Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Практическая 13\_3**»

Выполнил: Карамов Д.Э

Группа: ПР-24

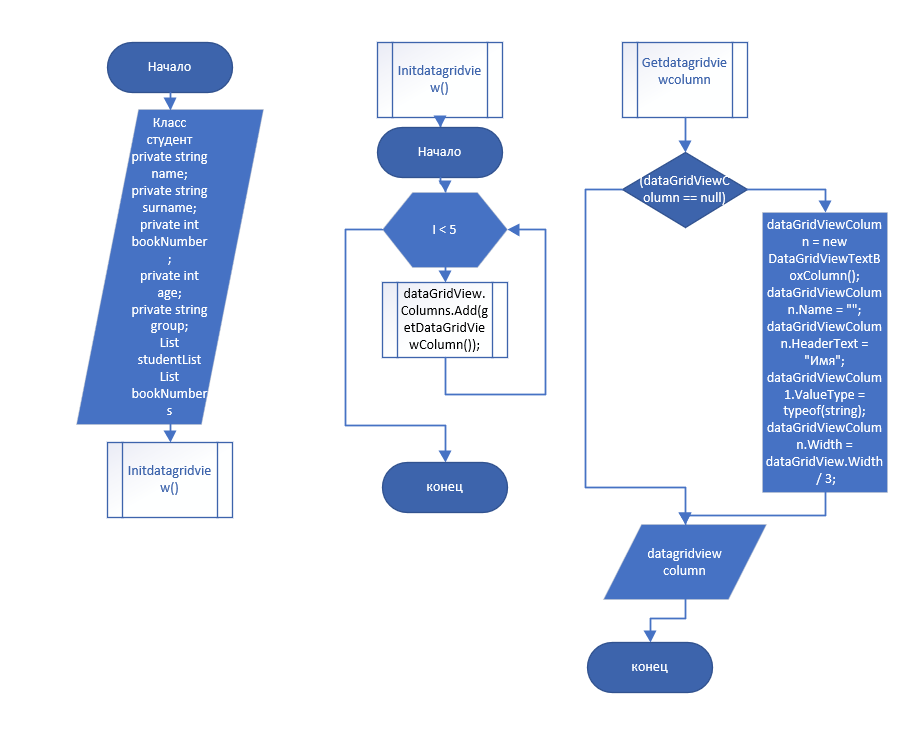
Преподаватель: Мирошниченко Г.В

2025

**Задание:**

Создать приложение для записи, редактирования и удаления компьютеров при помощи SortedDictionary

Инициализация таблицы блок схема



Класс Computer

SortedDictionary Computers

List numbers

Подпрограмма addComputer //заносит компьютер в таблицу

**Входные данные**

Класс Computer

{

private int number; //номер компьютера

private string name; //имя пользователя

private string system; //система

private string cpu; //процессор

private string gpu; //видеокарта

private int ozu; //оперативная память

}

**Выходные данные**

Класс Computer (в таблице)

{

private int number; //номер компьютера

private string name; //имя пользователя

private string system; //система

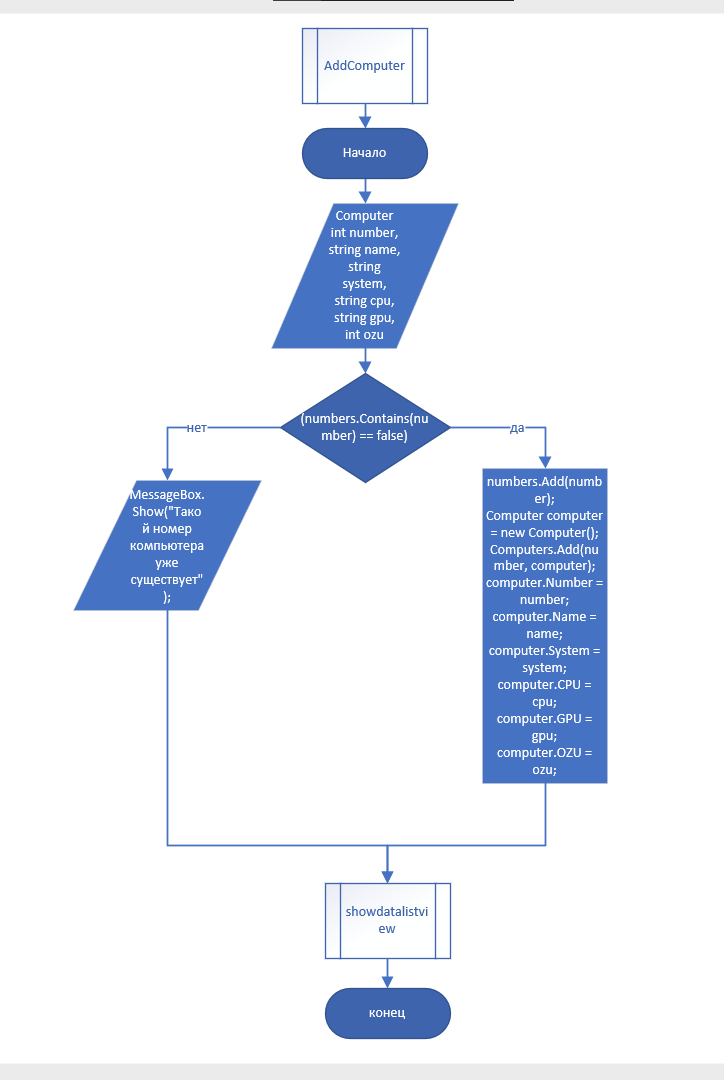
private string cpu; //процессор

private string gpu; //видеокарта

private int ozu; //оперативная память

}

Блок-схема



Подпрограмма deletecomputer //удаляет компьютер из таблицы

**Входные данные**

Класс Computer

{

private int number; //номер компьютера

private string name; //имя пользователя

private string system; //система

private string cpu; //процессор

private string gpu; //видеокарта

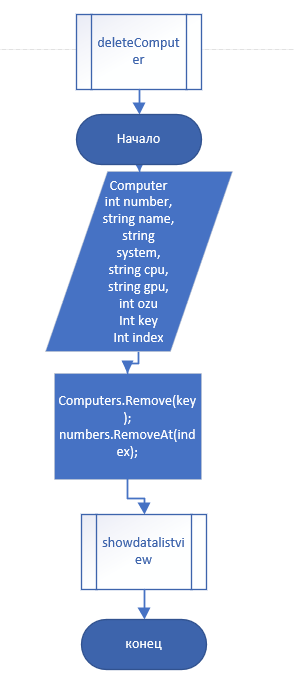
private int ozu; //оперативная память

}

**Выходные данные**

-

Блок схема



Подпрограмма redactComputer //редактирует информацию о компе

**Входные данные**

Класс Computer

{

private int number; //номер компьютера

private string name; //имя пользователя

private string system; //система

private string cpu; //процессор

private string gpu; //видеокарта

private int ozu; //оперативная память

}

**Выходные данные**

Класс Computer

{

private string name; //имя пользователя

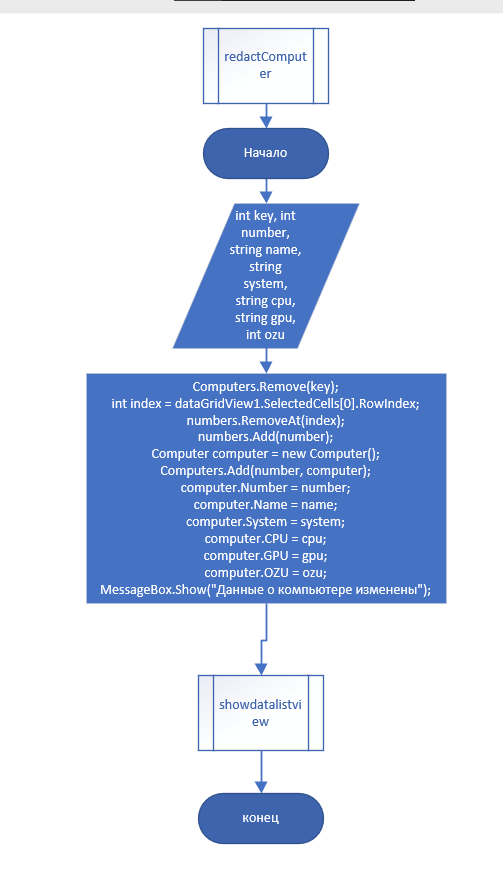
private string system; //система

private string cpu; //процессор

private string gpu; //видеокарта

private int ozu; //оперативная память

}  
блок схема



Подпрограмма showListInGrid Выводит студентов в таблицу

Входные данные:

Класс Computer

{

private string name; //имя пользователя

private string system; //система

private string cpu; //процессор

private string gpu; //видеокарта

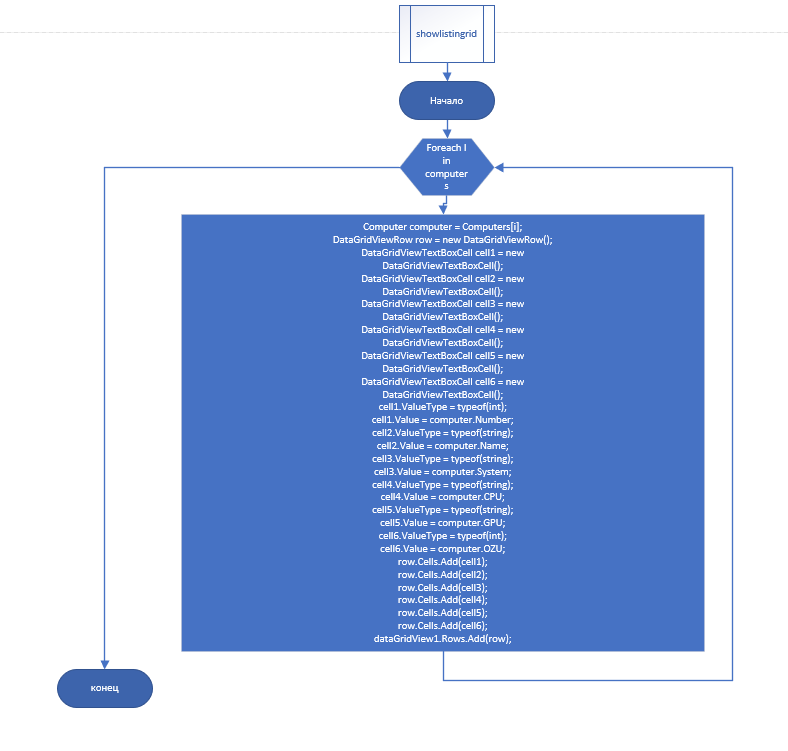
private int ozu; //оперативная память

}

Выходные данные

Datagridview

Блоксхема



Подпрограмма CountOldSystems Выводит количество устаревших систем

Входные данные:

Класс Computer

{

private string system; //система

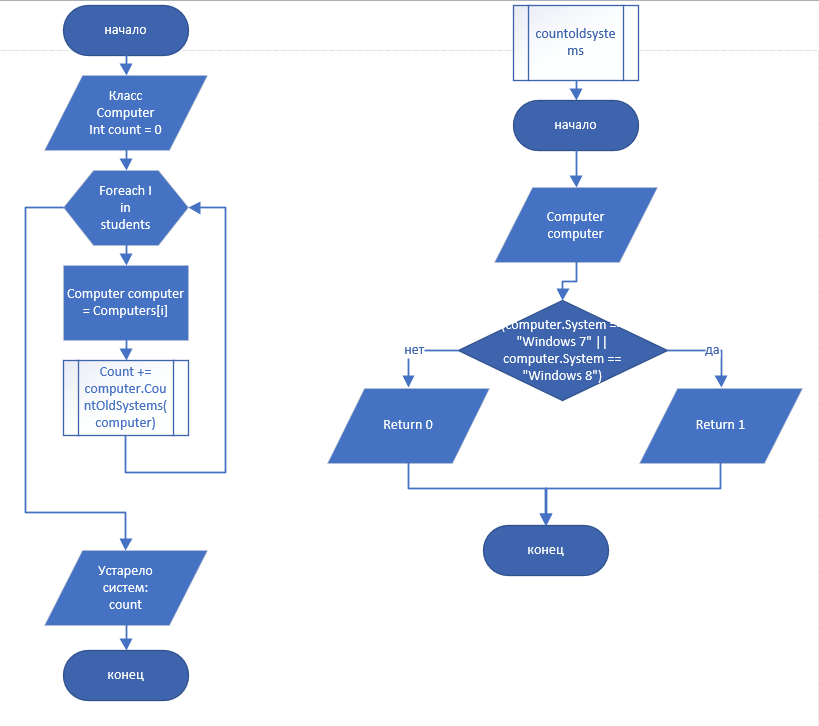
}

Count = 0

Выходные данные

count

Блоксхема

****

**Листинг программы**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Runtime.InteropServices;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace pract13\_3\_karamov

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

initDataGridView();

}

private DataGridViewColumn dataGridViewColumn1 = null;

private DataGridViewColumn dataGridViewColumn2 = null;

private DataGridViewColumn dataGridViewColumn3 = null;

private DataGridViewColumn dataGridViewColumn4 = null;

private DataGridViewColumn dataGridViewColumn5 = null;

private DataGridViewColumn dataGridViewColumn6 = null;

private SortedDictionary<int, Computer> Computers = new SortedDictionary<int, Computer>();

private List<int> numbers = new List<int>();

private void initDataGridView()

{

dataGridView1.DataSource = null;

dataGridView1.Columns.Add(getDataGridViewColumn1());

dataGridView1.Columns.Add(getDataGridViewColumn2());

dataGridView1.Columns.Add(getDataGridViewColumn3());

dataGridView1.Columns.Add(getDataGridViewColumn4());

dataGridView1.Columns.Add(getDataGridViewColumn5());

dataGridView1.Columns.Add(getDataGridViewColumn6());

dataGridView1.AutoResizeColumns();

}

private void showListInGrid()

{

dataGridView1.Rows.Clear();

foreach (int i in Computers.Keys)

{

Computer computer = Computers[i];

DataGridViewRow row = new DataGridViewRow();

DataGridViewTextBoxCell cell1 = new

DataGridViewTextBoxCell();

DataGridViewTextBoxCell cell2 = new

DataGridViewTextBoxCell();

DataGridViewTextBoxCell cell3 = new

DataGridViewTextBoxCell();

DataGridViewTextBoxCell cell4 = new

DataGridViewTextBoxCell();

DataGridViewTextBoxCell cell5 = new

DataGridViewTextBoxCell();

DataGridViewTextBoxCell cell6 = new

DataGridViewTextBoxCell();

cell1.ValueType = typeof(int);

cell1.Value = computer.Number;

cell2.ValueType = typeof(string);

cell2.Value = computer.Name;

cell3.ValueType = typeof(string);

cell3.Value = computer.System;

cell4.ValueType = typeof(string);

cell4.Value = computer.CPU;

cell5.ValueType = typeof(string);

cell5.Value = computer.GPU;

cell6.ValueType = typeof(int);

cell6.Value = computer.OZU;

row.Cells.Add(cell1);

row.Cells.Add(cell2);

row.Cells.Add(cell3);

row.Cells.Add(cell4);

row.Cells.Add(cell5);

row.Cells.Add(cell6);

dataGridView1.Rows.Add(row);

}

}

private DataGridViewColumn getDataGridViewColumn1()

{

if (dataGridViewColumn1 == null)

{

dataGridViewColumn1 = new DataGridViewTextBoxColumn();

dataGridViewColumn1.Name = "";

dataGridViewColumn1.HeaderText = "Номер";

dataGridViewColumn1.ValueType = typeof(int);

dataGridViewColumn1.Width = dataGridView1.Width / 3;

}

return dataGridViewColumn1;

}

private DataGridViewColumn getDataGridViewColumn2()

{

if (dataGridViewColumn2 == null)

{

dataGridViewColumn2 = new DataGridViewTextBoxColumn();

dataGridViewColumn2.Name = "";

dataGridViewColumn2.HeaderText = "Имя пользователя";

dataGridViewColumn2.ValueType = typeof(string);

dataGridViewColumn2.Width = dataGridView1.Width / 3;

}

return dataGridViewColumn2;

}

private DataGridViewColumn getDataGridViewColumn3()

{

if (dataGridViewColumn3 == null)

{

dataGridViewColumn3 = new DataGridViewTextBoxColumn();

dataGridViewColumn3.Name = "";

dataGridViewColumn3.HeaderText = "Система";

dataGridViewColumn3.ValueType = typeof(string);

dataGridViewColumn3.Width = dataGridView1.Width / 3;

}

return dataGridViewColumn3;

}

private DataGridViewColumn getDataGridViewColumn4()

{

if (dataGridViewColumn4 == null)

{

dataGridViewColumn4 = new DataGridViewTextBoxColumn();

dataGridViewColumn4.Name = "";

dataGridViewColumn4.HeaderText = "Процессор";

dataGridViewColumn4.ValueType = typeof(string);

dataGridViewColumn4.Width = dataGridView1.Width / 3;

}

return dataGridViewColumn4;

}

private DataGridViewColumn getDataGridViewColumn5()

{

if (dataGridViewColumn5 == null)

{

dataGridViewColumn5 = new DataGridViewTextBoxColumn();

dataGridViewColumn5.Name = "";

dataGridViewColumn5.HeaderText = "Видеокарта";

dataGridViewColumn5.ValueType = typeof(string);

dataGridViewColumn5.Width = dataGridView1.Width / 3;

}

return dataGridViewColumn5;

}

private DataGridViewColumn getDataGridViewColumn6()

{

if (dataGridViewColumn6 == null)

{

dataGridViewColumn6 = new DataGridViewTextBoxColumn();

dataGridViewColumn6.Name = "";

dataGridViewColumn6.HeaderText = "Память";

dataGridViewColumn6.ValueType = typeof(int);

dataGridViewColumn6.Width = dataGridView1.Width / 3;

}

return dataGridViewColumn6;

}

private void addComputer(int number, string name, string system, string cpu, string gpu, int ozu)

{

if (numbers.Contains(number) == false)

{

numbers.Add(number);

Computer computer = new Computer();

Computers.Add(number, computer);

computer.Number = number;

computer.Name = name;

computer.System = system;

computer.CPU = cpu;

computer.GPU = gpu;

computer.OZU = ozu;

}

else

{

MessageBox.Show("Такой номер компьютера уже существует");

}

showListInGrid();

}

private void deleteComputer(int key, int index)

{

Computers.Remove(key);

numbers.RemoveAt(index);

showListInGrid();

}

private void redactComputer(int key, int number, string name, string system, string cpu, string gpu, int ozu)

{

Computers.Remove(key);

int index = dataGridView1.SelectedCells[0].RowIndex;

numbers.RemoveAt(index);

numbers.Add(number);

Computer computer = new Computer();

Computers.Add(number, computer);

computer.Number = number;

computer.Name = name;

computer.System = system;

computer.CPU = cpu;

computer.GPU = gpu;

computer.OZU = ozu;

MessageBox.Show("Данные о компьютере изменены");

showListInGrid();

}

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

{

if (textBox1.Text != string.Empty && comboBox2.Text != string.Empty && comboBox3.Text != string.Empty && comboBox1.Text != string.Empty)

{

try

{

addComputer(Convert.ToInt32(numericUpDown1.Value), textBox1.Text, comboBox1.Text, comboBox2.Text, comboBox3.Text, Convert.ToInt32(numericUpDown3.Value));

}

catch

{

MessageBox.Show("Неверный формат данных");

}

}

else

{

MessageBox.Show("Все поля должны быть заполнены");

}

}

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

{

try

{

int selectedRow = dataGridView1.SelectedCells[0].RowIndex + 1;

int key = Convert.ToInt32(dataGridView1.SelectedCells[0].Value);

DialogResult dr = MessageBox.Show("Удалить компьютер?", "", MessageBoxButtons.YesNo);

if (dr == DialogResult.Yes)

{

try

{

deleteComputer(key, selectedRow - 1);

}

catch

{

MessageBox.Show("Ошибка");

}

}

}

catch

{

MessageBox.Show("Ошибка, Нужно выбрать ячейку номер");

}

}

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

{

try

{

int key = Convert.ToInt32(dataGridView1.SelectedCells[0].Value);

numericUpDown1.Value = Computers[key].Number;

textBox1.Text = Computers[key].Name;

comboBox1.Text = Computers[key].System;

comboBox2.Text = Computers[key].CPU;

comboBox3.Text = Computers[key].GPU;

numericUpDown3.Value = Computers[key].OZU;

button1.Visible = false;

button1.Enabled = false;

numericUpDown1.Enabled = false;

numericUpDown1.Visible = false;

label1.Visible = false;

button2.Visible = true;

button2.Enabled = true;

}

catch

{

MessageBox.Show("Ошибка, выберите ячейку номер");

}

}

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

{

int selectedRow = dataGridView1.SelectedCells[0].RowIndex + 1;

try

{

int key = Convert.ToInt32(dataGridView1.SelectedCells[0].Value);

redactComputer(key, Convert.ToInt32(numericUpDown1.Value), textBox1.Text, comboBox1.Text, comboBox2.Text, comboBox3.Text, Convert.ToInt32(numericUpDown3.Value));

}

catch

{

MessageBox.Show("Ошибка");

}

button1.Visible = true;

button1.Enabled = true;

button2.Visible = false;

button2.Enabled = false;

numericUpDown1.Enabled = true;

numericUpDown1.Visible = true;

label1.Visible = true;

}

private void button3\_Click(object sender, EventArgs e)

{

int count = 0;

foreach (int i in Computers.Keys)

{

Computer computer = Computers[i];

count += computer.CountOldSystems(computer);

}

label7.Text = $"Устарело систем: {count}";

}

}

}

Класс

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace pract13\_3\_karamov

{

class Computer

{

private int number; //номер компьютера

private string name; //имя пользователя

private string system; //система

private string cpu; //процессор

private string gpu; //видеокарта

private int ozu; //оперативная память

public int Number

{

get { return number; }

set { number = value; }

}

public string Name

{

get { return name; }

set { name = value; }

}

public string System

{

get { return system; }

set { system = value; }

}

public string CPU

{

get { return cpu; }

set { cpu = value; }

}

public string GPU

{

get { return gpu; }

set { gpu = value; }

}

public int OZU

{

get { return ozu; }

set { ozu = value; }

}

public int CountOldSystems(Computer computer)

{

if (computer.System == "Windows 7" || computer.System == "Windows 8")

{

return 1;

}

else

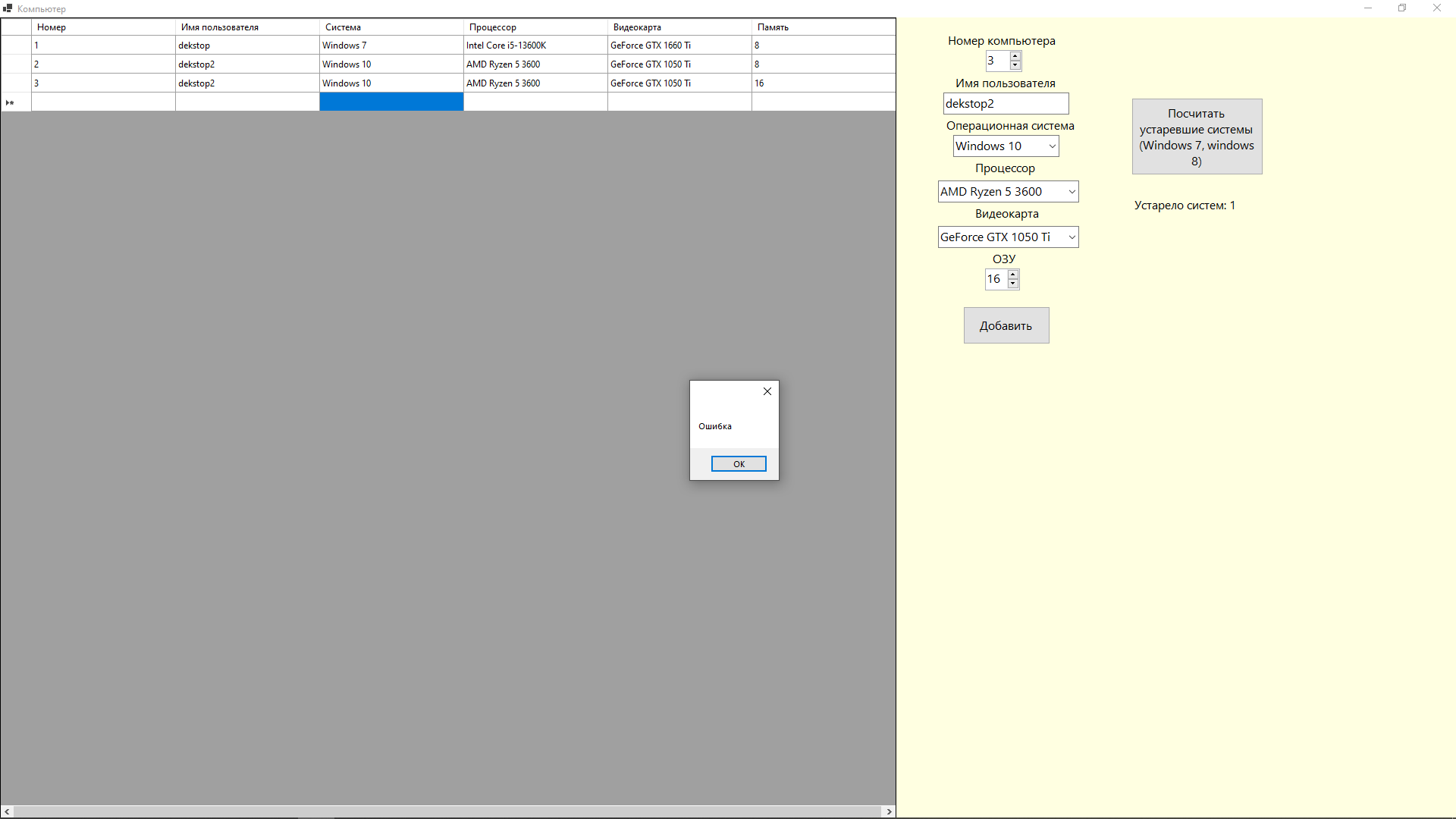
{

return 0;

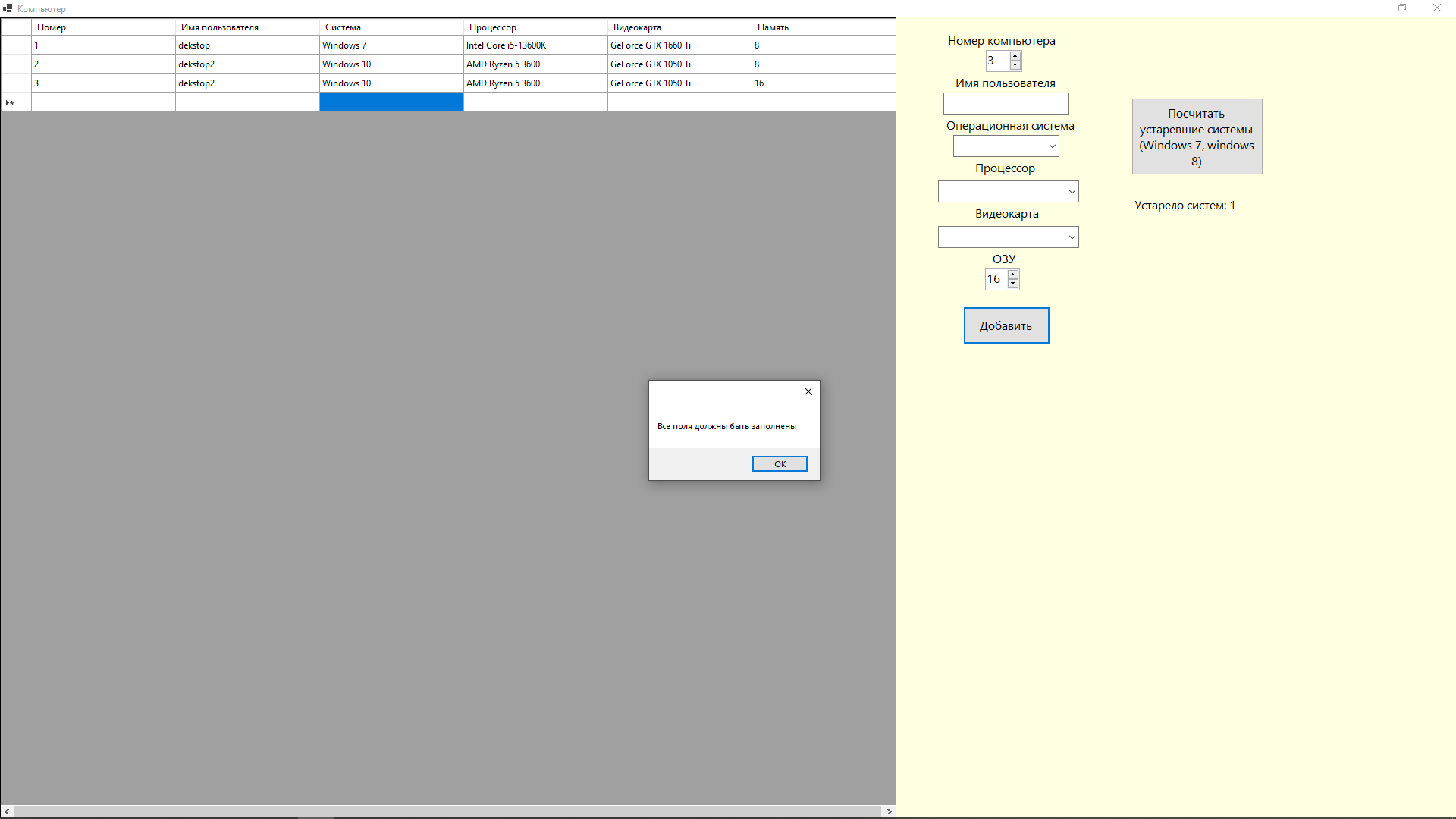
}

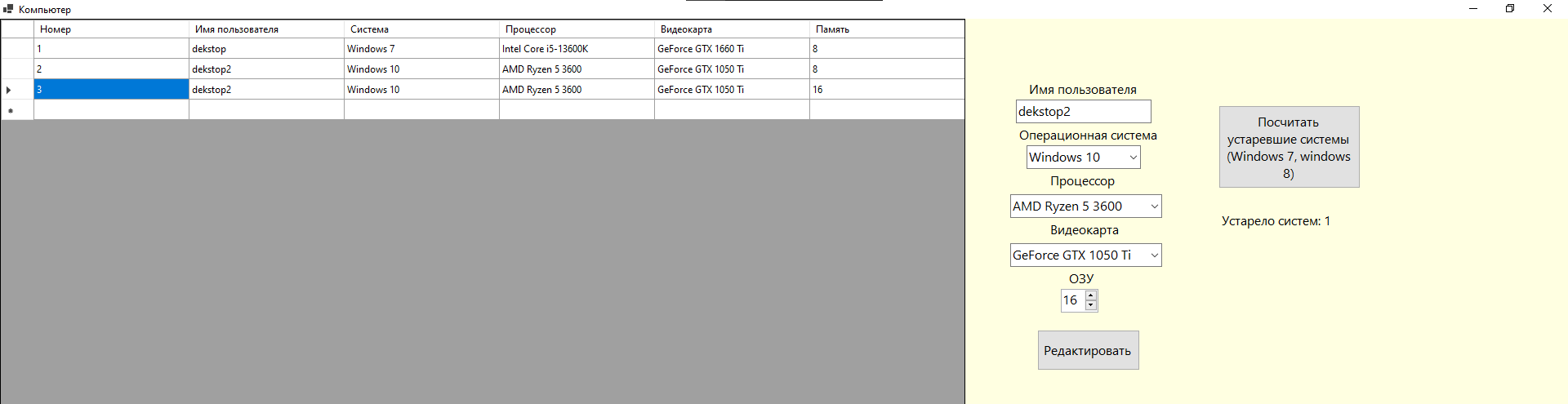
}

**Тестовые ситуации**



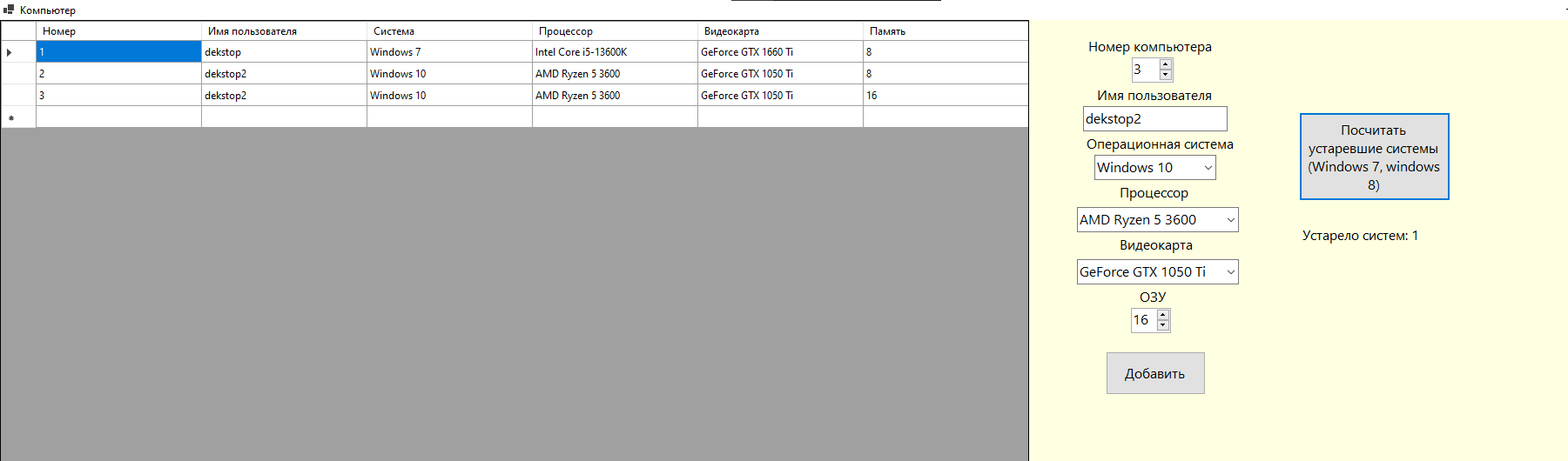
**(удаление студента в пустой строке)**



****

**(Редактирование строчки)**

**Результат**

****

**Вывод:**

**Я научился создавать коллекцию SortedDictionary компьютеров**