Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

Специальность 09.02.03 «Программирование в компьютерных системах»

**КВАЛИФИКАЦИОННЫЙ ЭКЗАМЕН**

Профессиональный модуль ПМ.01 Разработка модулей программного обеспечения для компьютерных систем

*(наименование профессионального модуля)*

Выполнил:

обучающийся группы № 3ПКС-420

Д.М. Борушко

*(И.О. Фамилия)*

Проверили:

Г.И. Киреева

*(И.О. Фамилия)*

И.В.Сибирев

*(И.О. Фамилия)*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Москва**

**2022**

Код программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Runtime.InteropServices;

using System.IO;

namespace ConsoleApp3

{

class Student

{

// импорт dll(функция percent\_knowledge)

[DllImport(@"..\..\..\Debug\function.dll", CallingConvention = CallingConvention.Cdecl)]

private static extern double percent\_knowledge(int five\_and\_four\_marks\_count, int num\_of\_student);

// импорт dll(функция sum\_of\_marks)

[DllImport(@"..\..\..\Debug\function.dll", CallingConvention = CallingConvention.Cdecl)]

private static extern int sum\_of\_marks(int[] array\_marks, int size\_array);

// импорт dll(функция avg\_of\_marks)

[DllImport(@"..\..\..\Debug\function.dll", CallingConvention = CallingConvention.Cdecl)]

private static extern double avg\_of\_marks(int[] array\_marks, int size\_array);

private string Surname { get; set; }

private string Marks { get; set; }

private static int count\_student = 0;

private static List<int> AllMarks = new List<int>();

private static List<string> AllSurname = new List<string>();

private static List<string> AllMarksString = new List<string>();

// первый конструктор Студент(Фамилия)

public Student(string surname)

Листинг Program.cs

{

Surname = surname;

AllSurname.Add(Surname);

List<int> marks = new List<int>();

Random random = new Random();

int num\_marks = random.Next(1, 10);

for (int i = 0; i < num\_marks; i++)

{

marks.Add(random.Next(2, 5));

}

Marks = string.Join(", ", marks);

AllMarksString.Add(Marks);

for (int i = 0; i < marks.Count; i++)

{

AllMarks.Add(marks[i]);

}

count\_student++;

}

// первый конструктор Студент(Фамилия, Оценки)

public Student(string surname, string marks)

{

try

{

count\_student++;

Surname = surname;

AllSurname.Add(Surname);

Marks = marks;

AllMarksString.Add(Marks);

var khow\_marks = Marks.Split(',').Select(x => int.Parse(x)).ToList();

for (int i = 0; i < khow\_marks.Count; i++)

{

AllMarks.Add(khow\_marks[i]);

}

Листинг Program.cs

}

catch

{

Console.WriteLine($" В строке \"Оценки\" у {AllSurname[count\_student - 1]} встречены неопознанные символы\n\n");

}

}

// функция, печатающая всю информацию о студенте

public void PrintInfo()

{

Console.WriteLine($" Фамилия: {Surname}\n Оценки: {Marks}\n Сумма оценок: {SumMarks()}\n Средний балл: {AVGMarks()}\n\n\n");

}

// функция, возвращающая процет качества знаний

public static string PercentKhowQuality()

{

int five\_and\_four\_marks\_count = AllMarks.Where(x => x == 5 || x == 4).Count();

double num = percent\_knowledge(five\_and\_four\_marks\_count, count\_student);

string res = "\n Качество знаний: " + num.ToString() + "%";

return res;

}

// функция, возвращающая сумму оценок конкретного студента (Assembler)

// в противном случае (если в вооде оценок не встретилось чисел или встретилось не число в последовательности) возвращает -1

private int SumMarks()

{

try

{

int[] marks = Marks.Split(',').Select(x => int.Parse(x)).ToArray();

return sum\_of\_marks(marks, marks.Length);

Листинг Program.cs

}

catch (Exception)

{

Console.WriteLine(" В строке Оценки встречены неопознанные символы");

return -1;

}

}

// функция, возвращающая среднюю балл конкретного студента (Assembler)

// в противном случае (если в вооде оценок не встретилось чисел или встретилось не число в последовательности) возвращает -1

private double AVGMarks()

{

try

{

int[] marks = Marks.Split(',').Select(x => int.Parse(x)).ToArray();

return avg\_of\_marks(marks, marks.Length);

}

catch (Exception)

{

return -1;

}

}

// функция создающая файл

public static void GetFile()

{

File.Create("Students.txt").Close();

FileStream stream = new FileStream(@"Students.txt", FileMode.Open, FileAccess.Write);

StreamWriter writer = new StreamWriter(stream);

int copy\_i = 0;

try

Листинг Program.cs

{

for (int i = 0; i < count\_student; i++)

{

writer.WriteLine($" Фамилия: {AllSurname[i]}\n Оценки: {AllMarksString[i]}\n Сумма оценок: {sum\_of\_marks(AllMarksString[i].Split(',').Select(x => int.Parse(x)).ToArray(), AllMarksString[i].Split(',').Count())}\n Средний балл: {avg\_of\_marks(AllMarksString[i].Split(',').Select(x => int.Parse(x)).ToArray(), AllMarksString[i].Split(',').Count())}\n\n\n");

copy\_i++;

}

writer.WriteLine(PercentKhowQuality());

}

catch(Exception)

{

writer.WriteLine($" Ошибка при вводе данных у студента {AllSurname[copy\_i]}");

}

writer.Close();

}

}

class Program

{

static void Main(string[] args)

{

// создание коллекции экземляров класса Student

List<Student> students = new List<Student>();

string Name;

for (int i = 1; i < 4; i++)

{

Листинг Program.cs

Console.WriteLine("Студент" + i);

Console.WriteLine("Введите фамилию");

Name = Console.ReadLine();

students.Add(new Student(Name));

};

// вывод на экран информации о каждом студенте

for (int i = 0; i < students.Count; i++)

{

students[i].PrintInfo();

}

// вызов функции, которая создает файл в папке с проектом

Student.GetFile();

// вывод на экран информации о качестве знаний учащихся

Console.WriteLine(Student.PercentKhowQuality());

Console.ReadLine();

}

}

}

Листинг Program.cs

#define function \_declspec(dllexport)

extern "C"

{

function int sum\_of\_marks(int array\_marks[], int size\_array)

{

int sum\_of\_marks = 0;

\_asm

{

Ассемблерная вставка Source.spp

mov eax, size\_array

mov ecx, 0

lea ebx, [array\_marks]

mov edx, [ebx]

add ecx, [edx]

dec eax

cmp eax, 0

jz exit\_

cycle :

dec eax

add edx, 4

add ecx, [edx]

cmp eax, 0

jnz cycle

exit\_ :

mov sum\_of\_marks, ecx

}

return sum\_of\_marks;

}

function double avg\_of\_marks(int array\_marks[], int size\_array)

{

int a, b;

\_asm

{

mov eax, size\_array

mov ecx, 0

lea ebx, [array\_marks]

mov edx, [ebx]

add ecx, [edx]

dec eax

cmp eax, 0

jz exit\_

Ассемблерная вставка Source.spp

cycle :

dec eax

add edx, 4

add ecx, [edx]

cmp eax, 0

jnz cycle

exit\_ :

mov edx, 0

mov eax, ecx

mov ebx, size\_array

div ebx

mov a, eax

mov eax, 100

mul edx

div ebx

mov b, eax

}

return double(a \* 100 + b) / 100;

}

function double percent\_knowledge(int five\_and\_four\_marks\_count, int num\_of\_student)

{

int a, b;

\_asm

{

mov eax, five\_and\_four\_marks\_count

mov edx, 0

mov ebx, num\_of\_student

div ebx

mov a, eax

mov eax, 100

Ассемблерная вставка Source.spp

mul edx

div ebx

mov b, eax

}

return double(a \* 100 + b) / 100;

}

}

Ассемблерная вставка Source.spp

Вывод на консоль:



Рисунок 1. Вывод в консоль

Сохранение в файл:

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2. Сохранение в файл