**Технически университет – гр. Варна**

Компютърни системи и технологии

КУРСОВА РАБОВА

по

C#

Вариант 1

Изготвил: Проверил:

Тихомир Каменов доц. Божикова

Ф.№: 61462005

спец: м. КСТ СИ

сем: 1

Избран е вариант 1 от възможните за курсова работа. Задачата реализира управление на фирма, това включва – добавяне, изтриване, сортиране и търсене на служители и т.н.

За реализацията е използван езикът за програмиране от високо ниво C# заедно с LINQ(Language-Integrated Query).

Код на задачата.

Class Person:

namespace coursework

{

public class Person

{

public Person():

this(string.Empty, 0)

{

}

public Person(string name, int age)

{

this.Name = name;

this.Age = age;

}

public string Name { get; set; }

public int Age { get; set; }

public override string ToString()

{

return "Name: " + this.Name +

" Age: " + this.Age.ToString();

}

}

}

Class Employee:

namespace coursework

{

using System;

public class Employee : Person

{

public Employee() :

this(string.Empty, 0, 0, 0, string.Empty, 0)

{

}

public Employee(string name, int age, int id, int experience, string workPosition, decimal salary) : base(name, age)

{

this.Id = id;

this.Experience = experience;

this.WorkPosition = workPosition;

this.Salary = salary;

}

public int Id { get; set; }

public int Experience { get; set; }

public string WorkPosition { get; set; }

public decimal Salary { get; set; }

public override string ToString()

{

return base.ToString() + " ID: " + this.Id.ToString() + " Experience: " + this.Experience.ToString() +

" years Work Position: " + this.WorkPosition + " Salary: " + this.Salary.ToString();

}

}

}

Class Firm:

namespace coursework

{

using System;

using System.IO;

using System.Linq;

using System.Collections.Generic;

public class Firm

{

private List<Employee> employeeList;

public Firm()

{

this.employeeList = new List<Employee>();

}

public List<Employee> EmployeeList

{

get

{

return this.employeeList;

}

set

{

this.employeeList = value;

}

}

public void Load(string fileName)

{

StreamReader reader = new StreamReader(fileName);

using (reader)

{

string line;

while ((line = reader.ReadLine()) != null)

{

string[] parts = line.Split(new string[] { "\*/\*" }, StringSplitOptions.None);

Employee employee = new Employee()

{

Name = parts[0],

Age = int.Parse(parts[1]),

Id = int.Parse(parts[2]),

Experience = int.Parse(parts[3]),

WorkPosition = parts[4],

Salary = Decimal.Parse(parts[5], System.Globalization.CultureInfo.InvariantCulture)

};

EmployeeList.Add(employee);

}

}

}

public void Save(string fileName)

{

if (EmployeeList.Count != 0)

{

string[] lines = new string[EmployeeList.Count];

int counter = 0;

foreach (Employee employee in EmployeeList)

{

lines[counter] = employee.Name + "\*/\*" + employee.Age + "\*/\*" + employee.Id + "\*/\*" +

employee.Experience + "\*/\*" + employee.WorkPosition + "\*/\*" + employee.Salary;

counter++;

}

System.IO.File.WriteAllLines(fileName, lines);

Console.WriteLine("Data was saved to file.");

}

else

{

Console.WriteLine("There is no data to be saved to file.");

}

}

public void Add()

{

Console.WriteLine("Add new employee. Please enter the next fields");

Employee employee = new Employee();

Console.Write("Full name:");

employee.Name = Console.ReadLine();

bool parseFlag = false;

int age;

do

{

Console.Write("Age:");

parseFlag = int.TryParse(Console.ReadLine(), out age);

} while (!parseFlag);

employee.Age = age;

parseFlag = false;

int id;

do

{

Console.Write("ID:");

parseFlag = int.TryParse(Console.ReadLine(), out id);

foreach(Employee emp in employeeList)

{

if (emp.Id == id)

{

Console.WriteLine("There is allready employee with this id, please change it. (exmp: {0})", employeeList.Count + 1);

parseFlag = false;

}

}

} while (!parseFlag);

employee.Id = id;

parseFlag = false;

int experience;

do

{

Console.Write("Experience:");

parseFlag = int.TryParse(Console.ReadLine(), out experience);

} while (!parseFlag);

employee.Experience = experience;

Console.Write("Work position:");

employee.WorkPosition = Console.ReadLine();

parseFlag = false;

decimal salary;

do

{

Console.Write("Salary:");

parseFlag = decimal.TryParse(Console.ReadLine(), System.Globalization.NumberStyles.Currency,

System.Globalization.CultureInfo.InvariantCulture, out salary);

} while (!parseFlag);

employee.Salary = salary;

EmployeeList.Add(employee);

Console.WriteLine("Employee was successfully added. ");

}

public List<Employee> Search()

{

Console.WriteLine("1. Name.");

Console.WriteLine("2. Age.");

Console.WriteLine("3. ID.");

Console.WriteLine("4. Experience.");

Console.WriteLine("5. WorkPosition.");

Console.WriteLine("6. Salary bigger then.");

Console.WriteLine("7. Salary less then.");

int min, max;

min = 1; max = 7;

int choise;

Console.WriteLine("Please select criteria... ");

while (!int.TryParse(Console.ReadLine(), out choise) || choise < min || choise > max)

{

Console.WriteLine("Error enter number form {0} to {1}", min, max);

};

Console.WriteLine("Enter criteria:");

string criteria = Console.ReadLine();

var result = Enumerable.Empty<Employee>();

var culture = System.Globalization.CultureInfo.InvariantCulture;

switch (choise)

{

case 1:

result = EmployeeList.Where(e => e.Name == criteria);

break;

case 2:

result = EmployeeList.Where(e => e.Age == int.Parse(criteria));

break;

case 3:

result = EmployeeList.Where(e => e.Id == int.Parse(criteria));

break;

case 4:

result = EmployeeList.Where(e => e.Experience == int.Parse(criteria));

break;

case 5:

result = EmployeeList.Where(e => e.WorkPosition == criteria);

break;

case 6:

result = EmployeeList.Where(e => e.Salary >= Decimal.Parse(criteria, culture));

break;

case 7:

result = EmployeeList.Where(e => e.Salary <= Decimal.Parse(criteria, culture));

break;

}

return result.ToList<Employee>();

}

public void Remove(List<Employee> list)

{

if (list.Count > 0)

{

foreach (Employee employee in list)

{

if (employeeList.Contains(employee))

{

EmployeeList.Remove(employee);

}

}

Console.WriteLine("Employees removed successfully");

}

else

{

Console.WriteLine("No employees found!");

}

}

public void Display(bool search = false)

{

string tabs = new String('\t', 10);

if (EmployeeList.Count > 0)

{

foreach (Employee employee in EmployeeList)

{

Console.WriteLine(employee);

Console.WriteLine(tabs);

}

}

else

{

if(search)

{

Console.WriteLine("Cant find employees with that criteria!");

}

else

{

Console.WriteLine("Data is not loaded !");

}

}

}

public void Sort()

{

Console.WriteLine("1. Name.");

Console.WriteLine("2. Age.");

Console.WriteLine("3. ID.");

Console.WriteLine("4. Experience.");

Console.WriteLine("5. Salary.");

int min, max;

min = 1; max = 5;

int choise;

Console.WriteLine("Please select sort method... ");

while (!int.TryParse(Console.ReadLine(), out choise) || choise < min || choise > max)

{

Console.WriteLine("Error enter number form {0} to {1}", min, max);

};

var culture = System.Globalization.CultureInfo.InvariantCulture;

var result = Enumerable.Empty<Employee>();

switch (choise)

{

case 1:

result = EmployeeList.OrderBy(e => e.Name);

break;

case 2:

result = EmployeeList.OrderBy(e => e.Age);

break;

case 3:

result = EmployeeList.OrderBy(e => e.Id);

break;

case 4:

result = EmployeeList.OrderBy(e => e.Experience);

break;

case 5:

result = EmployeeList.OrderBy(e => e.Salary);

break;

}

EmployeeList = result.ToList<Employee>();

}

}

}

Class Program:

namespace coursework

{

using System;

using System.Collections.Generic;

public class Program

{

static void Main(string[] args)

{

int min, max;

min = 1; max = 8;

Firm firmObject = new Firm();

string fileName = "../../employees.txt";

Console.WriteLine("- Coursework Tihomir Kamenov C# -");

int choise = 0;

do

{

Console.WriteLine("1. Load the data from file. ");

Console.WriteLine("2. Add new employee. ");

Console.WriteLine("3. Fire an employee. ");

Console.WriteLine("4. Search an employee. ");

Console.WriteLine("5. Sort employees. ");

Console.WriteLine("6. Save the changes to file. ");

Console.WriteLine("7. Display employees. ");

Console.WriteLine("8. Exit. ");

Console.WriteLine("\nPlease select from menu... ");

while (!int.TryParse(Console.ReadLine(), out choise) || choise < min || choise > max)

{

Console.WriteLine("Error enter number form {0} to {1}", min, max);

};

switch (choise)

{

case 1:

firmObject.Load(fileName);

Console.WriteLine("Data is successfully loaded ! \n");

break;

case 2:

Console.WriteLine("Add new employee to firm... ");

firmObject.Add();

break;

case 3:

Console.WriteLine("Removing employees from firm... ");

if (firmObject.EmployeeList.Count == 0)

firmObject.Load(fileName);

firmObject.Remove(firmObject.Search());

break;

case 4:

Console.WriteLine("Searching for employees... ");

if(firmObject.EmployeeList.Count == 0)

firmObject.Load(fileName);

Firm newFirmObject = new Firm();

newFirmObject.EmployeeList = firmObject.Search();

Console.WriteLine("Results: ");

newFirmObject.Display(true);

break;

case 5:

Console.WriteLine("Sorting employees... ");

if (firmObject.EmployeeList.Count == 0)

firmObject.Load(fileName);

firmObject.Sort();

Console.WriteLine("Employees sorted!");

break;

case 6:

firmObject.Save(fileName);

break;

case 7:

firmObject.Display();

break;

}

} while (choise != max);

}

}

}

employees.txt: за разделител на полетата е използван символният низ “\*/\*“.

Tihomir Kamenov\*/\*23\*/\*1\*/\*0\*/\*Web Developer\*/\*750

Dragomir Ivanov\*/\*23\*/\*2\*/\*2\*/\*C Programer\*/\*2000.99

Ivan Ivanov\*/\*68\*/\*3\*/\*25\*/\*Senior Web Developer\*/\*1500.99

Petur Kamenov\*/\*60\*/\*4\*/\*30\*/\*Teacher\*/\*1000.99

Petko Petkov\*/\*20\*/\*5\*/\*1\*/\*SEO\*/\*850