## Тема 5.1. Лабораторная работа

## «Стандартные модули и модульная структура приложений в VB»

### 5.1.2. Варианты задания

Таблица 5.1-1

|  |  |
| --- | --- |
|  | Произвели опрос **n** студентов литературного института (**n** - заданное число), каждый из которых назвал две фамилии своих любимых писателей. Составить алгоритм и программу, которые определяют, сколько и какие писатели были названы, и сколько раз был назван каждый из них. Сформировать 2 списка писателей: по алфавиту и по убыванию количества упоминаний студентами. |

### 5.1.3. Содержание отчета

1. Тема и название работы.
2. Задание на разработку проекта и вариант задания.
3. Формализация задания.
4. Разработка приложения:
5. Графический интерфейс пользователя;
6. Таблица свойств объектов;
7. Укрупненная схема алгоритма решения задачи и схемы алгоритмов

функциональных задач;

1. Программный код проекта с использованием процедур функциональных задач

и стандартных модулей.

1. Результаты выполнения приложения.
2. Доказательство правильности работы программы.

### 5.1.4. Примеры выполнения задания

1. *Тема лабораторной работы:*

Стандартные модули и модульная структура приложений в VB – Формирование списка фамилий студентов, которым по итогам сессии необходимо повысить успеваемость до фактического среднего уровня.

1. *Задание на разработку проекта и вариант задания:*

Разработать программный проект, для решения задачи формирования списка фамилий студентов, которым по итогам сессии необходимо повысить успеваемость до фактического среднего уровня, при условии, что для каждого студента известны по 5 оценок, полученных им в сессию, а всего в группе n студентов. Формируемый список фамилий студентов необходимо представить в порядке убывания среднего балла, полученного за сессию.

Исходные списки студентов и результатов сессии могут находиться как в текстовом файле, так и введены с клавиатуры.

Пользовательский интерфейс разрабатываемого приложения для решения поставленной задачи должен содержать как минимум две формы (для ввода и отображения на форме исходных данных, и для вывода результатов на другую форму).

Реализация программного кода должна использовать две формы и стандартный модуль.

*3. Формализация задания:*

Для решения всех вариантов задач необходимо:

* определить входные данные:

n – количество студентов в одной группе (As Integer);

m – количество экзаменов в одной группе (As Integer);

fam(n-1) – одномерный массив фамилий студентов одной группы (As String);

oc(n-1,m) – двумерный массив оценок студентов одной группы (As Integer);

* определить выходные данные:

**средние баллы за сессию** каждого из n студентов одной группы – одномерный массив so(n-1)(As Double);

**общий средний балл** группы sr (As Double);

**фамилии и средние баллы студентов,** для которых средняя оценка so(i)меньше средней по группе sr - одномерный массив f(k-1)(As String) и одномерный массив sb(k-1)(As Double), а также **число таких студентов** k;

* вычислить:

sr - средний балл за сессию студентов одной группы; для получения среднего балла следует накопить сумму баллов, полученных n студентами за m экзаменов, а затем разделить эту сумму на количество оценок m\*n (As Double);

so(n-1) - средний балл за сессию каждого из n студентов одной группы; для получения каждого i-го элемента следует накопить сумму баллов, полученных i-м студентом за m экзаменов, а затем разделить эту сумму на m  
(As Double);

* отсортировать:

баллы (массив sb(k-1)) и соответственно фамилии студентов (массив f(k-1)) в порядке убывания.

Геометрическая иллюстрация решения задачи приведена на рис. 5.1-1.

**Входные данные:**

**Вводимые Рассчитываемые**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Массив фамилий**  **fam( )**  0   |  | | --- | | Шакин | | Сосновиков | | Кравченко | | Загвоздкина | | Минязов | |  | |  |   n-1 | **Массив оценок**  **oc( , )**  0 m-1   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 2 | 3 | 3 | 4 | 2 | | 3 | 3 | 3 | 3 | 3 | | 4 | 4 | 4 | 4 | 4 | | 4 | 5 | 4 | 5 | 4 | | 5 | 5 | 5 | 5 | 5 | |  |  |  |  |  | |  |  |  |  |  |   n-1 | **Средние оценки каждого студента so( )**  0   |  | | --- | |  | |  | |  | |  | |  | |  | |  |   n-1 | **Общий средний балл всей группы**   |  | | --- | |  |   **sr** |

##### Выходные данные:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Массив фамилий студентов с низким средним баллом**  **f ( )**  0   |  | | --- | |  | |  | |  | |  |   k-1 | **Средние баллы, которые ниже общего среднего**  **sb( )**  0   |  | | --- | |  | |  | |  | |  |   k-1 |

Рис. 5.1-1

***1 вариант реализации проекта:***

**В программном коде проекта используется стандартный модуль и две формы; исходные данные вводятся с клавиатуры и отображаются на 1-ой форме; результаты отображаются на 2-ой форме.**

*4. Разработка проекта:*

1. **Разработка графического интерфейса пользователя**

Формы интерфейса пользователя приведены на рис. 5.1-2 и рис. 5.1-3.

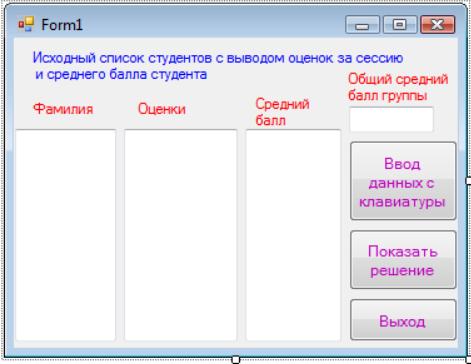


Рис. 5.1-2

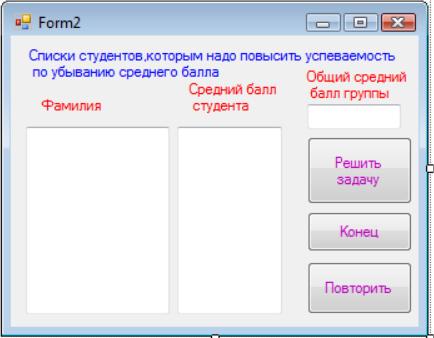


Рис. 5.1-3

1. **Установка свойств объекта**

Свойства объектов управления разработанных форм студенты устанавливают самостоятельно и отображают в таблице.

1. **Разработка схем алгоритмов**

В соответствии с заданием **1**-го варианта реализации проекта, решение задачи можно разбить на следующие функциональные части:

* + ввод количества студентов в группе – n;
  + ввод количества экзаменов в группе – m;
  + ввод исходных данных с клавиатуры – vvod(n,m, fam, oc);
  + отображение массивов данных (вещественного и строкового) и матрицы с оценками в текстовом поле на форме – процедуры vivodMas(),vivodMasStr() и vivodMatr();
  + нахождение sr - общего среднего балла за сессию студентов одной группы и so(i)-среднего балла каждого студента – sred(n,m,oc,so, sr);
  + решение задачи: подсчет количества k и формирование списка фамилий студентов f(), для которых средняя оценка меньше общей средней в группе, а также формирование массива sb()из их средних оценок – FormirSpisok(n,sr, so,fam, f, sb, k);
  + сортировка массива sb() в порядке убывания с одновременной перестановкой фамилий студентов – sort(f ,sb);

Исходные данные n- количество студентов,m - количество экзаменов,  
fam()-массив фамилий студентов,oc(,) – массив их оценок, и вычисленные из них значение общего среднего балла группы sr и массива средних оценок каждого студента so()должны быть доступны обеим формам проекта, поэтому эти данные объявляются как Public в стандартном модуле Module1. Там же в стандартном модуле Module1 описываются процедуры, которые могут вызываться из обеих форм проекта, а именно: процедуры вывода массивов и матрицы. Результирующие списки студентов, которым надо повысить успеваемость (т.е. количество таких студентов k, массивы их фамилий f()и средних баллов sb()) нужны только для второй формы, поэтому объявляются на ней в соответствующей событийной процедуре.

Так как решить задачу можно только после ввода исходных данных, на первой форме кнопку для перехода на вторую форму Button3 («Показать решение») сначала делаем недоступной (свойство Enable = False). После ввода данных и расчета средних оценок, в конце событийной процедуры кнопки Button1, изменяем для кнопки Button3 свойство Enable на True. Для отображения на форме исходных данных в процедурах vivodMas(),vivodMasStr()и vivodMatr()формируется строка z, содержащая элементы соответствующих массивов. Эта строка должна возвращаться из процедур и ее значение выводится в нужные текстовые поля*.* Аналогичным образом происходит и отображение результатов на 2-ой форме.

Общую (укрупненную) схему алгоритма решения задачи и схемы алгоритмов отдельных функциональных задач студенты должны разработать самостоятельно.

1. **Программный код:**

Программный код решаемой задачи, реализованный с использованием процедур, стандартного модуля и двух форм, представлен на рис. 5.1-4.

|  |
| --- |
| **Option Strict On**  **Option Explicit On**  **Module Module1**  **Public n As Integer ' Количество студентов**  **Public m As Integer ' Количество экзаменов**  **Public oc(,) As Integer ' Оценки, полученные в сессию**  **Public fam() As String ' Фамилии студентов**  **Public sr As Double ' Общий средний балл группы за сессию**  **Public so() As Double ' Средние баллы каждого студента**  **' Определение общего среднего балла всей группы sr**  **' и массива средних баллов каждого студента so(i)**  **Public Sub sred(ByVal n As Integer, ByVal m As Integer, \_**  **ByRef oc(,) As Integer, ByRef so() As Double, \_**  **ByRef As Double)**  **Dim i, j As Integer**  **Dim s As Double**  **sr = 0**  **For i = 0 To n - 1**  **s = 0**  **For j = 0 To m - 1**  **s = s + oc(i, j)**  **Next j**  **so(i) = s / m**  **sr = sr + s**  **Next i**  **sr = sr / (m \* n)**  **End Sub**  **'Процедура вывода строкового массива данных**  **Public Sub VivodMasStr(ByVal a() As String, ByRef z As String)**  **Dim i As Integer**  **z = ""**  **For i = 0 To UBound(a)**  **z = z + a(i) + vbNewLine**  **Next**  **End Sub**  **'Процедура вывода вещественного массива данных**  **Public Sub VivodMas(ByVal a() As Double, ByRef z As String)**  **Dim i As Integer**  **z = ""**  **For i = 0 To UBound(a)**  **z = z + Format(a(i), "0.0000") + vbNewLine**  **Next**  **End Sub**  **'Процедура вывода двумерного массива**  **'(собираем матрицу в одну строку)**  **Sub vivodMatr(ByRef x(,) As Integer, ByVal n As Integer, \_**  **ByVal m As Integer, ByRef z As String)**  **Dim i, j As Integer**  **z = ""**  **For i = 0 To n - 1**  **For j = 0 To m - 1**  **z = z + CStr(x(i, j)) & Space(3)**  **Next j**  **z = z + vbNewLine**  **Next i**  **End Sub**  **End Module**  **'на этой форме вводятся и выводятся исходные данные о студентах**  **Option Strict On**  **Public Class Form1**  **'Процедура ввода данных с клавиатуры**  **Public Sub vvod(ByVal n As Integer, ByVal m As Integer, \_**  **ByRef fam() As String, ByRef oc(,) As Integer)**  **Dim i, j As Integer**  **For i = 0 To n - 1**  **fam(i) = InputBox("Фамилия")**  **For j = 0 To m - 1**  **oc(i, j) = CInt(InputBox("Оценка"))**  **Next j**  **Next i**  **End Sub**  **'кнопка для ввода с клавиатуры**  **Private Sub Button1\_Click(ByVal sender As System.Object, \_**  **ByVal e As System.EventArgs) \_**  **Handles Button1.Click**  **Dim s As String = ""**  **n = CInt(InputBox("Введите количество студентов"))**  **m = CInt(InputBox("Введите количество экзаменов"))**  **ReDim fam(n - 1), oc(n - 1, m - 1)**  **'вызов процедуры ввода группы с клавиатуры**  **vvod(n, m, fam, oc)**  **ReDim so(n - 1)**  **'вызов процедуры вычисления среднего балла каждого студента**  **'и общего среднего балла группы**  **sred(n, m, oc, so, sr)**  **VivodMasStr(fam, s)**  **TextBox1.Text = s**  **vivodMatr(oc, n, m, s)**  **TextBox2.Text = s**  **VivodMas(so, s)**  **TextBox3.Text = s**  **TextBox4.Text = Format(sr, "0.0000")**  **'Кнопку3 делаем доступной после ввода данных и расчета**  **'средних баллов**  **Button3.Enabled = True**  **End Sub**  **'Процедура для продолжения работы программы(перехода на 2-ю форму)**  **Private Sub Button3\_Click(ByVal sender As System.Object, \_**  **ByVal e As System.EventArgs) \_**  **Handles Button3.Click**  **Form2.Show()**  **End Sub**  **Private Sub Button2\_Click(ByVal sender As System.Object, \_**  **ByVal e As System.EventArgs) \_**  **Handles Button2.Click**  **End**  **End Sub**  **End Class**  **Public Class Form2**  **' Формирование массива f() фамилий студентов,**  **' имеющих средний балл ниже общего среднего**  **' и массива sb() средних баллов этих студентов**  **Public Sub FormirSpisok(ByVal n As Integer,ByVal sr As Double, \_**  **ByVal so() As Double,ByVal fam() As String, \_**  **ByRef f() As String, ByRef sb() As Double,ByRef k As Integer)**  **Dim i As Integer**  **k = 0**  **For i = 0 To n - 1**  **If so(i) < sr Then**  **f(k) = fam(i)**  **sb(k) = so(i)**  **k = k + 1**  **End If**  **Next**  **ReDim Preserve f(k - 1), sb(k - 1)**  **End Sub**  **'сортировка по убыванию среднего балла студента**  **Sub sort(ByRef s() As String, ByRef kol() As Double)**  **Dim i, j, k As Integer, temp As String, t As Double**  **k = s.Length - 1**  **For i = 0 To k - 1**  **For j = i + 1 To k**  **If kol(i) < kol(j) Then**  **t = kol(i) : kol(i) = kol(j) : kol(j) = t**  **temp = s(i) : s(i) = s(j) : s(j) = temp**  **End If**  **Next**  **Next**  **End Sub**  **Private Sub Button1\_Click(ByVal sender As System.Object, \_**  **ByVal e As System.EventArgs)Handles Button1.Click**  **Dim z As String = "" 'строка для вывода массивов**  **Dim k As Integer 'кол-во студ,к-рым надо повысить успеваем**  **Dim f(n - 1) As String 'массив их фамилий**  **Dim sb(n - 1) As Double 'массив их средних баллов**  **FormirSpisok(n, sr, so, fam, f, sb, k)**  **sort(f, sb)**  **VivodMasStr(f, z)**  **TextBox1.Text = z**  **VivodMas(sb, z)**  **TextBox2.Text = z**  **TextBox3.Text = Format(sr, "0.0000")**  **End Sub**  **' Кнопка для повторения**  **Private Sub Button4\_Click(ByVal sender As System.Object, \_**  **ByVal e As System.EventArgs) Handles Button4.Click**  **Me.Close()**  **Form1.TextBox1.Text = ""**  **Form1.TextBox2.Text = ""**  **Form1.TextBox3.Text = ""**  **Form1.TextBox4.Text = ""**  **Form1.Button1.Enabled = True**  **Form1.Button2.Enabled = True**  **Form1.Button3.Enabled = False**  **Form1.Show()**  **End Sub**  **End Class** |

Рис. 5.1-4

*5. Выполнение приложения и получение результатов:*

Результаты работы проекта **(вариант 1)** представлены на рис. 5.1-5 и рис. 5.1.6.

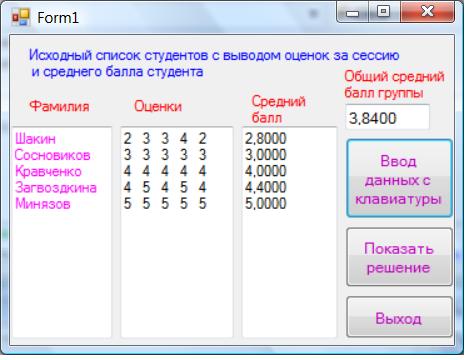


Рис. 5.1-5

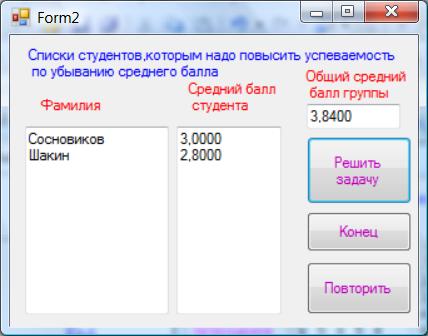


Рис. 5.1-6

*6. Доказательство правильности работы программы*

В результате работы проекта с клавиатуры введены исходные данные (массив фамилий студентов и матрица полученных в сессию оценок), которые использованы для подсчета средней оценки каждого студента и среднего балла всей группы и выведены на 1 форму. На второй форме представлен сформированный и отсортированный по убыванию среднего балла список фамилий тех студентов, которым надо повысить успеваемость до фактического среднего уровня (у которых средняя оценка ниже общего среднего балла всей группы).

***2 вариант реализации проекта:***

**В программном коде проекта используется стандартный модуль и две формы; исходные данные вводятся из текстового файла чтением построчно и отображаются на первой форме; результаты вычислений отображаются на второй форме и построчно записываются в другой текстовый файл.**

Формат исходного файла: каждая строка файла состоит из фамилии и **5** оценок, разделенных между собой одним или несколькими пробелами, например,

**Фамилия 5 4 5 3 4.**

1. *Разработка проекта:*
   1. **Разработка графического интерфейса пользователя**

Разработанная форма интерфейса приведена на рис. 5.1-7 и рис. 5.1-8.

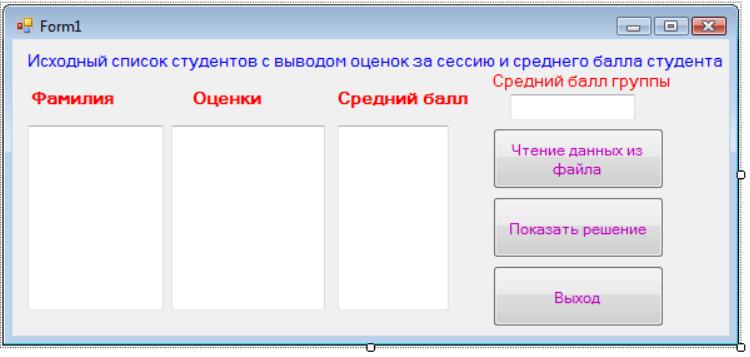


Рис. 5.1-7

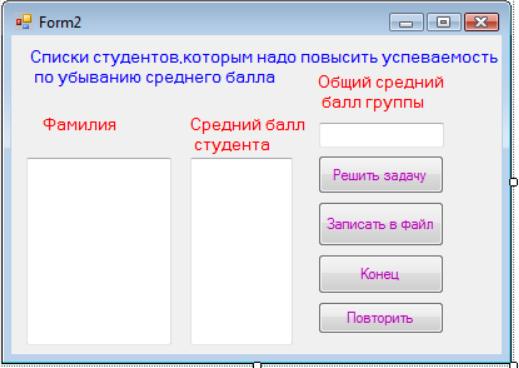


Рис. 5.1-8

* 1. **Установка свойств объектов**

Свойства объектов управления разработанной формы студенты устанавливают самостоятельно и отображают в таблице.

* 1. **Разработка схемы алгоритма**

В соответствии с заданием ***2-го*** варианта реализации проекта, решение задачи можно разбить на следующие функциональные задачи:

* + построчное чтение исходных данных из текстового файла и определение количества студентов в группе – readfile(“путь1”, n);
  + определение кол-ва экзаменов и создание массива фамилий студентов fam()и массива оценок oc()из читаемого файла – vvod(“путь1”, n, m,fam, oc);
  + отображение массивов данных (вещественного и строкового) и матрицы с оценками в текстовом поле на форме – процедуры vivodMas(), vivodMasStr() и vivodMatr();
  + нахождение sr - общего среднего балла за сессию студентов одной группы и so(i)- среднего балла каждого студента – sred(n, m, oc, so, sr);
  + решение задачи: подсчет количества k и формирование списка фамилий студентов f(), для которых средняя оценка меньше общей средней в группе, а также формирование массива sb() из их средних оценок –   
    FormirSpisok(n, sr, so, fam, f, sb, k);
  + сортировка массива sb() в порядке убывания с одновременной перестановкой фамилий студентов – sort(f ,sb);
  + построчная запись результирующих массивов f(),и sb() в текстовый файл – writefile(“путь2”,f, sb).

Исходные данные n - количество студентов, m - количество экзаменов, fam()-массив фамилий студентов, oc(,) – массив их оценок, и вычисленные из них значение общего среднего балла группы sr и массива средних оценок каждого студента so()должны быть доступны обеим формам проекта, поэтому эти данные объявляются как Public в стандартном модуле Module1. Там же в стандартном модуле Module1 описываются процедуры, которые могут вызываться из обеих форм проекта, а именно: процедуры вывода массивов и матрицы. Результирующие списки студентов, которым надо повысить успеваемость (т.е. количество таких студентов k, массивы их фамилий f()и средних баллов sb()) нужны только для второй формы, но для двух кнопок (событийных процедур), поэтому объявляются на второй форме, как глобальные для нее (перед всеми остальными процедурами).

Так как решить задачу можно только после ввода исходных данных, на первой форме кнопку для перехода на вторую форму Button3 («Показать решение») сначала делаем недоступной (свойство **Enable = False**). После ввода данных и расчета средних оценок, в конце событийной процедуры для кнопки Button1, изменяем для кнопки Button3 свойство **Enable** на **True**. Для отображения на форме массивов в процедурах vivodMas(), vivodMasStr() и vivodMatr() формируется строка z, содержащая элементы соответствующих массивов. Эта строка должна возвращаться из процедур и ее значение выводится в нужные текстовые поля.

Общую (укрупненную) схему алгоритма решения задачи и схемы алгоритмов отдельных функциональных задач студенты должны разработать самостоятельно.

* 1. **Написание программного кода с использованием модуля и процедур**

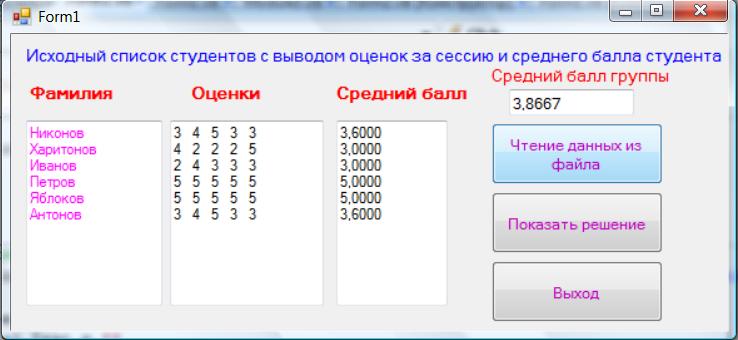
Программный код, решаемой задачи, реализованный с использованием процедур, стандартного модуля и двух форм, представлен на рис. 5.1-9.

|  |
| --- |
| **Option Strict On**  **Option Explicit On**  **Module Module1**  **Public n As Integer ' Количество студентов**  **Public m As Integer ' Количество экзаменов**  **Public oc(,) As Integer ' Оценки, полученные в сессию**  **Public fam() As String ' Фамилии студентов**  **Public sr As Double ' Общий средний балл группы за сессию**  **Public so() As Double ' Средние баллы каждого студента**  **' Определение общего среднего балла всей группы sr**  **' и массива средних баллов каждого студента so(i)**  **Public Sub sred(ByVal n As Integer, ByVal m As Integer, \_**  **ByRef oc(,) As Integer, ByRef so() As Double, \_**  **ByRef sr As Double)**  **Dim i, j As Integer**  **Dim s As Double**  **sr = 0**  **For i = 0 To n - 1**  **s = 0**  **For j = 0 To m - 1**  **s = s + oc(i, j)**  **Next j**  **so(i) = s / m**  **sr = sr + s**  **Next i**  **sr = sr / (m \* n)**  **End Sub**  **'Процедура вывода строкового массива данных**  **Public Sub VivodMasStr(ByVal a() As String, ByRef z As String)**  **Dim i As Integer**  **z = ""**  **For i = 0 To UBound(a)**  **z = z + a(i) + vbNewLine**  **Next**  **End Sub**  **'Процедура вывода вещественного массива данных**  **Public Sub VivodMas(ByVal a() As Double, ByRef z As String)**  **Dim i As Integer**  **z = ""**  **For i = 0 To UBound(a)**  **z = z + Format(a(i), "0.0000") + vbNewLine**  **Next**  **End Sub**  **'Процедура вывода двумерного массива**  **'(собираем матрицу в одну строку)**  **Sub vivodMatr(ByRef x(,) As Integer, ByVal n As Integer, \_**  **ByVal m As Integer, ByRef z As String)**  **Dim i, j As Integer**  **z = ""**  **For i = 0 To n - 1**  **For j = 0 To m - 1**  **z = z + CStr(x(i, j)) & Space(3)**  **Next j**  **z = z + vbNewLine**  **Next i**  **End Sub**  **End Module**  **'на 1-й форме вводятся и выводятся исходные данные о студентах**  **Option Strict On**  **Imports System.IO**  **Public Class Form1**  **'Процедура построчного чтения сведений о студентах из тек. файла,**  **'для подсчета числа студентов**  **Sub readfile(ByVal filename As String, ByRef n As Integer)**  **'создание потока для чтения из текстового файла**  **Dim f = New StreamReader(filename)**  **Dim line As String**  **n = 0 'количество строк в файле**  **Try**  **Do While Not (f.EndOfStream)'пока не достигнут конец потока**  **line = f.ReadLine() 'чтение строки из файла в line**  **n = n + 1**  **Loop**  **f.Close() 'закрытие потока**  **Catch ex As Exception**  **MsgBox("Невозможно прочитать данные из файла")**  **End Try**  **End Sub**  **'Процедура построчного считывания сведений о студентах из файла**  **'и создание массива фамилий студентов и массива оценок**  **Sub vvod(ByVal filename As String,ByVal n As Integer, \_**  **ByRef m As Integer, ByRef fam() As String, \_**  **ByRef oc(,) As Integer)**  **Dim i, j As Integer**  **Dim line As String**  **Dim buf() As String**  **'создание потока для чтения из текстового файла**  **Dim f As New StreamReader(filename)**  **Try**  **i = 0 'номер читаемой строки в файле**  **'пока не достигнут конец потока**  **Do While Not (f.EndOfStream)**  **line = f.ReadLine() 'чтение строки из файла в line**  **'Удаление из строки возможных лишних пробелов**  **'(цикл замены двух пробелов одним)**  **Do While line.IndexOf(Space(2)) >= 0**  **line = line.Replace(Space(2), Space(1))**  **Loop**  **If line = "" Then**  **n = n - 1**  **MsgBox("В файле есть пустая строка")**  **Exit Sub**  **Else**  **'Ф-ция Split разбивает строку line на массив**  **'подстрок buf()по символам пробела**  **buf = Split(line, Space(1))**  **'===================================================================**  **' При обработке первой строки файла(при i = 0)определяем**  **' кол-во экзаменов m и выделяем память массив оценок oc(,)**  **If i = 0 Then**  **'длина массива buf() -кол-во экзаменов минус фамилия**  **m = buf.Length - 1**  **ReDim oc(n - 1, m - 1)**  **End If '==================================================================**  **'Первый элемент этого массива buf()-фамилия,следующие-оценки**  **fam(i) = buf(0)**  **For j = 0 To m - 1**  **oc(i, j) = CInt(buf(j + 1))**  **Next**  **End If**  **i = i + 1**  **Loop**  **f.Close()**  **Catch ex As Exception**  **MsgBox("Невозможно прочитать данные из файла")**  **End Try**  **n = i**  **End Sub**  **'кнопка для ввода из файла**  **Private Sub Button1\_Click(ByVal sender As System.Object, \_**  **ByVal e As System.EventArgs) \_**  **Handles Button1.Click**  **Dim s As String = ""**  **'Вызов проц-ры построчного чтения из текстового файла**  **'для определения количества студентов n**  **readfile("myfile.txt", n)**  **ReDim fam(n - 1)**  **'вызов процедуры ввода группы из файла(создание массивов**  **'фамилий студентов и массива оценок)**  **vvod("myfile.txt", n, m, fam, oc)**  **ReDim so(n - 1)**  **'вызов процедуры вычисления среднего балла каждого студента**  **'и общего среднего балла группы**  **sred(n, m, oc, so, sr)**  **'вывод данных на форму**  **VivodMasStr(fam, s)**  **TextBox1.Text = s**  **vivodMatr(oc, n, m, s)**  **TextBox2.Text = s**  **VivodMas(so, s)**  **TextBox3.Text = s**  **TextBox4.Text = Format(sr, "0.0000")**  **'Кнопку3 делаем доступной после ввода данных**  **'и расчета средних баллов**  **Button3.Enabled = True**  **End Sub**  **'кнопка для перехода на 2-ю форму**  **Private Sub Button3\_Click(ByVal sender As System.Object, \_**  **ByVal e As System.EventArgs) \_**  **Handles Button3.Click**  **Form2.Show()**  **End Sub**  **'Процедура для окончания работы программы**  **Private Sub Button2\_Click(ByVal sender As System.Object, \_**  **ByVal e As System.EventArgs) \_**  **Handles Button2.Click**  **End**  **End Sub**  **End Class**  **'Программный код 2 формы**  **Option Strict On**  **Imports System.IO**  **Public Class Form2**  **'Объекты k,f(n-1)и sb(n-1)объяввляем глобальными для 2-й формы,**  **'т.к. они должны быть доступны сразу двум событийным процедурам**  **'(для кнопок "Решить задачу" и "Записать в файл")**  **Dim k As Integer 'кол-во студ, к-рым надо повысить успеваемость**  **Dim f(n - 1) As String 'массив их фамилий**  **Dim sb(n - 1) As Double 'массив их средних баллов**  **' Формирование массива f() фамилий студентов,**  **' имеющих средний балл ниже общего среднего**  **' и массива sb() средних баллов этих студентов**  **Public Sub FormirSpisok(ByVal n As Integer, ByVal sr As Double, \_**  **ByVal so() As Double, ByVal fam() As String, \_**  **ByRef f() As String, ByRef sb() As Double, \_**  **ByRef k As Integer)**  **Dim i As Integer**  **k = 0**  **For i = 0 To n - 1**  **If so(i) < sr Then**  **f(k) = fam(i)**  **sb(k) = so(i)**  **k = k + 1**  **End If**  **Next**  **ReDim Preserve f(k - 1), sb(k - 1)**  **End Sub**  **'сортировка по убыванию среднего балла студента**  **Sub sort(ByRef s() As String, ByRef kol() As Double)**  **Dim i, j, k As Integer, temp As String, t As Double**  **k = s.Length - 1**  **For i = 0 To k - 1**  **For j = i + 1 To k**  **If kol(i) < kol(j) Then**  **t = kol(i) : kol(i) = kol(j) : kol(j) = t**  **temp = s(i) : s(i) = s(j) : s(j) = temp**  **End If**  **Next**  **Next**  **End Sub**  **' Процедура построчной записи в текстовый файл отсортированного**  **' массива студентов,имеющих средний балл ниже общего среднего**  **Sub writefile(ByVal filename As String, ByVal f() As String, \_**  **ByVal sb() As Double)**  **'создание нового текст. файла и потока wf для записи в файл**  **Dim wf As StreamWriter = File.CreateText(filename)**  **Dim s As String 'строка для записи в файл**  **Dim i, k As Integer**  **k = f.Length - 1**  **For i = 0 To k**  **s = fam(i) + " " + CStr(sb(i))**  **'запись сведений об одном студенте в текстовый файл**  **wf.WriteLine(s)**  **Next**  **wf.Flush() : wf.Close() 'очистка и закрытие потока**  **End Sub**  **'кнопка "Решить задачу"**  **Private Sub Button1\_Click(ByVal sender As System.Object, \_**  **ByVal e As System.EventArgs) \_**  **Handles Button1.Click**  **Dim z As String = "" 'строка для вывода массивов**  **FormirSpisok(n, sr, so, fam, f, sb, k)**  **sort(f, sb)**  **VivodMasStr(f, z)**  **TextBox1.Text = z**  **VivodMas(sb, z)**  **TextBox2.Text = z**  **TextBox3.Text = Format(sr, "0.0000")**  **End Sub**  **'кнопка "Записать в файл"**  **Private Sub Button2\_Click(ByVal sender As System.Object, \_**  **ByVal e As System.EventArgs)\_**  **Handles Button2.Click**  **writefile("file2.txt", f, sb)**  **End Sub**  **'Процедура для окончания работы программы**  **Private Sub Button3\_Click(ByVal sender As System.Object, \_**  **ByVal e As System.EventArgs) \_**  **Handles Button3.Click**  **End**  **End Sub**  **'кнопка "Повторить"**  **Private Sub Button4\_Click(ByVal sender As System.Object, \_**  **ByVal e As System.EventArgs) \_**  **Handles Button4.Click**  **Me.Close()**  **Form1.TextBox1.Text = ""**  **Form1.TextBox2.Text = ""**  **Form1.TextBox3.Text = ""**  **Form1.TextBox4.Text = ""**  **Form1.Button1.Enabled = True**  **Form1.Button2.Enabled = True**  **Form1.Button3.Enabled = False**  **Form1.Show()**  **End Sub**  **End Class** |

Рис. 5.1-9.

1. *Выполнение приложения и получение результатов:*

Результаты работы проекта **Проект-5-1-Лаб-2 (вариант 2)** представлены на рис. 5.1-10 и рис. 5.1.11.



|  |
| --- |
| Рис. 5.1-10 |

Рис. 5.1-11

1. *Доказательство правильности работы программ.*

В результате работы проекта из файла построчнопрочитаны исходные данные и корректно преобразованы в массивы, которые использованы для подсчета средней оценки каждого студента и среднего балла всей группы и выведены на 1 форму. На второй форме представлены полученные и отсортированные по убыванию среднего балла массив фамилий и массив среднего балла тех студентов, которым надо повысить успеваемость до фактического среднего уровня (у которых средняя оценка ниже общего среднего балла всей группы). Полученные результаты построчно записаны в другой текстовый файл.

***3 вариант реализации проекта:***

**В программном коде проекта используется стандартный модуль и две формы; исходные данные вводятся из текстового файла чтением целиком в одну строку и отображаются на первой форме; результаты вычислений отображаются на второй форме и формируются в одну строку, которая записывается в другой текстовый файл.**

Формат исходного файла: допускается разделение данных в строке несколькими пробелами, т.е. может иметь формат вида: **фамилия 5 4 5 3 4.**

1. *Разработка проекта:*
   1. **Разработка графического интерфейса пользователя**

Разработанная форма интерфейса приведена на рис. 5.1-12 и рис. 5.1-13.

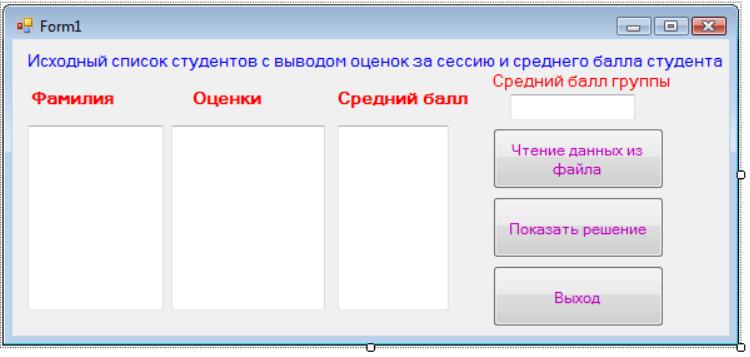


Рис. 5.1-12

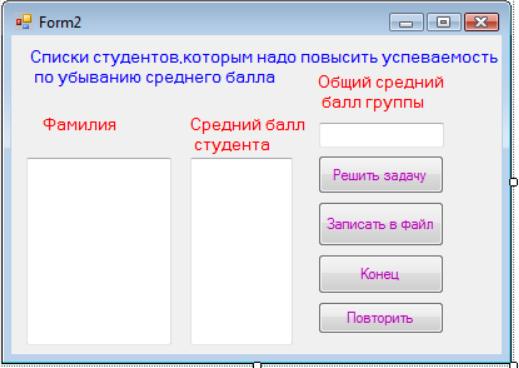


Рис. 5.1-13

* 1. **Установка свойств объектов**

Свойства объектов управления разработанной формы студенты устанавливают самостоятельно и отображают в таблице.

* 1. **Разработка схемы алгоритма**

В соответствии с заданием ***3-го*** варианта реализации проекта, решение задачи можно разбить на следующие функциональные задачи:

* + ввод исходных данных из текстового файла одной строкой – vvodFile(“путь1”, s);
  + получение из этой строки массива подстрок mas() с помощью функции Split()и определение количества студентов n;
  + определение кол-ва экзаменов и создание массива фамилий студентов fam()и массива оценок oc()из полученного массива mas() подстрок файла – vvod(mas, n, m,fam, oc);
  + отображение массивов данных (вещественного и строкового) и матрицы с оценками в текстовом поле на форме – процедуры vivodMas(), vivodMasStr() и vivodMatr();
  + нахождение sr -общего среднего балла за сессию студентов одной группы и so(i)- среднего балла каждого студента – sred(n, m, oc, so, sr);
  + решение задачи: подсчет количества k и формирование списка фамилий студентов f(), для которых средняя оценка меньше общей средней в группе, а также формирование массива sb() из их средних оценок – FormirSpisok(n, sr, so, fam, f, sb, k);
  + сортировка массива sb() в порядке убывания с одновременной перестановкой фамилий студентов – sort(f ,sb);
  + формирование строки S из результирующих массивов f()и sb()– функция  
    res(f, sb, k);
  + запись строки результата в текстовый файл – vivod("путь2", S)

Исходные данные n - количество студентов,m - количество экзаменов,fam()-массив фамилий студентов,oc(,) – массив их оценок, и вычисленные из них значение общего среднего балла группы sr и массива средних оценок каждого студента so()должны быть доступны обеим формам проекта, поэтому эти данные объявляются как Public в стандартном модуле Module1. Там же в стандартном модуле Module1 описываются процедуры, которые могут вызываться из обеих форм проекта, а именно: процедуры вывода массивов и матрицы. Результирующие списки студентов, которым надо повысить успеваемость (т.е. количество таких студентов k, массивы их фамилий f()и средних баллов sb()) нужны только для второй формы, но для двух кнопок (событийных процедур), поэтому объявляются на второй форме, как глобальные для нее (перед всеми остальными процедурами).

Так как решить задачу можно только после ввода исходных данных, на первой форме кнопку для перехода на вторую форму Button3 («Показать решение») сначала делаем недоступной (свойство **Enable = False**). После ввода данных и расчета средних оценок, в конце событийной процедуры для кнопки Button1, изменяем для кнопки Button3 свойство **Enable** на **True**. Для отображения на форме массивов в процедурах vivodMas(), vivodMasStr() и vivodMatr() формируется строка z, содержащая элементы соответствующих массивов. Эта строка должна возвращаться из процедур и ее значение выводится в нужные текстовые поля.

Общую схему алгоритма решения задачи и схемы алгоритмов отдельных функциональных задач студенты должны разработать самостоятельно.

* 1. **Написание программного кода с использованием модуля и процедур**

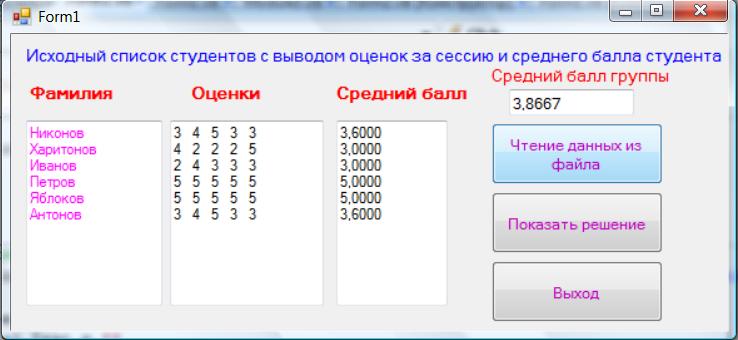
Программный код проекта, представлен на рис. 5.1-14.

|  |
| --- |
| **Option Strict On**  **Option Explicit On**  **Module Module1**  **Public n As Integer ' Количество студентов**  **Public m As Integer ' Количество экзаменов**  **Public oc(,) As Integer ' Оценки, полученные в сессию**  **Public fam() As String ' Фамилии студентов**  **Public sr As Double ' Общий средний балл группы за сессию**  **Public so() As Double ' Средние баллы каждого студента**  **' Определение общего среднего балла всей группы sr**  **' и массива средних баллов каждого студента so(i)**  **Public Sub sred(ByVal n As Integer, ByVal m As Integer, \_**  **ByRef oc(,) As Integer, ByRef so() As Double, \_**  **ByRef sr As Double)**  **Dim i, j As Integer**  **Dim s As Double**  **sr = 0**  **For i = 0 To n - 1**  **s = 0**  **For j = 0 To m - 1**  **s = s + oc(i, j)**  **Next j**  **so(i) = s / m**  **sr = sr + s**  **Next i**  **sr = sr / (m \* n)**  **End Sub**  **'Процедура вывода строкового массива данных**  **Public Sub VivodMasStr(ByVal a() As String, ByRef z As String)**  **Dim i As Integer**  **z = ""**  **For i = 0 To UBound(a)**  **z = z + a(i) + vbNewLine**  **Next**  **End Sub**  **'Процедура вывода вещественного массива данных**  **Public Sub VivodMas(ByVal a() As Double, ByRef z As String)**  **Dim i As Integer**  **z = ""**  **For i = 0 To UBound(a)**  **z = z + Format(a(i), "0.0000") + vbNewLine**  **Next**  **End Sub**  **'Процедура вывода двумерного массива**  **'(собираем матрицу в одну строку)**  **Sub vivodMatr(ByRef x(,) As Integer, ByVal n As Integer, \_**  **ByVal m As Integer, ByRef z As String)**  **Dim i, j As Integer**  **z = ""**  **For i = 0 To n - 1**  **For j = 0 To m - 1**  **z = z + CStr(x(i, j)) & Space(3)**  **Next j**  **z = z + vbNewLine**  **Next i**  **End Sub**  **End Module**  **'на 1-й форме вводятся и выводятся исходные данные о студентах**  **Option Strict On**  **Public Class Form1**  **'Процедура чтения сведений о студентах из текстового файла**  **'в одну строку**  **Sub vvodFile(ByVal filename As String, ByRef S As String)**  **' Открываем существующий файл и загружаем его текстовое**  **'содержимое целиком в одну строку S.**  **Try**  **S = My.Computer.FileSystem.ReadAllText(filename)**  **Catch ex As Exception**  **MsgBox("ошибка при чтении файла")**  **End Try**  **End Sub**  **'Процедура ввода данных из файла(создание массива фамилий ‘студентов и массива оценок из массива строк файла)**  **Public Sub vvod(ByVal mas() As String, ByVal n As Integer, \_**  **ByRef m As Integer, ByRef fam() As String, \_**  **ByRef oc(,) As Integer)**  **Dim i, j As Integer**  **Dim line As String**  **Dim buf() As String**  **For i = 0 To n - 1**  **line = mas(i)**  **'Удаление из строки возможных лишних пробелов**  **'(цикл замены двух пробелов одним)**  **Do While line.IndexOf(Space(2)) >= 0**  **line = line.Replace(Space(2), Space(1))**  **Loop**  **If mas(i) = "" Then**  **n = n - 1**  **MsgBox("В файле есть пустая строка")**  **Exit For**  **Else**  **'Ф-ция Split разбивает строку line на массив**  **'подстрок buf()по символам пробела**  **buf = Split(line, Space(1))**  **'=================================================================**  **' При обработке первой строки файла(при i = 0)определяем**  **' кол-во экзаменов m и выделяем память массив оценок oc(,)**  **If i = 0 Then**  **'длина массива buf() -кол-во экзаменов минус фамилия**  **m = buf.Length - 1**  **ReDim oc(n - 1, m - 1)**  **End If**  **'================================================================**  **'Первый элемент этого массива-фамилия,следующие-оценки**  **fam(i) = buf(0)**  **For j = 0 To m - 1**  **oc(i, j) = CInt(buf(j + 1))**  **Next**  **End If**  **Next**  **End Sub**  **'кнопка для ввода исходных данных из файла и**  **'создания массивов исходных данных**  **Private Sub Button1\_Click(ByVal sender As System.Object, \_**  **ByVal e As System.EventArgs) \_**  **Handles Button1.Click**  **Dim s As String = ""**  **'Вызов проц-ры чтения сведений о студентах из текстового**  **'файла целиком в одну строку s**  **vvodFile("myfile.txt", s)**  **Dim mas() As String**  **'Ф-ция Split разбивает строку s на массив подстрок mas()**  **'по символам конца строки vbNewLine**  **mas = Split(s, vbNewLine)**  **n = mas.Length'кол-во эл-тов получ. массива(кол-во студентов)**  **'По полученному количеству студентов n**  **'переопределяем размер массивов**  **ReDim fam(n - 1), so(n - 1)**  **'вызов процедуры ввода группы из файла(создание массивов**  **'фамилий студентов и массива оценок)**  **vvod(mas, n, m, fam, oc)**  **'вызов процедуры вычисления среднего балла каждого студента**  **'и общего среднего балла группы**  **sred(n, m, oc, so, sr)**  **'Вывод полученных массивов на форму**  **VivodMasStr(fam, s)**  **TextBox1.Text = s**  **vivodMatr(oc, n, m, s)**  **TextBox2.Text = s**  **VivodMas(so, s)**  **TextBox3.Text = s**  **TextBox4.Text = Format(sr, "0.0000")**  **'Кнопку3 делаем доступной после ввода данных**  **' и расчета средних баллов**  **Button3.Enabled = True**  **End Sub**  **'кнопка для перехода на 2-ю форму**  **Private Sub Button3\_Click(ByVal sender As System.Object, \_**  **ByVal e As System.EventArgs) \_**  **Handles Button3.Click**  **Form2.Show()**  **End Sub**  **Private Sub Button2\_Click(ByVal sender As System.Object, \_**  **ByVal e As System.EventArgs) \_**  **Handles Button2.Click**  **End**  **End Sub**  **End Class**  **'Программный код 2 формы**  **Option Strict On**  **Public Class Form2**  **'Объекты k,f(n-1)и sb(n-1)объявляем глобальными для 2-й формы,**  **'т.к. они должны быть доступны сразу двум событийным процедурам**  **'(для кнопок "Решить задачу" и "Записать в файл")**  **Dim k As Integer 'кол-во студ, к-рым надо повысить успеваемость**  **Dim f(n - 1) As String 'массив их фамилий**  **Dim sb(n - 1) As Double 'массив их средних баллов**  **' Формирование массива f() фамилий студентов,**  **' имеющих средний балл ниже общего среднего**  **' и массива sb() средних баллов этих студентов**  **Public Sub FormirSpisok(ByVal n As Integer, ByVal sr As Double, \_**  **ByVal so() As Double, ByVal fam() As String, \_**  **ByRef f() As String, ByRef sb() As Double, \_**  **ByRef k As Integer)**  **Dim i As Integer**  **k = 0**  **For i = 0 To n - 1**  **If so(i) < sr Then**  **f(k) = fam(i)**  **sb(k) = so(i)**  **k = k + 1**  **End If**  **Next**  **ReDim Preserve f(k - 1), sb(k - 1)**  **End Sub**  **'сортировка по убыванию среднего балла студента**  **Sub sort(ByRef s() As String, ByRef kol() As Double)**  **Dim i, j, k As Integer, temp As String, t As Double**  **k = s.Length - 1**  **For i = 0 To k - 1**  **For j = i + 1 To k**  **If kol(i) < kol(j) Then**  **t = kol(i) : kol(i) = kol(j) : kol(j) = t**  **temp = s(i) : s(i) = s(j) : s(j) = temp**  **End If**  **Next**  **Next**  **End Sub**  **' Функция формирования одной строки из отсортированного массива**  **' студентов,имеющих средний балл ниже общего среднего**  **Function res(ByVal f() As String, ByVal sb() As Double, \_**  **ByVal k As Integer) As String**  **'Создание объекта result типа StringBuilder(результирующей строки)**  **'для очень быстрой и эффективной обработки строковых данных**  **Dim result As New System.Text.StringBuilder**  **For i As Integer = 0 To k - 1**  **result.Append(f(i)) 'добавление фамилии в объект result**  **result.Append(Space(1)) 'добавление пробела**  **result.Append(sb(i).ToString) 'добавление среднего балла**  **result.Append(vbNewLine)'добавление признака конца строки**  **Next i**  **Return result.ToString**  **End Function**  **' Процедура записи результата в текстовый файл**  **Sub vivodFile(ByVal filename As String, ByRef S As String)**  **'Запись строки-результата S в текстовый файл;значение False**  **'третьего параметра задает,что файл будет перезаписан.**  **Try**  **My.Computer.FileSystem.WriteAllText(filename, S, False)**  **Catch ex As Exception**  **MsgBox("ошибка при чтении файла")**  **End Try**  **End Sub**  **'кнопка "Решить задачу"**  **Private Sub Button1\_Click\_1(ByVal sender As System.Object, \_**  **ByVal e As System.EventArgs) \_**  **Handles Button1.Click**  **Dim z As String = "" 'строка для вывода массивов**  **FormirSpisok(n, sr, so, fam, f, sb, k)**  **sort(f, sb)**  **VivodMasStr(f, z)**  **TextBox1.Text = z**  **VivodMas(sb, z)**  **TextBox2.Text = z**  **TextBox3.Text = Format(sr, "0.0000")**  **End Sub**  **'кнопка "Записать в файл"**  **Private Sub Button2\_Click(ByVal sender As System.Object, \_**  **ByVal e As System.EventArgs) \_**  **Handles Button2.Click**  **Dim S As String = ""**  **'Вызов функции формирования одной строки из отсортированного**  **'массива студентов,имеющих средний балл ниже общего среднего**  **S = res(f, sb, k)**  **'Вызов процедуры записи результата в текстовый файл**  **vivodFile("file2.txt", S)**  **End Sub**  **Private Sub Button3\_Click\_1(ByVal sender As System.Object, \_**  **ByVal e As System.EventArgs) \_**  **Handles Button3.Click**  **End**  **End Sub**  **'кнопка "Повторить"**  **Private Sub Button4\_Click\_1(ByVal sender As System.Object, \_**  **ByVal e As System.EventArgs) \_**  **Handles Button4.Click**  **Me.Close()**  **Form1.TextBox1.Text = ""**  **Form1.TextBox2.Text = ""**  **Form1.TextBox3.Text = ""**  **Form1.TextBox4.Text = ""**  **Form1.Button1.Enabled = True**  **Form1.Button2.Enabled = True**  **Form1.Button3.Enabled = False**  **Form1.Show()**  **End Sub**  **End Class** |

Рис. 5.1-14

1. *Выполнение приложения и получение результатов:*

Результаты работы проекта Проект-5-1-Лаб-3 (вариант 3) представлены на рис. 5.1-15 и рис. 5.1.16



|  |
| --- |
| Рис. 5.1-15 |

Рис. 5.1-16

1. *Доказательство правильности работы программ.*

В результате работы проекта из файла введены исходные данные чтением целиком в одну строку. Прочитанная строка корректно преобразована в массивы, которые использованы для подсчета средней оценки каждого студента и среднего балла всей группы и выведены на 1 форму. На второй форме представлены полученные и отсортированные по убыванию среднего балла массив фамилий и массив среднего балла тех студентов, которым надо повысить успеваемость до фактического среднего уровня (у которых средняя оценка ниже общего среднего балла всей группы). Полученные результаты сформированы в одну строку, которая целиком записана в другой текстовый файл.

### 5.1.5. Контрольные вопросы по теме «Стандартные модули и модульная структура приложений в VB»

1. Какие архитектурные компоненты платформы .NET известны?
2. Какую роль (какие задачи) выполняет среда CLR?
3. Что представляет из FCL?
4. Из каких этапов состоит компиляция .NET-приложения?
5. Какова структура Windows-приложения?
6. Что представляет стандартный модуль?
7. Каковы сходства и различия стандартного модуля, модуля формы и модуля класса?
8. Что необходимо для создания проекта с несколькими формами и организации взаимодействия между ними?