauna = current.Parvietoties(rest[0]);//parbauda vai funkcija atgriez derigu mapi. Ja tiek - programma parvietojas uz to mapi

if (jauna != null)

current = jauna;

else

Console.WriteLine("Mape nav atrasta.");//ja netiek, tad tiek atbildets atbilstosi

}

}

else if (cmd == "dir")//funkcija dir tiek sakta

{

if (rest.Length == 1 && rest[0].StartsWith(">"))//parbauda vai ir vnks dir vai dir>

{

var nos = rest[0].Substring(1);

try

{

File.WriteAllLines(nos + ".txt", current.Children.Select(c => c.Name));//ja tiek izmantota komanda dir>, tad tiek izveidots fails, kura ieraksta visas pasreizejas mapes bernu nosaukumus

}

catch { }

var vecais = current.GetChild(nos);//tiek meginats dzests ieprieksejo bernu saraksts ar so nosaukumu ja tads ir

if (vecais != null)

current.RemoveChild(nos);

current.AddChild(new FileObject(nos, current));

}

else

{

current.Show();//ja komanda ir vnks dir, tad tiek paradits mapes sastavu (mapes un datnes)

}

}

else if (cmd == "edit")//uzsak edit funkciju

{

if (rest.Length == 1)

{//TRY... CATCH error funkcija

try

{

string saturs = File.ReadAllText(rest[0] + ".txt");

Console.WriteLine(saturs);//programma nolasa un izvada atbilstosa failu saturu

}

catch

{

Console.WriteLine("Fails nav pieejams.");//ja nevar nolasit atbilstoso failu, tiek izvadita atbilstosa atbilde

}

}

}

else

{

Console.WriteLine("Nezinama komanda.");//Ja visparigi kada komanda netiek saprasta vai tada neeksiste - atbild atbilstosi.

}

}

}

}

class Program

{

static void Main()

{

FileSystem fs = new FileSystem(); //Uzsak jauna faila sistemas objektu

fs.Run();//aktivize visu programmas darbibu

}

}