­­Министерство науки и высшего образования РФ  
ФГАОУ ВПО

Национальный исследовательский технологический университет «МИСиС»

Институт Информационных технологий и компьютерных наук (ИТКН)

Кафедра Инфокоммуникационных технологий (ИКТ)

**Отчет по лабораторной работе №12(4)**

по дисциплине «Объектно-ориентированное программирование»

на тему «Структуры в Windows Forms»

Выполнил:  
студент группы БИВТ-22-15

Ханцева А.А.

Проверил:   
Буянов С.И.

Москва, 2022

Оглавление

[Цель 2](#_Toc132844595)

[Задание 2](#_Toc132844596)

[I уровень 2](#_Toc132844597)

[II уровень 2](#_Toc132844598)

[III уровень 3](#_Toc132844599)

[Листинг программы 4](#_Toc132844600)

[Пример работы программ 19](#_Toc132844601)

[I уровень. Задание 1 19](#_Toc132844602)

[I уровень. Задание 2 20](#_Toc132844603)

[II уровень. Задание 8 21](#_Toc132844604)

[II уровень. Задание 9 22](#_Toc132844605)

[III уровень. Задание 2 23](#_Toc132844606)

[III уровень. Задание 3 24](#_Toc132844607)

## Цель

Повторить структуры и способы работы с ними через Windows Forms.

## Задание

### I уровень

1. Результаты соревнований по прыжкам в длину определяются по сумме двух попыток. В протоколе для каждого участника указываются: фамилия, общество, результаты первой и второй попыток. Вывести протокол в виде таблицы с заголовком в порядке занятых мест.
2. Составить программу для обработки результатов кросса на 500 м для женщин. Для каждой участницы ввести фамилию, группу, фамилию преподавателя, результат. Получить результирующую таблицу, упорядоченную по результатам, в которой содержится также информация о выполнении норматива. Определить суммарное количество участниц, выполнивших норматив.

### II уровень

1. Для формирования сборной по хоккею предварительно отобрано 30 игроков. На основании протоколов игр составлена таблица, в которой содержится штрафное время каждого игрока по каждой игре (2, 5 или 10 мин). Написать программу, которая составляет список кандидатов в сборную в порядке возрастания суммарного штрафного времени. Игрок, оштрафованный на 10 мин, из списка кандидатов исключается.
2. Результаты соревнований фигуристов по одному из видов многоборья представлены оценками семи судей в баллах (от 0,0 до 6,0). По результатам оценок судьи определяется место каждого участника у этого судьи. Места участников определяются далее по сумме мест, которые каждый участник занял у всех судей. Составить программу, определяющую по исходной таблице оценок фамилии и сумму мест участников в порядке занятых ими мест.

### III уровень

1. Соревнования по футболу между командами проводятся в два этапа. Для проведения первого этапа участники разбиваются на две группы по 12 команд. Для проведения второго этапа выбирается 6 лучших команд каждой группы по результатам первого этапа. Составить список команд участников второго этапа.
2. В соревнованиях участвуют три команды по 6 человек. Результаты соревнований представлены в виде мест участников каждой команды (1 - 18). Определить команду – победителя, вычислив количество баллов, набранное каждой командой. Участнику, занявшему 1-е место, начисляется 5 баллов, 2-е – 4, 3-е – 3, 4-е – 2, 5-е – 1, остальным – 0 баллов. При равенстве баллов победительницей считается команда, за которую выступает участник, занявший 1-е место.

# Листинг программы

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Net.NetworkInformation;

using System.Reflection.Emit;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using static System.Net.Mime.MediaTypeNames;

using static System.Windows.Forms.VisualStyles.VisualStyleElement.Rebar;

namespace Lab12

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

this.Text = "Лабораторная работа №4 (за 2 семестр)";

string[] levels = { "1 Уровень", "2 Уровень", "3 Уровень" };

listBox1.Items.AddRange(levels);

listBox1.Location = new Point(10, 10);

listBox1.Width = (this.ClientRectangle.Width - 20) / 2;

listBox2.Size = listBox1.Size;

listBox2.Top = listBox1.Top;

listBox2.Left = listBox1.Left + listBox1.Width;

listBox2.Hide();

label1.Location = new Point(10, listBox1.Bottom);

label1.Hide();

}

int y = -1;

private void listBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

string[] task = { " ", " ", " " };

if (listBox1.SelectedIndex != y)

{

listBox2.Show();

y = listBox1.SelectedIndex;

switch (y)

{

case 0:

task[0] = "Задание 1";

task[1] = "Задание 2";

break;

case 1:

listBox2.Items.Clear();

task[0] = "Задание 8";

task[1] = "Задание 9";

break;

case 2:

task[0] = "Задание 2";

task[1] = "Задание 3";

break;

}

listBox2.Items.Clear();

listBox2.Items.AddRange(task);

Controls.Add(listBox2);

}

else

{

listBox2.Items.Clear();

//Controls.Remove(listBox2);

//listBox2.Dispose();

}

y = listBox1.SelectedIndex;

}

private void listBox2\_SelectedIndexChanged\_1(object sender, EventArgs e)

{

label1.Show();

label1.Text = "";

switch ((listBox1.SelectedIndex, listBox2.SelectedIndex))

{

case (0, 0):

l1t1();

break;

case (0, 1):

l1t2();

break;

case (1, 0):

l2t8();

break;

case (1, 1):

l2t9();

break;

case (2, 0):

l3t2();

break;

case (2, 1):

l3t3();

break;

case ( < 3, -1):

label1.Hide();

label1.Text = "";

break;

}

this.AutoSize = false;

this.AutoSize = true;

}

struct Athlete

{

public string lastname;

public string company;

public double result1, result2;

}

//play = play.OrderByDescending(p => p.win).ToArray();

void l1t1()

{

Random r = new Random((int)DateTime.Now.Ticks);

label1.Text += "Сложность 1.Задание 1\n Введённые данные:\n" + $"{"№",3} " + $"{"Фамилия",12}" + $"{"Общество",12}" + $"{"Попытка№1",12}" + $"{"Попытка№2",12}\n";

int n = r.Next(5, 21);

Athlete[] athlete = new Athlete[n];

for (int i = 0; i < n; i++)

{

athlete[i].lastname = $"Фам{i}илия";

athlete[i].company = $"Комап{n - i}ания";

athlete[i].result1 = r.Next(100, 290) \* 0.01;

athlete[i].result2 = r.Next(100, 290) \* 0.01;

label1.Text += $"{i + 1,3})" + $"{athlete[i].lastname,12}" + $"{athlete[i].company,12}" + $"{athlete[i].result1,12}" + $"{athlete[i].result2,12}\n";// + $"{(athlete[i].result2 + athlete[i].result1) / 2,12}\n";

}

label1.Text += " Протокол в порядке занятых мест(по среднему результату): \n" + $"{"№",3} " + $"{"Фамилия",12}" + $"{"Общество",12}" + $"{"Попытка№1",12}" + $"{"Попытка№2",12}\n";

//this.Height =10+ listBox1.Height+ label1.Padding.Vertical+ TextRenderer.MeasureText(label1.Text, label1.Font).Height ;

athlete = athlete.OrderByDescending(p => (p.result1 + p.result2)).ToArray();

for (int i = 0; i < n; i++)

label1.Text += $"{i + 1,3})" + $"{athlete[i].lastname,12}" + $"{athlete[i].company,12}" + $"{athlete[i].result1,12}" + $"{athlete[i].result2,12}\n"; //+ $"{(athlete[i].result2 + athlete[i].result1) / 2,12}\n";

}

struct cross

{

public string name;

public string group;

public string teacher;

public double time;

}

void l1t2()

{

Random r = new Random((int)DateTime.Now.Ticks);

int n = r.Next(5, 21);

label1.Text += "Сложность 1.Задание 2\n Введённые данные:\n" + $"{"№",3} " + $"{"Фамилия",12}" + $"{"Группа",12}" + $"{"Учитель",12}" + $"{"Результат",12}\n";

cross[] run = new cross[n];

for (int i = 0; i < run.Length; i++)

{

run[i].name = $"И{i}мя";

run[i].group = $"Гру{i + 1 - i % 4}ппа";

run[i].teacher = $"Физ{n - i}рук";

run[i].time = r.Next(18, 35) \* 0.1;

label1.Text += $"{i + 1,3})" + $"{run[i].name,12}" + $"{run[i].group,12}" + $"{run[i].teacher,12}" + $"{run[i].time,12}\n";

}

run = run.OrderBy(p => p.time).ToArray();

label1.Text += " Протокол по результатам:\n" + $"{"№",3} " + $"{"Фамилия",12}" + $"{"Группа",12}" + $"{"Учитель",12}" + $"{"Результат",12}\n";

for (int i = 0; i < run.Length; i++)

label1.Text += $"{i + 1,3})" + $"{run[i].name,12}" + $"{run[i].group,12}" + $"{run[i].teacher,12}" + $"{run[i].time,12}\n";

int k = 0;

for (int i = 0; i < run.Length; i++)

if (run[i].time < 2.5)

k++;

label1.Text += $" Сдавшим нормативом считаются те, кто пробежал менее чем за 2.5 мин. Кол-во сдавших {k}\n";

}

struct hockey

{

public string name;

public int r1;

public int r2;

public int r3;

}

void l2t8()

{

Random r = new Random((int)DateTime.Now.Ticks);

label1.Text += " Сложность 2. Задание 8\n Полный список кандидатов:\n"+ $"{"Фамилия",12}" + $"{"#1",3}" + $"{"#2",3}" + $"{"#3",3}\n";

hockey[] team = new hockey[30];

int[] time = new int[] { 0, 2, 5, 10 };

for (int i = 0; i < team.Length; i++)

{

team[i].name = $"Хок{i}кеист";

team[i].r1 = time[r.Next(3)];

team[i].r2 = time[r.Next(time.Length)];

team[i].r3 = time[r.Next(time.Length)];

label1.Text+=$"{team[i].name,12}" + $"{team[i].r1,3}" + $"{team[i].r2,3}" + $"{team[i].r3,3}\n";

}

label1.Text += " Урезанный список кандидатов:\n"+ $"{"№",3}" + $"{"Фамилия",12}" + $"{"#1",3}" + $"{"#2",3}" + $"{"#3",3}\n";

int n = 0;

for (int i = 0; i < team.Length; i++)

if (team[i].r1 == 10 || team[i].r2 == 10 || team[i].r3 == 10)

{

team[i].r1 = 10;

team[i].r2 = 10;

team[i].r3 = 10;

}

team = team.OrderBy(p => (p.r1+ p.r2+ p.r3)).ToArray();

while((n < team.Length) && (team[n].r1 != 10) && (team[n].r3 != 10) && (team[n].r3 != 10))

n++;

for (int i = 0; i < n; i++)

label1.Text += $"{i + 1,3}" + $"{team[i].name,12}" + $"{team[i].r1,3}" + $"{team[i].r2,3}" + $"{team[i].r3,3}\n";// + $"{team[i].r1+ team[i].r2+team[i].r3,3}\n";

}

struct grades

{

public double r1;

public double r2;

public double r3;

public double r4;

public double r5;

public double r6;

public double r7;

public string name;

}

void l2t9()

{

Random r = new Random((int)DateTime.Now.Ticks);

grades[] skate = new grades[15];

label1.Text += " Сложность 2. Задание 9\n Обычный список участников:\n" + $"{"Фамилия",12}" + $"{"#1",6}" + $"{"#2",6}" + $"{"#3",6}" + $"{"#4",6}" +

$"{"#5",6}" + $"{"#6",6}" + $"{"#7",6}\n";

for (int i = 0; i < skate.Length; i++)

{

skate[i].name = $"Фигу{i}рист";

skate[i].r1 = r.Next(0, 60 + 1) \* 0.1;

skate[i].r2 = r.Next(0, 60 + 1) \* 0.1;

skate[i].r3 = r.Next(0, 60 + 1) \* 0.1;

skate[i].r4 = r.Next(0, 60 + 1) \* 0.1;

skate[i].r5 = r.Next(0, 60 + 1) \* 0.1;

skate[i].r6 = r.Next(0, 60 + 1) \* 0.1;

skate[i].r7 = r.Next(0, 60 + 1) \* 0.1;

label1.Text += $"{skate[i].name,12}" + $"{skate[i].r1,6}" + $"{skate[i].r2,6}" + $"{skate[i].r3,6}" +

$"{skate[i].r4,6}" + $"{skate[i].r5,6}" + $"{skate[i].r6,6}" + $"{skate[i].r7,6}\n"; ;

}

int[] sum = new int[skate.Length];

place(skate, sum);

for (int i = 0; i < skate.Length; i++)

{

Console.WriteLine($"{skate[i].name,12}" + $"{skate[i].r1,6}" + $"{skate[i].r2,6}" + $"{skate[i].r3,6}" +

$"{skate[i].r4,6}" + $"{skate[i].r5,6}" + $"{skate[i].r6,6}" + $"{skate[i].r7,6}" + $"{sum[i],6}");

}

for (int i = 0; i < skate.Length; i++)

for (int j = i + 1; j < skate.Length; j++)

if (sum[i] > sum[j])

{

int z = sum[i];

sum[i] = sum[j];

sum[j] = z;

grades s = skate[i];

skate[i] = skate[j];

skate[j]= s;

}

label1.Text += "Сортированный список участников:\n" + $"{"№",5}" + $"{"Фамилия",12}" + $"{"#1",6}" + $"{"#2",6}" + $"{"#3",6}" + $"{"#4",6}" +

$"{"#5",6}" + $"{"#6",6}" + $"{"#7",6}";

for (int i = 0; i < skate.Length; i++)

label1.Text += $"{i + 1,5}" + $"{skate[i].name,12}" + $"{skate[i].r1,6}" + $"{skate[i].r2,6}" + $"{skate[i].r3,6}" +

$"{skate[i].r4,6}" + $"{skate[i].r5,6}" + $"{skate[i].r6,6}" + $"{skate[i].r7,6}\n";// + $"{sum[i],6}\n";

}

static void place(grades[] x, int[] y)

{

double[,] z = new double[y.Length, 2];

for (int i = 0; i < y.Length; i++)

{

z[i, 0] = i;

z[i, 1] = x[i].r1;

y[i] = 0;

}

sort(z, y.Length);

for (int i = 0; i < y.Length; i++)

y[(int)z[i, 0]] += (i + 1);

for (int i = 0; i < y.Length; i++)

{

z[i, 0] = i;

z[i, 1] = x[i].r2;

}

sort(z, y.Length);

for (int i = 0; i < y.Length; i++)

y[(int)z[i, 0]] += (i + 1);

for (int i = 0; i < y.Length; i++)

{

z[i, 0] = i;

z[i, 1] = x[i].r3;

}

sort(z, y.Length);

for (int i = 0; i < y.Length; i++)

y[(int)z[i, 0]] += (i + 1);

for (int i = 0; i < y.Length; i++)

{

z[i, 0] = i;

z[i, 1] = x[i].r4;

}

sort(z, y.Length);

for (int i = 0; i < y.Length; i++)

y[(int)z[i, 0]] += (i + 1);

for (int i = 0; i < y.Length; i++)

{

z[i, 0] = i;

z[i, 1] = x[i].r5;

}

sort(z, y.Length);

for (int i = 0; i < y.Length; i++)

y[(int)z[i, 0]] += (i + 1);

for (int i = 0; i < y.Length; i++)

{

z[i, 0] = i;

z[i, 1] = x[i].r6;

}

sort(z, y.Length);

for (int i = 0; i < y.Length; i++)

y[(int)z[i, 0]] += (i + 1);

for (int i = 0; i < y.Length; i++)

{

z[i, 0] = i;

z[i, 1] = x[i].r7;

}

sort(z, y.Length);

for (int i = 0; i < y.Length; i++)

y[(int)z[i, 0]] += (i + 1);

}

static void sort(double[,] z, int n)

{

for (int i = 0; i < n; i++)

{

for (int j = i + 1; j < n; j++)

if (z[i, 1] < z[j, 1])

{

double c;

c = z[i, 0];

z[i, 0] = z[j, 0];

z[j, 0] = c;

c = z[i, 1];

z[i, 1] = z[j, 1];

z[j, 1] = c;

}

}

}

struct team

{

public string name;

public int win;

}

void l3t2()

{

Random rand = new Random((int)DateTime.Now.Ticks);

label1.Text+="Сложность 3. Задание 2\n Команды и кол-во из победы на 1-м этапе:\n"+ $"{"Команда",12}" + $"{" Кол-во побед",6}\n";

team[] teams = new team[12];

for (int i = 0; i < teams.Length; i++)

{

teams[i].name = $"Команда{i + 1}";

teams[i].win = rand.Next(1, 7);

label1.Text += $"{teams[i].name,12}" + $"{teams[i].win,6}\n";

}

teams = teams.OrderByDescending(p => p.win).ToArray();

label1.Text += " Участники второго этапа:\n" + $"{"Команда",12}" + $"{" Кол-во побед",6} \n";

for (int i = 0; i < 6; i++)

label1.Text += $"{teams[i].name,12}" + $"{teams[i].win,6} \n";

}

struct competitor

{

public string name;

public int place;

public int team;

}

void l3t3()

{

Random rand = new Random((int)DateTime.Now.Ticks);

label1.Text+="Сложность 3. Задание 3\n"+ $"{"Игрок",12}" + $"{"Место",6}" + $"{"Ком.",6}\n";

competitor[] com = new competitor[18];

int place1 = 0;

for (int i = 0; i < com.Length; i++)

{

com[i].name = $"Участник{i + 1}";

bool f;

do

{

f = true;

com[i].place = rand.Next(1, 18 + 1);

for (int j = 0; j < i; j++)

if (com[i].place == com[j].place)

f = false;

} while (f == false);

if (com[i].place == 1)

place1 = com[i].team;

com[i].team = i % 3 + 1;

label1.Text += $"{com[i].name,12}" + $"{com[i].place,6}" + $"{com[i].team,6}\n";

}

int r1 = 0, r2 = 0, r3 = 0;

points(com, ref r1, ref r2, ref r3);

label1.Text += $"Баллы команды№1: {r1}, команды№2: {r2}, команды№3: {r3}\n";

if ((r1 == r2) && (r2 == r3))

Console.Write("Ничья => победителя нет");

else

{

int[] n = { r1, r2, r3 };

label1.Text += $"Победителем является команда №";

if (((r1 == r2) && (r1 > r3)) || ((r2 == r3) && (r2 > r1)) || ((r1 == r3) && (r1 > r2)))

Console.Write(place1);

else

{

if (n.Max() == r1)

label1.Text += (1);

else

{

if (n.Max() == r2)

label1.Text += (2);

else

{

if (n.Max() == r3)

label1.Text += (3);

}

}

}

label1.Text += $", набравшая {n.Max()} баллов";

}

}

static void points(competitor[] c, ref int x, ref int y, ref int z)

{

for (int i = 0; i < c.Length; i++)

switch (c[i].team)

{

case 1:

if (c[i].place < 6)

x += 6 - c[i].place;

break;

case 2:

if (c[i].place < 6)

y += 6 - c[i].place;

break;

case 3:

if (c[i].place < 6)

z += 6 - c[i].place;

break;

}

}

}

}

# Пример работы программ

### I уровень. Задание 1

### I уровень. Задание 2

### II уровень. Задание 8

### II уровень. Задание 9

### III уровень. Задание 2

### III уровень. Задание 3