using RequestTrackerModelLibrary;

namespace RequestTrackerModelLibrary

{

public class Employee

{

int age;

DateTime dob;

public int Id { get; set; }

public string Name { get; set; } = string.Empty;

public int Age

{

get

{

return age;

}

}

public DateTime DateOfBirth

{

get => dob;

set

{

dob = value;

age = ((DateTime.Today - dob).Days) / 365;

}

}

public double Salary { get; set; }

public Employee()

{

Id = 0;

Name = string.Empty;

Salary = 0.0;

DateOfBirth = new DateTime();

}

/// <summary>

/// Initializes the values for the properties

/// </summary>

/// <param name="id"></param>

/// <param name="name"></param>

/// <param name="dateOfBirth"></param>

/// <param name="salary"></param>

public Employee(int id, string name, DateTime dateOfBirth, double salary)

{

Id = id;

Name = name;

DateOfBirth = dateOfBirth;

Salary = salary;

}

public void BuildEmployeeFromConsole()

{

Console.WriteLine("Please enter the Name");

Name = Console.ReadLine() ?? String.Empty;

Console.WriteLine("Please enter the Date of birth");

DateOfBirth = Convert.ToDateTime(Console.ReadLine());

Console.WriteLine("Please enter the Basic Salary");

Salary = Convert.ToDouble(Console.ReadLine());

}

public void PrintEmployeeDetails()

{

Console.WriteLine("Employee Id : " + Id);

Console.WriteLine("Employee Name " + Name);

Console.WriteLine("Date of birth : " + DateOfBirth);

Console.WriteLine("Employee Age : " + Age);

Console.WriteLine("Employee Salary : Rs." + Salary);

}

}

}

using RequestTrackerModelLibrary;

namespace RequestTrackerApplication

{

internal class Program

{

Employee[] employees;

public Program()

{

employees = new Employee[3];

}

void PrintMenu()

{

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

Console.WriteLine("2. Print Employees");

Console.WriteLine("3. Search Employee by ID");

Console.WriteLine("4. Delete an Employee by ID");

Console.WriteLine("5. Update Employee Details");

Console.WriteLine("0. Exit");

}

/// <summary>

/// All the user options

/// </summary>

void EmployeeInteraction()

{

int choice = 0;

do

{

PrintMenu();

Console.WriteLine("Please select an option");

choice = Convert.ToInt32(Console.ReadLine());

switch (choice)

{

case 0:

Console.WriteLine("Bye.....");

break;

case 1:

AddEmployee();

break;

case 2:

PrintAllEmployees();

break;

case 3:

SearchAndPrintEmployee();

break;

case 4:

SearchAndDeleteEmployee();

break;

case 5:

SearchAndUpdate();

break;

default:

Console.WriteLine("Invalid choice. Try again");

break;

}

} while (choice != 0);

}

/// <summary>

/// gets the choice inputs from user console

/// </summary>

void AddEmployee()

{

if (employees[employees.Length - 1] != null)

{

Console.WriteLine("Sorry we have reached the maximum number of employees");

return;

}

for (int i = 0; i < employees.Length; i++)

{

if (employees[i] == null)

{

employees[i] = CreateEmployee(i);

}

}

}

/// <summary>

/// Adds employees to the array

/// </summary>

void PrintAllEmployees()

{

bool flag = false;

for (int i = 0; i < employees.Length; i++)

{

if (employees[i] != null)

{

flag = true;

PrintEmployee(employees[i]);

}

}

if(flag == false)

{

Console.WriteLine("No Employees are present..");

}

}

/// <summary>

/// Prints all the employees details

/// </summary>

/// <param name="id"></param>

/// <returns>return nothing</returns>

Employee CreateEmployee(int id)

{

Employee employee = new Employee();

employee.Id = 101 + id;

employee.BuildEmployeeFromConsole();

return employee;

}

void PrintEmployee(Employee employee)

{

Console.WriteLine("---------------------------");

employee.PrintEmployeeDetails();

Console.WriteLine("---------------------------");

}

/// <summary>

/// Invoke the printEmployeeDetails method

/// </summary>

/// <returns>return nothing</returns>

int GetIdFromConsole()

{

int id = 0;

Console.WriteLine("Please enter the employee Id");

while (!int.TryParse(Console.ReadLine(), out id))

{

Console.WriteLine("Invalid entry. Please try again");

}

return id;

}

/// <summary>

/// get the id from the user from console

/// </summary>

void SearchAndDeleteEmployee()

{

int id = GetIdFromConsole();

DeleteEmployeeById(id);

Console.WriteLine($"The Employee id {id} is been deleted!!");

}

/// <summary>

/// method searches using getidfromconsole from user and delete by its id

/// </summary>

void SearchAndUpdate()

{

int id = GetIdFromConsole();

UpdateEmployeeById(id);

Console.WriteLine($"The Employee details of id : {id} has been updated");

}

/// <summary>

/// Update employee details using id taken from the user console.

/// </summary>

void SearchAndPrintEmployee()

{

Console.WriteLine("Print One employee");

int id = GetIdFromConsole();

Employee employee = SearchEmployeeById(id);

if (employee == null)

{

Console.WriteLine("No such Employee is present");

return;

}

PrintEmployee(employee);

}

/// <summary>

/// Update the employee by id

/// </summary>

/// <param name="id"></param>

void UpdateEmployeeById(int id)

{

Employee employee = null;

bool flag = false;

for (int i = 0; i < employees.Length; i++)

{

if (employees[i] != null && employees[i].Id == id)

{

flag = true;

Console.WriteLine("What do u want to update 1.Name\t 2.Date Of Birth\t 3.Basic Salary\t");

int ch = Convert.ToInt32(Console.ReadLine());

switch (ch)

{

case 1:

Console.WriteLine("Enter the new name for the employee : ");

employees[i].Name = Console.ReadLine();

break;

case 2:

Console.WriteLine("Enter the new DOB : ");

employees[i].DateOfBirth = Convert.ToDateTime(Console.ReadLine());

break;

case 3:

Console.WriteLine("Enter the new Basic Salary : ");

employees[i].Salary = Convert.ToDouble(Console.ReadLine());

break;

}

break;

}

}

if(flag == false)

{

Console.WriteLine($"No Employee with id {id}");

}

}

/// <summary>

/// search for the employee in the employees array and retun the employee reference

/// </summary>

/// <param name="id"></param>

/// <returns>return the searched employee reference</returns>

Employee SearchEmployeeById(int id)

{

Employee employee = null;

for (int i = 0; i < employees.Length; i++)

{

if (employees[i] != null && employees[i].Id == id)

{

employee = employees[i];

break;

}

}

return employee;

}

/// <summary>

/// Delete the employee by id

/// </summary>

/// <param name="id"></param>

void DeleteEmployeeById(int id)

{

Employee employee = null;

for (int i = 0; i < employees.Length; i++)

{

if (employees[i] != null && employees[i].Id == id)

{

employees[i] = null;

break;

}

}

}

static void Main(string[] args)

{

Program program = new Program();

program.EmployeeInteraction();

//program.StartGame();

}

