ЛР №16 Классы

Задание 1.

Поля класса: Число А,Б,С означающие часы, минуты, секунды соответсвенно

Методы класса:

* Конструктор ,
* Метод вывода времени на экран в формате «чч:мм:сс», «чч:мм:сс АМ/РМ»
* Нахождение кол-ва часов, минут и секунд между двумя значениями времени,
* Нахождение времени суток(утро, день, вечер, ночь) для времени,
* Перевод времени в числовой формат

Операторы перегрузки:

1. Сравнение двух значений времени(>),
2. Инкремент (++): увеличение часа времени на один час, минут – на одну минуту, секунд – одну секунду
3. Сложение двух значений времени(+)

Задание 2.

Класс Конус

Поля класса: радиус основания r и высоты конуса h

Методы класса:

1. Вычисление площади боковой поверхности конуса;
2. Объема конуса;
3. Площади сечения, параллельного плоскости основания конуса;
4. Площади сечения, перпендикулярного плоскости основания конуса;

Операторы перегрузки:

1. Сложение конусов (+): Конус1(r1,h1), Конус2(r2,h2), тогда Конус1+Конус2 имеет поля (r1+r2,h1+h2).
2. Сравнение двух конусов(>): конус(r1,h1)>конус(r2,h2), если объем конус1>объём конус2
3. Декремент конуса(--): Конус(r,h) , тогда – конус(r-1,h-1)

Задание 3.

Класс анкета:

1. ФИО;
2. Группа;
3. Направление подготовки;
4. Год поступления;
5. Пол;
6. Национальность;
7. Дата рождения;
8. Размер стипендии;

Методы класса:

1. Возраст на момент поступления;
2. Сумма на выплату стипендии за все годы обучения;
3. Текущий семестр;

1-3 задания написано на C#

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace лр16сем2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

struct MyDateTime

{

public static string hours, minutes, seconds, h1, m1, s1;

public static string gettime() { return $"{hours}:{minutes}:{seconds} {ampm}"; }

public static string ampm;

public static string difftime;

public static string timelevel;

public static int converttodate;

public static string conv\_int\_to\_date;

public static string crement;

}

class MyClass

{

public int hour = Convert.ToInt32(MyDateTime.hours), minute = Convert.ToInt32(MyDateTime.minutes), second = Convert.ToInt32(MyDateTime.seconds);

public int hour1 = Convert.ToInt32(MyDateTime.h1), minute1 = Convert.ToInt32(MyDateTime.m1), second1 = Convert.ToInt32(MyDateTime.s1);

public void correcttime()

{

if (hour >= 0 && hour <= 12) { MyDateTime.ampm = "am"; } else { MyDateTime.ampm = "pm"; MyDateTime.hours = (hour - 12).ToString(); }

if (hour.ToString().Length < 2) MyDateTime.hours = "0" + MyDateTime.hours;

if (minute.ToString().Length < 2) MyDateTime.minutes = "0" + MyDateTime.minutes;

if (second.ToString().Length < 2) MyDateTime.seconds = "0" + MyDateTime.seconds;

if (hour1.ToString().Length < 2) MyDateTime.h1 = "0" + MyDateTime.h1;

if (minute1.ToString().Length < 2) MyDateTime.m1 = "0" + MyDateTime.m1;

if (second1.ToString().Length < 2) MyDateTime.s1 = "0" + MyDateTime.s1;

if (MyDateTime.crement.Split(':')[0].Length < 2) MyDateTime.crement = MyDateTime.crement.Insert(0, "0");

if (MyDateTime.crement.Split(':')[1].Length < 2) MyDateTime.crement = MyDateTime.crement.Insert(3, "0");

if (MyDateTime.crement.Split(':')[2].Length < 2) MyDateTime.crement = MyDateTime.crement.Insert(6, "0");

}

public void findtimelevel()

{

int h = hour;

if (MyDateTime.ampm == "am" && h > 5 && h < 10) MyDateTime.timelevel = "утро";

else if (MyDateTime.ampm == "am" && h > 16 && h < 20) MyDateTime.timelevel = "вечер";

else if (MyDateTime.ampm == "pm") MyDateTime.timelevel = "ночь";

}

public void differ()

{

MyDateTime.difftime = diff(hour, minute, second, hour1, minute1, second1);

int h = MyDateTime.converttodate / 3600; MyDateTime.converttodate %= 3600;

int m = MyDateTime.converttodate / 60; MyDateTime.converttodate %= 60;

int s = MyDateTime.converttodate;

MyDateTime.conv\_int\_to\_date = $"{h}:{m}:{s}";

}

public void crem()

{

MyDateTime.crement = crement(hour, minute, second);

}

}

private void button1\_Click(object sender, EventArgs e)

{

MyClass myClass = new MyClass();

string date = textBox1.Text, date2 = textBox2.Text;

MyDateTime.hours = date.Split()[0]; MyDateTime.minutes = date.Split()[1]; MyDateTime.seconds = date.Split()[2];

MyDateTime.h1 = date2.Split()[0]; MyDateTime.m1 = date2.Split()[1]; MyDateTime.s1 = date2.Split()[2];

MyDateTime.converttodate = int.Parse(textBox3.Text);

myClass.hour = Convert.ToInt32(date.Split()[0]); myClass.minute = Convert.ToInt32(date.Split()[1]); myClass.second = Convert.ToInt32(date.Split()[2]);

myClass.hour1 = Convert.ToInt32(date2.Split()[0]); myClass.minute1 = Convert.ToInt32(date2.Split()[1]); myClass.second1 = Convert.ToInt32(date2.Split()[2]);

myClass.differ();

myClass.findtimelevel();

myClass.crem();

myClass.correcttime();

label6.Text = "разница часов " + MyDateTime.difftime;

label3.Text = $"время1 {MyDateTime.hours}:{MyDateTime.minutes}:{MyDateTime.seconds}";

label5.Text = $"время2 {MyDateTime.h1}:{MyDateTime.m1}:{MyDateTime.s1}";

label10.Text = compare(Convert.ToInt32(MyDateTime.hours), Convert.ToInt32(MyDateTime.minutes), Convert.ToInt32(MyDateTime.seconds), Convert.ToInt32(MyDateTime.h1), Convert.ToInt32(MyDateTime.m1), Convert.ToInt32(MyDateTime.s1));

label9.Text = "время суток " + MyDateTime.timelevel + " " + MyDateTime.ampm;

label12.Text = "сложение времени " + summ(Convert.ToInt32(MyDateTime.hours), Convert.ToInt32(MyDateTime.minutes), Convert.ToInt32(MyDateTime.seconds), Convert.ToInt32(MyDateTime.h1), Convert.ToInt32(MyDateTime.m1), Convert.ToInt32(MyDateTime.s1));

label11.Text = "время после инкремента " + MyDateTime.crement;

label8.Text = "время " + MyDateTime.conv\_int\_to\_date;

}

public static string diff(int h, int m, int s, int h1, int m1, int s1)

{

int contos = h \* 3600 + m \* 60 + s, contos1 = h1 \* 3600 + m1 \* 60 + s1;

int diff = contos - contos1;

int hour = diff / 3600; diff %= 3600;

int minute = diff / 60; diff %= 60;

int sec = diff;

return $"{hour}:{minute}:{sec}";

}

public static string compare(int h, int m, int s, int h1, int m1, int s1)

{

if (h > h1) return ">"; else if (h < h1) return "<";

if (m > m1) return ">"; else if (m < m1) return "<";

if (s > s1) return ">"; else if (s < s1) return "<";

return "=";

}

public static string crement(int h, int m, int s)

{

int h1 = h + 1;

int m1 = m + 1;

int s1 = s + 1;

if (s1 > 59) { s1 -= 60; m1++; }

if (m1 > 59) { m1 -= 60; h1++; }

if (h1 > 23) { h1 -= 24; }

return $"{h1}:{m1}:{s1}";

}

public static string summ(int h, int m, int s, int h1, int m1, int s1)

{

h += h1;

m += m1;

s += s1;

if (s > 59) { s -= 60; m++; }

if (m > 59) { m -= 60; h++; }

if (h > 23) { h -= 24; }

return $"{h} {m} {s}";

}

public static Dictionary<int, List<string>> people = new Dictionary<int, List<string>>(25);

public static int isvisited;

struct MyStruct

{

public static int[] group;

public static string[] learningto;

public static string[] pname;

public static string[] pol;

public static int[] yearofenter;

public static string[] nationality;

public static int[] stipen;

public static string[] dateTimes;

}

private void button2\_Click(object sender, EventArgs e)

{

}

public class myclass

{

public int agewhenenter(int yearofenter,int birth )

{

return yearofenter-birth;

}

public int sumstipen(int yearofenter,int stipen)

{

return (2022 - yearofenter) \* stipen \* 9;

}

public int currrentsemestr(int yearofinter)

{

return 2022 - yearofinter;

}

}

private void Form1\_Load(object sender, EventArgs e)

{

myclass myclass = new myclass();

MyStruct.pname = new string[] {

"andrey Иванов",

"Александра Смирнов",

"Алена Кузнецов",

"Алина Попов",

"Алиса Васильев",

"Алла Петров",

"Анастасия Соколов",

"Ангелина Михайлов",

"Анна Новиков",

"Антонина Федоров",

"Арина Морозов",

"Валентина Волков",

"Валерия Алексеев",

"Варвара Лебедев",

"Василиса Семенов",

"Вера Егоров",

"Вероника Павлов",

"Виктория Козлов",

"Галина Степанов",

"Дарья Николаев",

"Ева Орлов",

"Евгения Андреев",

"Евдокия Макаров",

"Екатерина Никитин",

"Елена Захаров",

"Елизавета Зайцев",

"Жанна Соловьев",

"Зинаида Борисов",

"Злата Яковлев",

"Зоя Григорьев",

"Иванна Романов",

"Ирина Воробьев",

"Ия Сергеев",

"Кира Кузьмин",

"Кристина Казаков",

"Ксения Афанасьев",

"Лидия Данилов",

"Любовь Тимофеев",

"Людмила Фомин",

"Маргарита Чернов",

"Марина Абрамов",

"Мария Мартынов",

"Милана Ефимов",

"Надежда Федотов",

"Наталья Щербаков",

"Нина Назаров",

"Ольга Калинин",

"Полина Исаев",

"Раиса Чернышев",

"Светлана Быков",

"София Маслов"

};

MyStruct.pol = new string[] { "мужской", "женской" };

MyStruct.group = new int[] { 101, 102, 103, 104, 105, 106 };

MyStruct.learningto = new string[] { "программная инженерия", "ПМиИТ", "ФМиИТ", "ИБ", "Химия Физика", "Приклодная Химия" };

MyStruct.yearofenter = new int[] { 2018, 2019, 2020, 2021, 2022 };

MyStruct.nationality = new string[] { "таджик", "туркмен", "русский", "узбек", "казах", "индейц", "китайц", "корейц" };

MyStruct.stipen = new int[] { 2300, 1800, 2000, 2500 };

MyStruct.dateTimes = new string[] { "1995,06,20", "1999,06,26", "2003,03,07", "2001,02,25", "2002,12,15", "2001,01,03", "2004,01,15" };

dataGridView1.ColumnCount = 11;

dataGridView1.Columns[0].Name = "ФИО";

dataGridView1.Columns[1].Name = "Группа";

dataGridView1.Columns[2].Name = "Направление подготовки";

dataGridView1.Columns[3].Name = "Год поступление";

dataGridView1.Columns[4].Name = "Пол";

dataGridView1.Columns[5].Name = "Национальность";

dataGridView1.Columns[6].Name = "Дата рождение";

dataGridView1.Columns[7].Name = "Размер стипендии";

dataGridView1.Columns[8].Name = "Возраст на момент поступления";

dataGridView1.Columns[9].Name = "Сумма на выплату стипендии за все годы обучения";

dataGridView1.Columns[10].Name = "Текущий семестр";

var random = new Random();

dataGridView1.RowCount = MyStruct.pname.Length;

for (int item = 0; item < MyStruct.pname.Length; item++)

{

people.Add(item, new List<string> {

MyStruct.pname[item],

Convert.ToString( MyStruct.group[random.Next(0,MyStruct.group.Length)]),

MyStruct.learningto[random.Next(0, MyStruct.learningto.Length)],

Convert.ToString(MyStruct.yearofenter[random.Next(0, MyStruct.yearofenter.Length)]),

MyStruct.pol[random.Next(0, MyStruct.pol.Length)],

MyStruct.nationality[random.Next(0, MyStruct.nationality.Length)],

MyStruct.dateTimes[random.Next(0, MyStruct.dateTimes.Length)],

MyStruct.stipen[random.Next(0,MyStruct.stipen.Length)].ToString()

});

dataGridView1.Rows[item].Cells[0].Value = people[item][0];

dataGridView1.Rows[item].Cells[1].Value = people[item][1];

dataGridView1.Rows[item].Cells[2].Value = people[item][2];

dataGridView1.Rows[item].Cells[3].Value = people[item][3];

dataGridView1.Rows[item].Cells[4].Value = people[item][4];

dataGridView1.Rows[item].Cells[5].Value = people[item][5];

dataGridView1.Rows[item].Cells[6].Value = people[item][6];

dataGridView1.Rows[item].Cells[7].Value = people[item][7];

dataGridView1.Rows[item].Cells[8].Value = myclass.agewhenenter(int.Parse(dataGridView1.Rows[item].Cells[3].Value.ToString()),int.Parse(dataGridView1.Rows[item].Cells[6].Value.ToString().Split(',')[0])).ToString();

dataGridView1.Rows[item].Cells[9].Value = myclass.sumstipen(int.Parse(dataGridView1.Rows[item].Cells[3].Value.ToString()), int.Parse(dataGridView1.Rows[item].Cells[7].Value.ToString())).ToString();

dataGridView1.Rows[item].Cells[10].Value = myclass.currrentsemestr(int.Parse(dataGridView1.Rows[item].Cells[3].Value.ToString()))+1;

}

}

}

}

Решение задание 2 на языке C++

#include <iostream>

#include <math.h>

using namespace std;

class konus {

private:

double r;

double h;

const double PI = 3.14159265;

public:

konus(double radius, double height) {

r = radius;

h = height;

}

double areaSide() {

return PI \* r \* sqrt(r \* r + h \* h);

}

double volume() {

return 1. / 3 \* PI \* r \* r \* h;

}

double secarea(double height) {

double radius = height \* r / h;

return PI \* radius \* radius;

}

double areasecnorm(double d) {

double height = d \* h / 2;

return height;

}

konus operator + (konus c) {

konus cTemp(r + c.r, h + c.h);

return cTemp;

}

bool operator > (konus c) {

if (this -> volume() > c.volume()) {

return true;

cout << "Первый конус больше второго\n";

}

else {

return false;

}

}

konus operator - () {

r--;

h--;

return \*this;

}

void display() {

cout << "\nРадиус = " << r;

cout << "\nВысота = " << h << endl;

}

};

int main()

{

setlocale(LC\_ALL, "RUS");

double r, h;

cout << "Введите радиус конуса: \n";

cin >> r;

cout << "Введите высоту конуса: \n";

cin >> h;

konus c1(r, h);

cout << "\nПлощадь боковой поверхности конуса: " << c1.areaSide() << endl;

cout << "\nОбъём конуса равен: " << c1.volume() << endl;

cout << "\nВведите высоту на которой находится сечение, параллельное основанию конуса, \n для нахождения его площади (меньше или равно " << h << " и больше 0): ";

cin >> h;

cout << "\nПлощадь сечения параллельного основанию конуса равна: " << c1.secarea(h) << endl;

cout << "\nВведите диаметр основания сечения, перпендикулярного основанию конуса,\n для нахождения его площади (меньше или равно " << 2 \* r << " и больше 0): ";

cin >> r;

cout << "\nПлощадь сечения перпендикулярного основанию конуса равна: " << c1.areasecnorm(r) << endl;

cout << "\nВведите радиус конуса: \n";

cin >> r;

cout << "Введите высоту конуса: \n";

cin >> h;

konus c2(r, h);

cout << "\nСложение двух конусов: ";

konus c3 = c1 + c2;

c3.display();

cout << "\nСравнение двух конусов (>):\n";

if (c1 > c2) {

cout << "Первый конус больше второго\n";

}

else {

cout << "Первый конус меньше второго или равен ему\n";

}

- c1;

cout << "\nДекремент первого конуса (--):";

c1.display();

system("pause");

return 0;

}

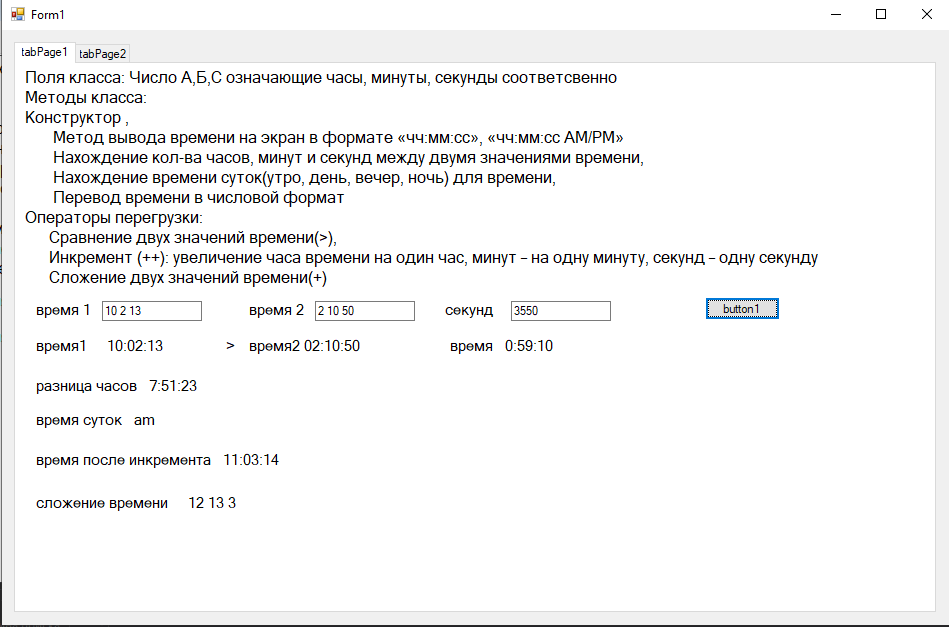


Рисунок 1.1 Результат работы первого теста

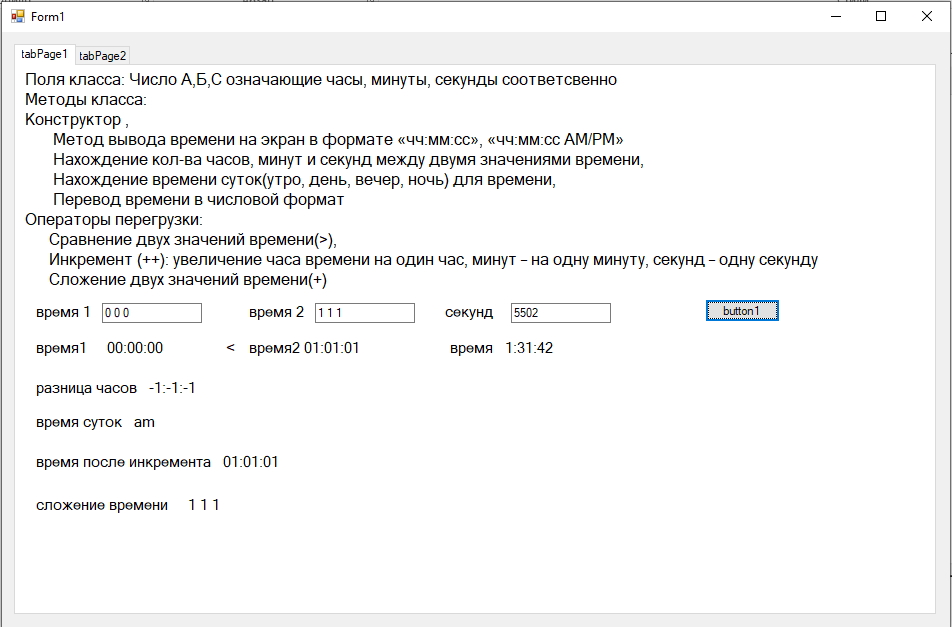


Рисунок 1.2 Результат работы второго теста



Рисунок 2.1 Результат первого теста

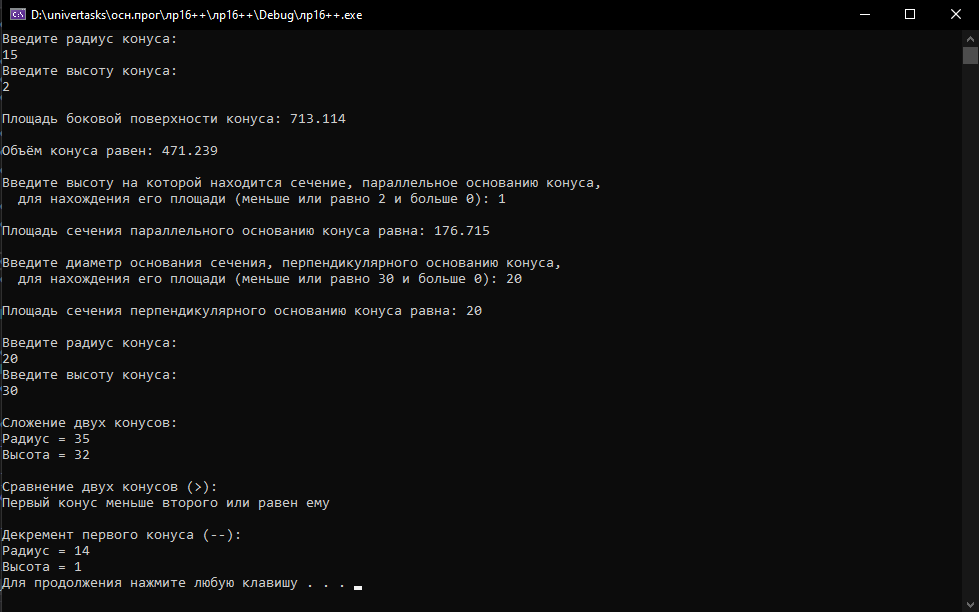


Рисунок 2.2 Результат второго теста

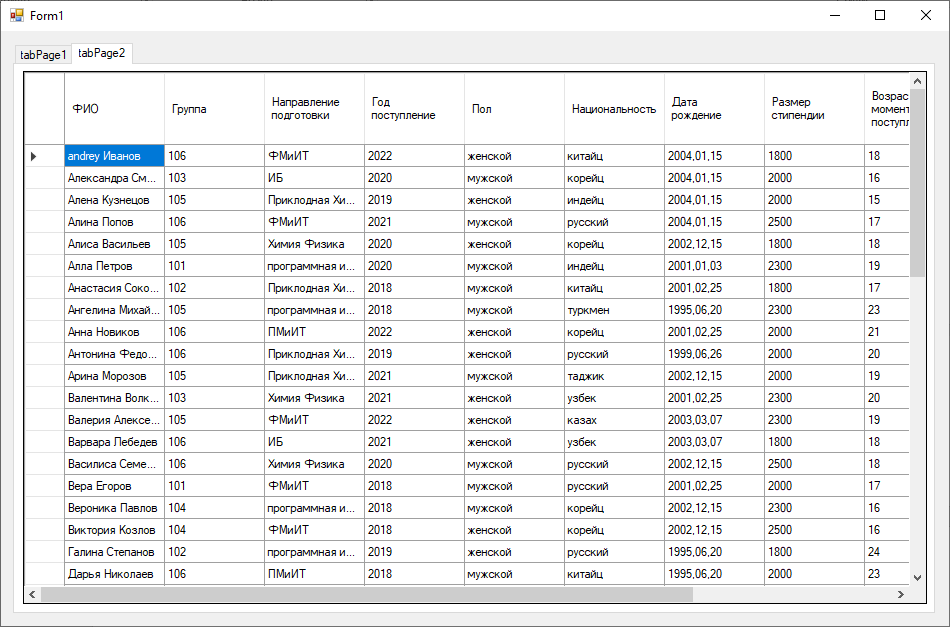


Рисунок 3.1 Структура

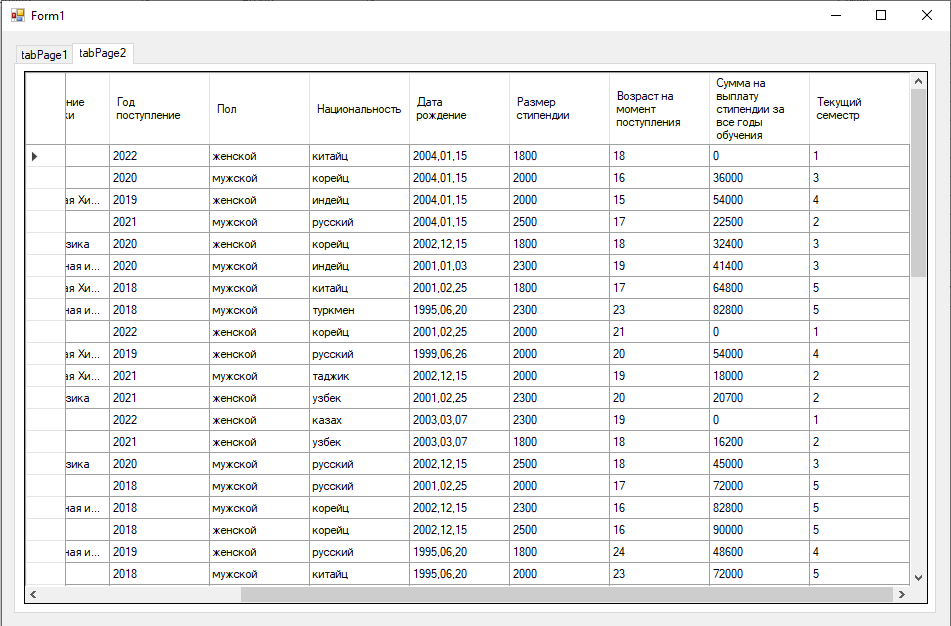


Рисунок 3.2 Результат работы третий заданий