**Федеральное государственное образовательное бюджетное учреждение  
высшего образования   
«Финансовый университет при Правительстве РФ»**

**КОЛЛЕДЖ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ**

Отчёт о выполнении Квалификационного экзамена

по ПМ.01 Разработка программных модулей программного обеспечения для компьютерных систем.

Выполнила студентка

Краснянская Екатерина

Группа 3ПКС-320

Экзаменационный билет №3.

Проверил: преподаватель

\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

оценка подпись

Москва, 2022

**Цель работы**: освоить работу с классами и коллекциями на языке С#. Разработать программного кода на языке Assembler для обработки данных.

# Результаты выполнения задания №1.

Написано консольное приложение, в котором: создан класс турист, имеющий характеристики: Фамилия, коллекция маршрутов. Класс имеет: конструкторы для создания туристов; метод заполнения сведений о маршрутах туриста; метод вывода информации в консоль и в текстовый файл; метод подсчёта среднего маршрута. Написано отдельное консольное приложение на языке Assembler, вычисляющее средний маршрут туриста и выводящее в консоль такие данные как: фамилию туриста, перечень маршрутов, средний маршрут туриста

**Пример результатов выполнения кода на языке С#:**

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1. Вывод данных в консоли.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2. Вывод данных в текстовый файл.

**Код программы:**

using Turistt;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

namespace Turistt

{

class Turist

{

private string surname; // фамилия туриста

private List<int> routes; // коллекция маршрутов

public string Surname { get => surname; } //делаем фамилию доступной

public List<int> Routes { get => routes; } //делаем маршруты доступными

public Turist(string surname) //конструктор для создания фамилии туриста и объявлении коллекции маршрутов

{

this.surname = surname;

this.routes = new List<int>();

}

// Constructor to create a turist with a surname and a list of routes

//public Turist(string surname, string routes) //конструктор для создания фамилии и присваивания коллекции маршрутов значения

//{

// this.surname = surname;

// this.routes = ParseRoutes(routes);

//}

private List<int> ParseRoutes(string routes) //метод заполнения коллекции

{

try

{

return routes.Split(',').Select(int.Parse).ToList();

}

catch (FormatException)

{

return new List<int>();

}

}

public int CalculateTotal() //метод вычисления суммы маршрутов

{

return routes.Sum();

}

public int CalculateRoutes() //метод возвращающий колличество маршрутов

{

return routes.Count();

}

public double CalculateAverage() // расчет среднего арифметического

{

if (routes.Count == 0)

{

return 0;

}

return (double)CalculateTotal() / routes.Count;

}

}

}

class Program

{

static void Main(string[] args)

{

//Console.WriteLine("Введите фамилию первого туриста"); string name1 = Console.ReadLine();

//Console.WriteLine("Введите фамилию второго туриста"); string name2 = Console.ReadLine();

//Console.WriteLine("Введите фамилию третьего туриста"); string name3 = Console.ReadLine();

List<Turist> turists = new List<Turist>(); //коллекция туристов

//туристы

turists.Add(new Turist("Иванов"));

turists.Add(new Turist("Петров"));

turists.Add(new Turist("Сидоров"));

Random random = new Random();//подключение рандома

foreach (Turist turist in turists) //генерация колличества маршрутов и их значений

{

for (int i = 0; i < random.Next(1, 11); i++)

{

turist.Routes.Add(random.Next(1, 40001));

}

}

foreach (Turist turist in turists) //вывод данных в консоль и в файл

{

Console.WriteLine("Маршруты туриста " + turist.Surname + ": " + string.Join(", ", turist.Routes));

Console.WriteLine("среднии маршрут : " + Math.Round(turist.CalculateAverage(), 2));

try

{

FileInfo fileInfo = new FileInfo(@"D:\пп\Krasnaynskay.txt");

StreamWriter sw = fileInfo.AppendText();

sw.WriteLine("Маршруты туриста " + turist.Surname + ": " + string.Join(", ", turist.Routes));

sw.WriteLine("среднии маршрут : " + Math.Round(turist.CalculateAverage(), 2));

//sw.WriteLine("Кол-во положительных оценок ");

sw.Close();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

//int numFives = turists.Sum(turist => turist.Routes.Count(grade => grade == 5));

//int numFours = turists.Sum(turist => turist.Routes.Count(grade => grade == 4));

//int numFours1 = turists.Sum(turist => turist.Routes.Count());

//Console.WriteLine(numFives);

//Console.WriteLine(numFours);

//Console.WriteLine(numFours1);

//double percentage = (numFives + numFours) \* 100 / numFours1;

// Display the percentage on the screen

//Console.WriteLine("Percentage of knowledge quality: " + percentage + "%");

Console.ReadLine();

}

}

**Пример результатов выполнения кода на языке Assembler:**

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3. Вывод данных в консоли на языке Assembler.

Код программы:

|  |
| --- |
| format PE console include 'win32a.inc'  start:  macro prntn op1,op2  {  local metka   mov esi,op1   mov ecx,op2  mov eax,0  metka:   mov eax,[esi]   push ecx   cinvoke printf,fmat,eax   pop ecx   add esi,4   loop metka  }  lea ebx,[A]  mov [ebx], dword 10958 mov [ebx+4],dword 20725 mov [ebx+8],dword 38516 mov [ebx+12],dword 27747 cinvoke printf, text6  prntn A,[N]  lea eax,[A]  push eax  mov eax,[N]  push eax  call average  mov ebx,eax  cinvoke printf, text3,eax  cinvoke printf, text  lea eax,[A]  push eax  mov eax,[N]  push eax  push ebx  call save  lea ebx,[B]  mov [ebx], dword 14212 mov [ebx+4],dword 36703 mov [ebx+8],dword 9235 cinvoke printf, text7  prntn B,[Nn]  lea eax,[B]  push eax  mov eax,[Nn]  push eax  call average  mov ebx,eax  cinvoke printf, text3,eax  cinvoke printf, text  lea eax,[B]  push eax  mov eax,[Nn]  push eax  push ebx  call save  lea ebx,[C]  mov [ebx], dword 34455  mov [ebx+4],dword 12132  mov [ebx+8],dword 4408 ;mov [ebx+12],dword 4  cinvoke printf, text8  prntn C,[NNN]  lea eax,[C] push eax  mov eax,[NNN]  push eax  call average  mov ebx,eax  cinvoke printf, text3,eax  lea eax,[C]  push eax  mov eax,[NNN] push eax  push ebx  call save   ;mov eax,20 ; mov ebx,100 ;imul ebx ;mov ebx,30 ;div ebx ;cinvoke printf, text5,eax invoke sleep,1000000  invoke exit,0  proc average  push ebp  mov ebp, esp  push ebx ush ecx  mov ecx, [ebp+8]  mov ebx, [ebp+12]  xor eax,eax  xor edi,edi  jcxz m4  cycln:  cmp [ebx],esi  jg m5  inc edi  add eax,[ebx]  m5:  add ebx,4  loop cycln  cdq  idiv edi  m4:  pop ecx  pop ebx  pop ebp  ret 8  endp  proc save  push ebp  mov ebp, esp  push ebx  push ecx  mov edx, [ebp+8]  mov ecx, [ebp+12] mov ebx, [ebp+16]  jcxz m1  sev:  cmp [ebx], dword 0  jge m2  mov eax, edx  mov [ebx], eax m2:  add ebx, 4  loop sev  m1:  pop ecx  pop ebx  pop ebp  ret 12  endp  A dd 4 dup(?) N dd 4 B dd 3 dup(?) Nn dd 3 C dd 3 dup(?) NNN dd 3 text db 10, 13, '', 10, 13, 0  text6 db 10,13,'routes Petrov: ',0 text7 db 10,13,'routes Ivanov: ',0 text8 db 10,13,'routes Sidorov: ',0 text3 db 10,13,'sredniy marshrut: %d ',0 text4 db 10,13,'sredniy marshrut: %d ',0 text5 db 10,13,' %d ',0  fmat db '%d',9,0  spp db '%d',0  ; import data in the same section  data import  library msvcrt,'MSVCRT.DLL',\  kernel32,'KERNEL32.DLL'  import kernel32,\  sleep,'Sleep'  import msvcrt,\ printf,'printf',\  scanf,'scanf',\  exit,'exit',\  setlocale,'setlocale',\  system,'system'  end data |

Вывод: Я освоила работу с классами и коллекциями на языке С#, продемонстрировав знания выполнением задания. Я разработала программный кода на языке Assembler для обработки данных.