**Kodi i aplikacionit per Aplikacioni i Menaxhimit të Lëndës ne Console**

using System; // Libraria Kryesore

using System.Collections.Generic; // Libraria qe permban llojet e koleksioneve te gjenerikeve te drejtperdrejta dhe jo te drejtperdrejta

using System.Linq; // Libraria qe ofron klasa dhe ndërfaqe që mbështesin queries që përdorin Language-Integrated Query (LINQ).

using System.Text; // Libraria qe permban sintaksa te shkrimit te tekstit

using System.Threading.Tasks;// Libraria e sistemit qe ndihmon ne rastet kur programuesi shkruan gabim dhe ndihmon ne korrigjim,

// gjithashtu mbron programin

namespace tester // Emri i skedarit

{

internal class Program //Klasa internale me emrin program ne rastin kur duam te aksesojme nga shumë klasa

//të tjera brenda të njëjtit montim, por që dëshironi të siguroni kodin në asambletë e tjera

{

static void Main(string[] args) //Metoda kryesore per ekzekutim

{

// Krijo kurs

Course mathCourse = new Course

{

CourseName = "Mathematics"

};

// Menu loop

while (true)

{

Console.WriteLine("1. Add Student");

Console.WriteLine("2. Delete Student");

Console.WriteLine("3. Add/Delete Subjects and Grades");

Console.WriteLine("4. Generate Reports");

Console.WriteLine("5. Display All Students, Subjects, and Grades");

Console.WriteLine("6. Exit");

Console.Write("Enter your choice (1-6): ");

if (int.TryParse(Console.ReadLine(), out int choice)) //Kushti cdo numer qe do te zgjidhet dhe mund te jete imput dhe

//output te jete i plote

{

// Kur zgjedhim njerin duke shkruajtur numrat

switch (choice)

{

case 1:

AddStudent(mathCourse);

break;

case 2:

DeleteStudent(mathCourse);

break;

case 3:

AddOrDeleteSubjectsAndGrades(mathCourse);

break;

case 4:

GenerateReports(mathCourse);

break;

case 5:

DisplayAllStudentsSubjectsAndGrades(mathCourse);

break;

case 6:

Console.WriteLine("Exiting program.");

return;

default:

Console.WriteLine("Invalid choice. Please enter a number between 1 and 6.");

break;

}

}

else

{

Console.WriteLine("Invalid input. Please enter a number between 1 and 6.");

}

}

} **[Ergi]**

// funksioni statik pa argument dhe me parametra kur ndodh shtimi i studentit

static void AddStudent(Course course)

{

Console.Write("Enter student name: "); // Shfaqja ne terminal

string studentName = Console.ReadLine();// tipi string dhe te lexoje dhe te ruaje ne memorie

Student newStudent = new Student { Name = studentName }; // deklarate ne rastin kur shtojme student te ri

course.Students.Add(newStudent); // kjo gje e lejon shtimin e studentit

Console.WriteLine($"Student '{studentName}' added to the course."); // Shfaqja ne terminal pepohimin e regjistrimit te studentit

}

// funksioni pa argument dhe me parametra kur heqim studentin

static void DeleteStudent(Course course)

{

Console.Write("Enter student name to delete: "); // shfaqja ne terminal per te fshire studentin

string studentName = Console.ReadLine(); // tipi string ,lexim dhe te ruhet imputi ne memorje

var studentToRemove = course.Students.FirstOrDefault(s => s.Name.Equals(studentName, StringComparison.OrdinalIgnoreCase));

//Metoda qe kthen elementin e parë të një sekuence, ose një vlerë të prezgjedhur nëse nuk gjendet asnjë element.

if (studentToRemove != null) // kushti nese studenti qe kerkojme nuk eshte hequr

{

course.Students.Remove(studentToRemove); // Metoda e heqjes se studentit

Console.WriteLine($"Student '{studentName}' deleted from the course."); // Shfaqja ne terminal qe studenti u hoq nga kursi

}

else

{

Console.WriteLine($"Student '{studentName}' not found in the course."); // perndryshe te shfaqe qe studenti nuk u gjet

}

}

// funksioni pa argument dhe me parametra kur duam te fshijme ose te shtojme lendet dhe piket

static void AddOrDeleteSubjectsAndGrades(Course course)

{

Console.Write("Enter student name to add/delete subjects and grades: ");

string studentName = Console.ReadLine();

var student = course.Students.FirstOrDefault(s => s.Name.Equals(studentName, StringComparison.OrdinalIgnoreCase));

if (student != null) // Kushti nese studenti nuk eshte fshire

{

Console.WriteLine($"Adding/Deleting subjects and grades for {studentName}:");

while (true) // Cikli while shtohet ose hiqet nje bllok per te dhenat studenti lendet piket

{

Console.WriteLine($"Subjects for {studentName}: {string.Join(", ", student.Grades.Keys)}");

Console.Write("1. Add Grade\n2. Delete Grade\n3. Add Subject\n4. Delete Subject\n5. Exit\nEnter your choice (1-5): ");

//Shfaqja ne terminal e menyve te tjera

if (int.TryParse(Console.ReadLine(), out int choice)) // e njejta llogjike me pjesen e pare me if...else dhe switch

{

switch (choice)

{

case 1:

AddGrade(student);

break;

case 2:

DeleteGrade(student);

break;

case 3:

AddSubject(student);

break;

case 4:

DeleteSubject(student);

break;

case 5:

Console.WriteLine("Exiting subject and grade modification.");

return;

default:

Console.WriteLine("Invalid choice. Please enter a number between 1 and 5.");

break;

}

}

else

{

Console.WriteLine("Invalid input. Please enter a number between 1 and 5.");

}

}

}

else

{

Console.WriteLine($"Student '{studentName}' not found in the course.");

}

}

**[Sara xh]**

// funksion pa argument dhe me parametra per shtimin e pikeve

static void AddGrade(Student student)

{

Console.Write("Enter subject: ");

string subject = Console.ReadLine();

Console.Write($"Enter grade for {subject}: ");

while (true)

{

if (int.TryParse(Console.ReadLine(), out int grade)) // kushti nese piket te jene integer te lexohen imputet dhe te ruhen ne memorie

{

student.ModifyOrAddGrade(subject, grade); // deklarate per shtimin ose modifikimin e pikeve

Console.WriteLine($"Grade for {subject} added for {student.Name}."); // Shfaqja ne terminal

break;

}

else

{

Console.WriteLine("Invalid input. Please enter a numeric grade."); // Shfaqja ne terminal kur imputi eshte i pavlefshem

Console.Write($"Enter grade for {subject}: "); // Shfaqja e kerkeses ne terminal ne rastin e kerkeses se pikeve per lendet

}

}

}

//funksioni pa argument dhe me parametra kur hiqen piket

static void DeleteGrade(Student student)

{

Console.Write("Enter subject to delete grade from: "); // Shfaqja ne terminal e kerkeses per shkrimin e lendes per te fshire piket

string subjectToDelete = Console.ReadLine();// shkrimi i lendes te jete string , te lexohet, dhe te ruhet ne memoriie

//Kushti ne rastin kur hiqen piket e lendes kryesore (math)

if (student.Grades.ContainsKey(subjectToDelete))

{

student.Grades.Remove(subjectToDelete);// deklarata per fshirjen e pikeve kur lenda eshte fshire

Console.WriteLine($"Grade for {subjectToDelete} deleted for {student.Name}."); //Shfaqja ne terminal per piket e fshira te temes

}

else

{

Console.WriteLine($"Subject '{subjectToDelete}' not found for {student.Name}."); // shfaqja ne terminal kur lenda e hequr nuk gjinded

}

}

//funksioni pa argument dhe me parametra kur shtojme lende

static void AddSubject(Student student)

{

Console.Write("Enter new subject to add: "); // shfaqja ne terminal e nje kerkese per shtimin e lenndes

string newSubject = Console.ReadLine(); // te jete imputi ne string dhe lexim dhe ruajtje ne memorie

if (!student.Grades.ContainsKey(newSubject)) // kushti piket e lendes se shtuar nuk eshte shfaqur

{

student.Grades.Add(newSubject, 0); // te shtohen piket e lendes se re

Console.WriteLine($"Subject '{newSubject}' added for {student.Name}."); //shfaqja ne terminal qe lenda eshte shtuar

}

else

{

Console.WriteLine($"Subject '{newSubject}' already exists for {student.Name}."); // shfaqja ne termminal kur lenda ekziston

}

}

**[Amer]**

// funksioni me parametra dhe pa argument kur fshihet lenda

static void DeleteSubject(Student student)

{

Console.Write("Enter subject to delete: "); // shfaqja ne terminal per shkrimin e lendes per tu fshire

string subjectToDelete = Console.ReadLine(); // te jete imputi ne string

if (student.Grades.ContainsKey(subjectToDelete)) // Kushti nese piket e lendes se studentit nuk jane fshire

{

student.Grades.Remove(subjectToDelete); // deklarate e fshirjes se pikeve kur lenda eshte fshire

Console.WriteLine($"Subject '{subjectToDelete}' deleted for {student.Name}."); // shfaqja ne terminal kur lenda eshte fshire

}

else

{

Console.WriteLine($"Subject '{subjectToDelete}' not found for {student.Name}."); // shfaqja ne terminal kur lenda e fshire nuk gjendet

}

}

//funksioni pa argument dhe me parametra per gjenerimin e raportave

static void GenerateReports(Course course)

{

Console.WriteLine($"Reports for the course: {course.CourseName}\n"); // Shfaqja ne terminal per raportin e kursit

//Kur studentet jane te regjistruara dhe dua te shfaq raportet per njerin

foreach (var student in course.Students)

{

Console.WriteLine($"{student.Name}'s Report:"); // Shfapja ne terminal e Raportit te studentit

if (student.Grades.Any())

{

foreach (var grade in student.Grades)

{

Console.WriteLine($"{grade.Key}: {grade.Value}");

}

Console.WriteLine($"Average Grade: {student.CalculateAverageGrade():F2}"); // Shfaqja e pikeve mesatare te kalkuluara

Console.WriteLine($"Highest Grade: {student.GetHighestGrade()}"); // shfapja e pikes me te larte

Console.WriteLine($"Lowest Grade: {student.GetLowestGrade()}\n"); // shfaqja e pikes me te ulet

}

else

{

Console.WriteLine("No grades available for this student.\n");

}

}

}

// funksioni per shfaqjen e studenteve pjesemarres te kurseve

static void DisplayAllStudentsSubjectsAndGrades(Course course)

{

Console.WriteLine($"All Students, Subjects, and Grades for the course: {course.CourseName}\n");

// Shfaqja ne terminal e studenteve , lendeve dhe pikeve

**[Sara H]**

// Per cdo Student e listuara ne kursin e studenteve ne varesi te perzgjedhjes te shfaqet njeri

foreach (var student in course.Students)

{

Console.WriteLine($"{student.Name}'s Report:"); // Shfaqja ne terminal e Raportit te studentit

if (student.Grades.Any())

{

foreach (var grade in student.Grades)

{

Console.WriteLine($"{grade.Key}: {grade.Value}"); // Shfaqja e pikeve

}

}

else

{

Console.WriteLine("No grades available for this student."); // Shfaqja ne terminal qe kjo pike nuk eshte per studentin

}

Console.WriteLine();

}

}

}

class Student // Klasa student

{

public string Name { get; set; }

public Dictionary<string, int> Grades { get; set; }

public Student()

{

Grades = new Dictionary<string, int>();

}

public double CalculateAverageGrade() // metoda qe kthen mesataren e pikeve

{

return Grades.Values.Any() ? Grades.Values.Average() : 0.0; // kthe llogaritjen mesatare te pikeve

}

public int GetHighestGrade() // metoda qe shfaq piken me te larte

{

return Grades.Values.Any() ? Grades.Values.Max() : 0;

}

public int GetLowestGrade() // metoda qe shfaq piken me te ulet

{

return Grades.Values.Any() ? Grades.Values.Min() : 0;

}

public void ModifyOrAddGrade(string subject, int grade) // metoda qe na mundeson modifikimin ose shtimin e pikeve

{

if (Grades.ContainsKey(subject)) // kushti if else me rast se modifikojme ose shtojme piket

{

Grades[subject] = grade;

}

else

{

Grades.Add(subject, grade);

}

}

}

class Course // klasa kryesore e ekzekutimit

{

public string CourseName { get; set; } // Parameter PER EMRIN E LENDES

public List<Student> Students { get; set; } // Parameter PER TE LISTUAR EMRAT E STUDENTEVE

public Course() // metoda qe shfaq listen e studenteve te rregjistruar

{

Students = new List<Student>();

}

}

}



