**Занятие № 6**

**Дата выполнения работы:** 22.04.2023

# **Тема работы:** «Регулярные выражения»

**Ход работы**

**Задание 1**

Университет: Объект – «Студент». Поля: ФИО, возраст,

специальность, дата рождения, курс, группа, средний балл, пол,

адреса и др. Агрегируемый объект – «Адрес». Поля: город, индекс,

улица, дом, квартира.

**Листинг программы:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Net;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace App1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

public class Address

{

public string City { get; set; }

public int Index { get; set; }

public string Street { get; set; }

public string HouseNumber { get; set; }

public string ApartmentNumber { get; set; }

public bool CheckIndexFormat()

{

return true;

}

public bool CheckCityFormat()

{

return true;

}

public bool CheckStreetFormat()

{

return true;

}

public bool CheckHouseNumber()

{

return true;

}

public bool CheckApartmentNumber()

{

return true;

}

}

public class Student

{

public string FullName { get; set; }

public int Age { get; set; }

public string Specialization { get; set; }

public DateTime DateOfBirth { get; set; }

public int Course { get; set; }

public string Group { get; set; }

public double AverageGrade { get; set; }

public string Gender { get; set; }

public Address Address { get; set; }

public bool CheckFullNameFormat()

{

return true;

}

public bool CheckAgeRange()

{

return true;

}

public bool CheckSpecialization()

{

return true;

}

}

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

{

Student student = new Student

{

FullName = textBox1.Text,

Age = Convert.ToInt32(textBox2.Text),

Specialization = textBox3.Text,

Address = new Address

{

City = textBox4.Text,

Index = Convert.ToInt32(textBox5.Text),

Street = textBox6.Text,

HouseNumber = textBox7.Text,

ApartmentNumber = textBox8.Text

}

};

if (student.CheckFullNameFormat())

{

textBox16.Text = "ФИО студента: " + student.FullName;

}

else

{

textBox16.Text = "Некорректный формат ФИО студента!";

}

if (student.CheckAgeRange())

{

textBox15.Text = "Возраст студента: " + student.Age + " лет";

}

else

{

textBox15.Text = "Некорректный возраст студента!";

}

if (student.CheckSpecialization())

{

textBox14.Text = "Специальность: " + student.Specialization;

}

else

{

textBox14.Text = "Некорректная специльность!";

}

if (student.Address.CheckIndexFormat())

{

textBox12.Text = "Индекс: " + student.Address.Index;

}

else

{

textBox12.Text = "Некорректный формат индекса!";

}

if (student.Address.CheckCityFormat())

{

textBox11.Text = "Улица: " + student.Address.Street;

}

else

{

textBox11.Text = "Некорректный формат улицы!";

}

if (student.Address.CheckHouseNumber())

{

textBox10.Text = "Номер дома: " + student.Address.HouseNumber;

}

else

{

textBox13.Text = "Некорректный формат номера дома!";

}

if (student.Address.CheckStreetFormat())

{

textBox13.Text = "Улица: " + student.Address.Street;

}

else

{

textBox13.Text = "Некорректный формат улицы!";

}

if (student.Address.CheckApartmentNumber())

{

textBox9.Text = "Номер квартиры: " + student.Address.ApartmentNumber;

}

else

{

textBox9.Text = "Некорректный формат номера квартиры!";

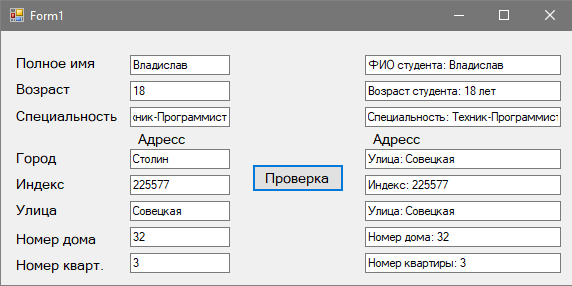
}

}

}

}

**Результат:**



**Задание 2**

Поиск по: ФИО (по шаблону), специальности, курсу, среднему

баллу (>n , диапазон). Сортировка по стажу работы и курсу.

**Листинг программы:**

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 App2

{

public partial class Form1 : Form

{

private List<Student> students;

public Form1()

{

InitializeComponent();

students = new List<Student>();

}

private void RefreshStudentList()

{

// Очистка и обновление списка студентов на форме

listBox1.Items.Clear();

foreach (var student in students)

{

listBox1.Items.Add(student);

}

}

private class Student

{

public string FullName { get; set; }

public string Specialty { get; set; }

public int Course { get; set; }

public double GPA { get; set; }

public int YearsOfWorkExperience { get; set; }

public Student(string fullName, string specialty, int course, double gpa, int yearsOfWorkExperience)

{

FullName = fullName;

Specialty = specialty;

Course = course;

GPA = gpa;

YearsOfWorkExperience = yearsOfWorkExperience;

}

public override string ToString()

{

return $"{FullName} - {Specialty}, Курс {Course}, Средний балл: {GPA}, Стаж работы: {YearsOfWorkExperience} год(а)";

}

}

private void button1\_Click\_1(object sender, EventArgs e)

{

string searchPattern = textBoxSearchPattern.Text;

string searchSpecialty = textBoxSearchSpecialty.Text;

int searchCourse = Convert.ToInt32(numericUpDown1.Value);

double searchMinGPA = Convert.ToDouble(textBoxSearchMinGPA.Text);

double searchMaxGPA = Convert.ToDouble(textBoxSearchMaxGPA.Text);

// Фильтрация студентов по параметрам поиска

var filteredStudents = students.Where(student =>

student.FullName.Contains(searchPattern) &&

student.Specialty.Contains(searchSpecialty) &&

student.Course == searchCourse &&

student.GPA >= searchMinGPA &&

student.GPA <= searchMaxGPA

);

// Сортировка студентов по стажу работы и курсу

filteredStudents = filteredStudents.OrderBy(student => student.YearsOfWorkExperience)

.ThenBy(student => student.Course);

// Обновление списка студентов на форме

listBox1.Items.Clear();

foreach (var student in filteredStudents)

{

listBox1.Items.Add(student);

}

}

private void button2\_Click\_1(object sender, EventArgs e)

{

// Очистка всех полей ввода

textBoxSearchPattern.Text = string.Empty;

textBoxSearchSpecialty.Text = string.Empty;

numericUpDown1.Value = 0;

textBoxSearchMinGPA.Text = string.Empty;

textBoxSearchMaxGPA.Text = string.Empty;

RefreshStudentList(); // Обновление списка студентов на форме

}

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

{

students.Add(new Student("Иванов Иван Иванович", "Математика", 3, 4.5, 2));

students.Add(new Student("Петров Петр Петрович", "Физика", 2, 3.8, 1));

students.Add(new Student("Сидорова Елена Викторовна", "История", 4, 4.2, 3));

students.Add(new Student("Смирнов Сергей Александрович", "Филология", 1, 3.5, 0));

students.Add(new Student("Кузнецова Ольга Владимировна", "Экономика", 3, 4.0, 1));

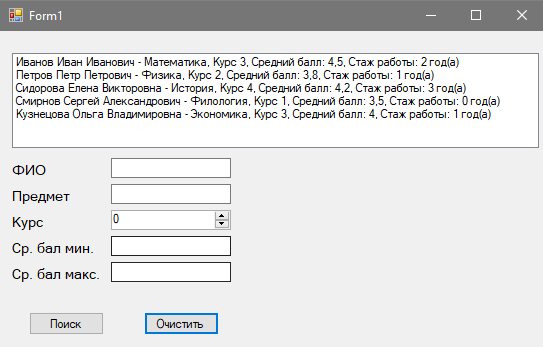
RefreshStudentList();

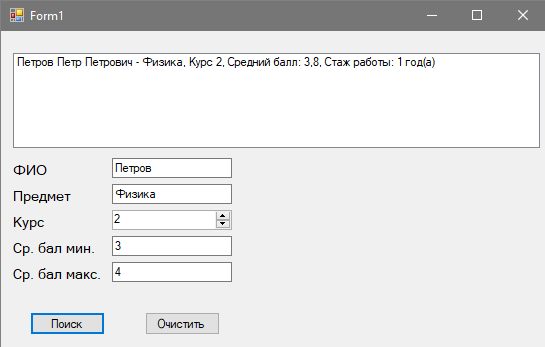
}

}

}

**Результат:**



****

**Вывод:** Прохождение курса по регулярным выражениям позволило мне, как пользователю, осознать важность и полезность этого инструмента в разработке на C# с использованием Windows Forms. Регулярные выражения позволяют мне выполнять сложные операции по поиску, фильтрации и изменению текстовых данных на основе шаблонов. Однако, они также требуют тщательной проверки и тестирования, чтобы избежать ошибок. В целом, знание регулярных выражений стало ценным дополнением к моим навыкам разработки, позволяющим более эффективно работать с текстовыми данными в моих проектах на C# с использованием Windows Forms.