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»

**ОТЧЕТ**

**ПО КВАЛИФИКАЦИОННОМУ ЭКЗАМЕНУ**

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

*(наименование профессионального модуля)*

Выполнил:

обучающийся учебной группы № 3ПКС-320

Болоцкий Дмитрий Сергеевич

*(И.О. Фамилия)*

Проверил:

руководитель практики от колледжа:

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

*(И.О. Фамилия)*

**Москва**

**2022**

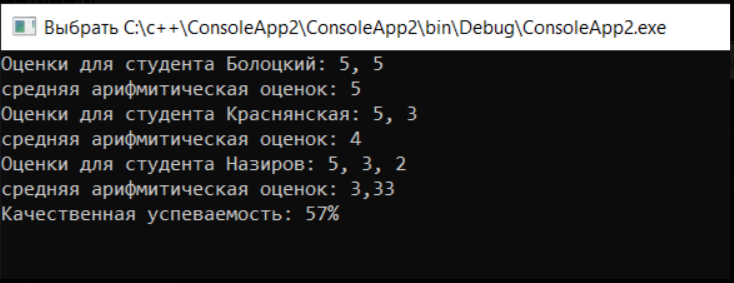
Экзаминационный билет 1

**Цель работы:** Создать класс студент с характеристиками: фамилия, коллекция оценок за практические работы. в данном классе предусмотреть способы создания экземпляров класса вида название класса(фамилия). название класса(фамилия,оценки) (оценки перечислены через запятую и передаются строкой). при создании класса соблюдать принцип инкапсуляции и обработать неправильный ввод данных. сформировать коллекцию с 3 объектами класса. оценки сформировать генератором случайных чисел в диапазоне от 2 до 5 в количестве от 1 до 10. эту информацию вывести на экран с помощью предусмотренного метода. также обработать возникающие исключения.

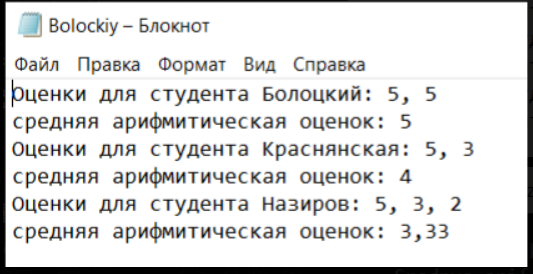
С помощью ассемблерной вставки рассчитать оценок для каждого студента, далее рассчитать среднюю оценку с точностью до 2 знаков после запятой и вывести на экран для каждого студента все оценки студента и среднюю оценку. подсчитать с помощью ассемблера % качества знаний (качественная успеваемость) = [кол-во "отл." + кол-во "хор." (всех студентов)] х 100% / общее кол-во учащихся. вывести на экран эту информацию на экран и сохранить в файле.

фамилии, оценки за 5 работ и среднюю оценку студентов вывести в файл.

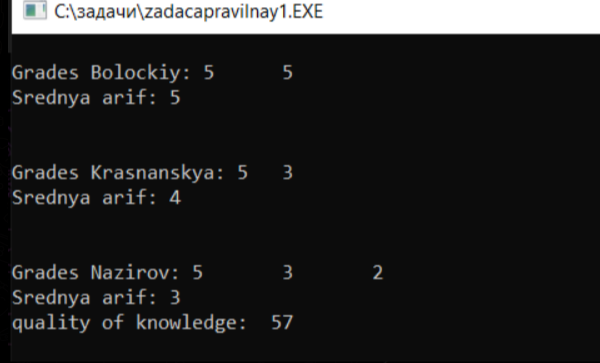
Результат выполнения кода в c#



Результат выполнения кода в c# с выводом в текстовый файл



Результат выполнения кода в c Assembler



Код программы на c#

using Studentt;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

namespace Studentt

{

//создаём класс Student

class Student

{

// поля для хранения фамилии и оцено

private string surname;

private List<int> grades;

// доступ к фамилии

public string Surname { get => surname; }

// доступ к коллекции

public List<int> Grades { get => grades; }

// Конструктор для создания ученика с фамилией

public Student(string surname)

{

this.surname = surname;

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

}

// метод для возврата списка целых чисел через запятую

private List<int> ParseGrades(string grades)

{

try

{

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

}

catch (FormatException)

{

return new List<int>();

}

}

// метод для вычисления суммы всех оценок

public int CalculateTotal()

{

return grades.Sum();

}

// метод расчета средней оценки

public double CalculateAverage()

{

if (grades.Count == 0)

{

return 0;

}

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

}

}

}

class Program

{

static void Main(string[] args)

{

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

students.Add(new Student("Болоцкий"));

students.Add(new Student("Краснянская"));

students.Add(new Student("Назиров"));

Random random = new Random();

foreach (Student student in students)

{

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

{

student.Grades.Add(random.Next(2, 6));

}

}

foreach (Student student in students)

{

Console.WriteLine("Оценки для студента " + student.Surname + ": " + string.Join(", ", student.Grades));

Console.WriteLine("средняя арифмитическая оценок: " + Math.Round(student.CalculateAverage(), 2));

try

{

// вывод оценок и среднего ариф в текстовый документ

FileInfo fileInfo = new FileInfo(@"C:\c++\Bolockiy.txt");

StreamWriter sw = fileInfo.AppendText();

sw.WriteLine("Оценки для студента " + student.Surname + ": " + string.Join(", ", student.Grades));

sw.WriteLine("средняя арифметическая оценок: " + Math.Round(student.CalculateAverage(), 2));

sw.Close();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

// подсчёт количества 5 и 4 среди всех учащихся,а также подсчёт всех оценок

int numFives = students.Sum(student => student.Grades.Count(grade => grade == 5));

int numFours = students.Sum(student => student.Grades.Count(grade => grade == 4));

int numFours1 = students.Sum(student => student.Grades.Count());

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

Console.WriteLine("Качественная успеваемость: " + percentage + "%");

Console.ReadLine();

}

}

Код программы на Assembler

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 5

mov [ebx+4],dword 5

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 5

mov [ebx+4],dword 3

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 5

mov [ebx+4],dword 3

mov [ebx+8],dword 2

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,4

mov ebx,100

imul ebx

mov ebx,7

div ebx

cinvoke printf, text5,eax

invoke sleep,1000000

invoke exit,0

proc average

push ebp

mov ebp, esp

push ebx

push 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 2 dup(?)

N dd 2

B dd 2 dup(?)

Nn dd 2

C dd 3 dup(?)

NNN dd 3

text db 10, 13, '', 10, 13, 0

text6 db 10,13,'Grades Bolockiy: ',0

text7 db 10,13,'Grades Krasnanskya: ',0

text8 db 10,13,'Grades Nazirov: ',0

text3 db 10,13,'Srednya arif: %d ',0

text4 db 10,13,'Srednya arif: %d ',0

text5 db 10,13,'quality of knowledge: %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