Отчет по лабораторному занятию 8

Тема: «Разработка структур данных.»

Выполнил: студент группы ИСП-6 Алёшин Максим

Вариант 3

1. Разработать тип Сотрудник, в котором содержаться поля: имя; фамилия; отчество; возраст; должность; отдел разработки. Возможные действия, совершаемые над сотрудником: Добавить сотрудника; изменить возраст сотрудника; изменить должность сотрудника; изменить отдел разработки сотрудника.

2. Разработать тип Отдел разработки, в котором содержаться поля: название отдела; управляющий отделом (выбирается из сотрудников), кол-во сотрудников. Возможные действия совершаемые над отделом разработки: поменять название отдела; поменять управляющего отделом; поменять кол-во сотрудников.

Код программы:

using System;

using System.Collections.Generic;

namespace ConsoleApp7

{

class Jober

{

public string name;

public string familia;

public string otchestvo;

public int age;

public string doljnost;

public string otdelRazrabotki;

}

class DevelopmentDepartment

{

public string name;

public Jober manager;

public int howgobers;

}

class Program

{

//static Jober Sotrudnik = new Jober();

static Jober[] rabotniki;

static DevelopmentDepartment Depo = new DevelopmentDepartment();

static void AddToArray(Jober rabotnik)

{

if (rabotniki == null)

{

rabotniki = new Jober[] { rabotnik };

}

else

{

Jober[] joberInfo = new Jober[rabotniki.Length + 1];

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

{

joberInfo[i] = rabotniki[i];

}

joberInfo[joberInfo.Length - 1] = rabotnik;

rabotniki = joberInfo;

}

Depo.howgobers = rabotniki.Length;

}

static void SelectAnEmployee()

{

Console.Clear();

if (Depo.manager!=null)

{

AddToArray(Depo.manager);

Depo.manager = null;

}

Vivid();

Console.WriteLine("Выберите сотрудника, которого хотите выбрать на должностью менеджера: ");

int nomer = int.Parse(Console.ReadLine());

Depo.manager = rabotniki[nomer - 1];

//Depo.manager = new Jober()

//{

// name = rabotniki[nomer-1].name,

// familia = rabotniki[nomer-1].familia,

// otchestvo = rabotniki[nomer-1].otchestvo,

// age = rabotniki[nomer-1].age,

// otdelRazrabotki = rabotniki[nomer-1].otdelRazrabotki,

// doljnost = rabotniki[nomer-1].doljnost

//};

DeleteAtArray(nomer-1);

Console.Clear();

}

static void NameDepartment()

{

Console.Clear();

Console.WriteLine("Введите название отдела: ");

Depo.name = Console.ReadLine();

Console.Clear();

}

static void DobavkaJobers()

{

Console.Clear();

Jober jober = new Jober();

Console.Write($"Введите имя сотрудника: ");

jober.name = Console.ReadLine();

Console.Write($"Введите фамилию сотрудника: ");

jober.familia = Console.ReadLine();

Console.Write($"Введите отчество сотрудника: ");

jober.otchestvo = Console.ReadLine();

Console.Write($"Введите возраст сотрудника: ");

jober.age = int.Parse(Console.ReadLine());

Console.Write($"Введите должность сотрудника: ");

jober.doljnost = Console.ReadLine();

Console.Write($"Введите отдел разработки сотрудника: ");

jober.otdelRazrabotki = Console.ReadLine();

AddToArray(jober);

Console.Clear();

Depo.howgobers = rabotniki.Length;

}

static void Vivid()

{

Console.Clear();

if (Depo.name!=null)

{

Console.WriteLine($"Название отдела: ");

Console.WriteLine(Depo.name);

}

if (Depo.manager!=null)

{

Console.WriteLine($"Менеджер: {Depo.manager.name} {Depo.manager.familia} {Depo.manager.otchestvo}");

}

Console.WriteLine($"Кол-во сотрудников: {Depo.howgobers}\n");

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

{

Console.WriteLine($"Номер сотрудника: {i + 1}\n" +

$"Имя сотрудника: {rabotniki[i].name}\n" +

$"Фамилия сотрудника: {rabotniki[i].familia}\n" +

$"Отчество сотрудника: {rabotniki[i].otchestvo}\n" +

$"Возраст сотрудника: {rabotniki[i].age}\n" +

$"Должность сотрудника: {rabotniki[i].doljnost}\n" +

$"Отдел разработки сотрудника: {rabotniki[i].otdelRazrabotki}\n");

}

}

static void ChangeAge()

{

Vivid();

Console.Write("Номер сотрудника, у которого вы хотите изменить возраст: ");

int number = int.Parse(Console.ReadLine());

Console.Write($"Введите новый возраст: ");

rabotniki[number - 1].age = int.Parse(Console.ReadLine());

Console.Clear();

}

static void DeleteJober()

{

Vivid();

Console.Write("Номер сотрудника, которого хотите удалить: ");

int number = int.Parse(Console.ReadLine());

DeleteAtArray(number);

Console.Clear();

Depo.howgobers = rabotniki.Length;

}

static void DeleteAtArray(int index)

{

Jober[] tmpArray = new Jober[rabotniki.Length - 1];

int tmpArrayIndex = 0;

if (rabotniki.Length == 1)

{

rabotniki = null;

}

else

{

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

{

if (index != i)

{

tmpArray[tmpArrayIndex] = rabotniki[i];

tmpArrayIndex++;

}

}

rabotniki = tmpArray;

}

Depo.howgobers = rabotniki.Length;

}

static void ChangeDoljnostJober()

{

Vivid();

Console.Write("Номер сотрудника, у которого вы хотите изменить должность: ");

int number = int.Parse(Console.ReadLine());

Console.Write($"Введите новую должность: ");

rabotniki[number - 1].doljnost = (Console.ReadLine());

Console.Clear();

}

static void ChangeOtdelRazrabotki()

{

Vivid();

Console.Write("Номер сотрудника, у которого вы хотите изменить отдел разработки: ");

int number = int.Parse(Console.ReadLine());

Console.Write($"Введите новый отдел разработки: ");

rabotniki[number - 1].otdelRazrabotki = (Console.ReadLine());

Console.Clear();

}

static void Main(string[] args)

{

while (true)

{

Console.Write($"Выберите действие которое хотите сделать:\n" +

$"1) Добавить сотрудника\n" +

$"2) Изменить возраст сотрудника\n" +

$"3) Изменить должность сотрудника\n" +

$"4) Изменить отдел разработки сотрудника\n" +

$"5) Вывести всех сотрудников\n" +

$"6) Удалить сотрудника\n" +

$"7) Добавить менджера\n" +

$"8) Изменить название отдела\n\n" +

$"Ваш выбор: ");

int vibrator = int.Parse(Console.ReadLine());

switch (vibrator)

{

case 1:

DobavkaJobers();

break;

case 2:

ChangeAge();

break;

case 3:

ChangeDoljnostJober();

break;

case 4:

ChangeOtdelRazrabotki();

break;

case 5:

Vivid();

break;

case 6:

DeleteJober();

break;

case 7:

SelectAnEmployee();

break;

case 8:

NameDepartment();

break;

}

}

}

}

}

