МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ИНЖЕНЕРНО-ТЕХНОЛОГИЧЕСКАЯ АКАДЕМИЯ

Институт компьютерных технологий и информационной безопасности

Кафедра математического обеспечения и применения ЭВМ

**Лабораторная работа № 6**

по курсу «Объектно-ориентированное программирование»

**«Создание графического интерфейса в С#»**

**15 ВАРИАНТ**

Выполнил:

студенты гр. КТбо2-1

Самардак А.В.

Проверил:

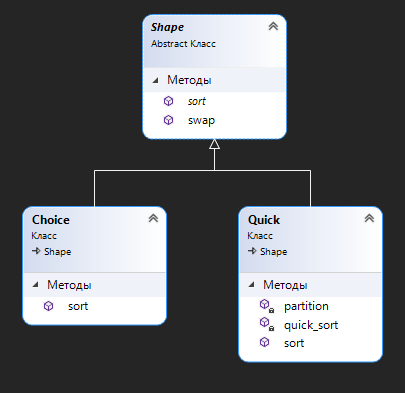
Тарасов С.А/ Быстров Э.А

**Таганрог 2020**

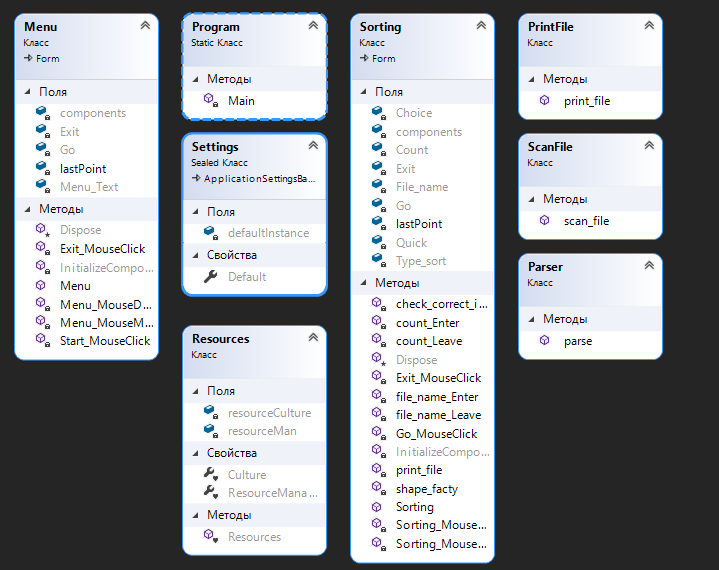
1. **Вариант задания**

Создать класс Sorting(сортировка), и на его основе классы Choice(сортировка выбором) и Quick(быстрая сорти-ровка). Размер сортируемых массивов задается при их создании, а элементы считываются из файла.

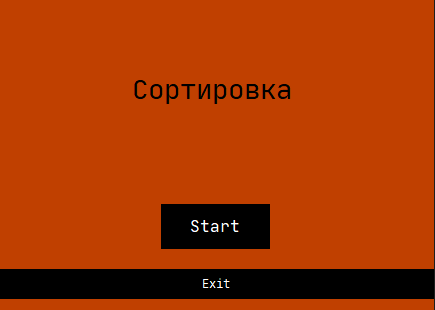
1. **UML-диаграмма классов: Shape, Quick, Choice**



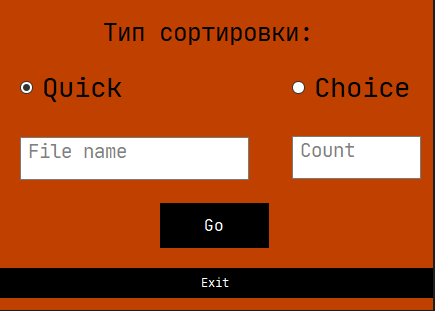
1. **UML-диаграмма классов: Menu, PrintFile, ScanFile, Parser, Program, Settings, Resources, Sorting**



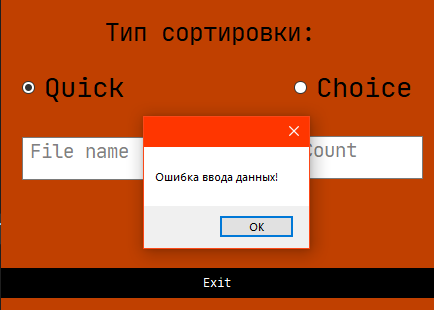
1. **Внешний вид**



Menu



Sorting



Ошибка ввода данных в Sorting

1. **Листинг программы на языке С#**
2. **Menu.cs**

using System;

using System.Drawing;

using System.Windows.Forms;

namespace OOP\_Laba6

{

public partial class Menu : Form

{

public Menu()

{

InitializeComponent();

}

private void Exit\_MouseClick(object sender, MouseEventArgs e)

{

Application.Exit();

}

private Point lastPoint;

private void Menu\_MouseMove(object sender, MouseEventArgs e)

{

if (e.Button == MouseButtons.Left)

{

this.Left += e.X - lastPoint.X;

this.Top += e.Y - lastPoint.Y;

}

}

private void Menu\_MouseDown(object sender, MouseEventArgs e)

{

lastPoint = new Point(e.X, e.Y);

}

private void Start\_MouseClick(object sender, MouseEventArgs e)

{

this.Hide();

Sorting sort = new Sorting();

sort.Show();

}

}

}

1. **Sorting.cs**

using System;

using System.Drawing;

using System.Windows.Forms;

namespace OOP\_Laba6

{

public partial class Sorting : Form

{

public Sorting()

{

InitializeComponent();

}

private void Exit\_MouseClick(object sender, MouseEventArgs e)

{

Application.Exit();

}

private Point lastPoint;

private void Sorting\_MouseMove(object sender, MouseEventArgs e)

{

if (e.Button == MouseButtons.Left)

{

this.Left += e.X - lastPoint.X;

this.Top += e.Y - lastPoint.Y;

}

}

private void Sorting\_MouseDown(object sender, MouseEventArgs e)

{

lastPoint = new Point(e.X, e.Y);

}

private void Go\_MouseClick(object sender, MouseEventArgs e)

{

try

{

check\_correct\_input();

string name = File\_name.Text;

int size = Parser.parse(Count.Text);

int[] arr = ScanFile.scan\_file(ref name, size);

shape\_facty(ref arr);

print\_file(ref arr);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void file\_name\_Enter(object sender, EventArgs e)

{

if (File\_name.Text == "File name")

{

File\_name.Text = "";

File\_name.ForeColor = Color.Black;

}

}

private void file\_name\_Leave(object sender, EventArgs e)

{

if (File\_name.Text == "")

{

File\_name.Text = "File name";

File\_name.ForeColor = Color.Gray;

}

}

private void count\_Enter(object sender, EventArgs e)

{

if (Count.Text == "Count")

{

Count.Text = "";

Count.ForeColor = Color.Black;

}

}

private void count\_Leave(object sender, EventArgs e)

{

if (Count.Text == "")

{

Count.Text = "Count";

Count.ForeColor = Color.Gray;

}

}

private void check\_correct\_input()

{

if (!(File\_name.Text != "File name" && Count.Text != "Count"))

{

throw new Exception("Ошибка ввода данных!");

}

}

private void shape\_facty(ref int[] arr)

{

if (Quick.Checked)

{

Shape shape = new Quick();

shape.sort(ref arr);

}

else

{

Shape shape = new Choice();

shape.sort(ref arr);

}

}

private void print\_file(ref int[] arr)

{

PrintFile.print\_file(ref arr);

MessageBox.Show(String.Format("Успешно, файл будет выгружен в корневую папку проекта!"));

}

}

}

1. **ScanFile.cs**

using System;

using System.IO;

using System.Text;

namespace OOP\_Laba6

{

class ScanFile

{

static public int[] scan\_file(ref string name\_file, int size)

{

if (File.Exists(@name\_file))

{

string line;

int[] arr = new int[size];

using (StreamReader stream\_reader = new StreamReader(@name\_file))

{

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

{

if ((line = stream\_reader.ReadLine()) == null)

{

throw new Exception("Недопустимое количество элементов в файле!... Завершение работы");

}

else

{

arr[i] = Parser.parse(ref line);

}

}

}

return arr;

}

else

{

throw new Exception("Нет такого файла!... Завершение работы");

}

}

}

}

1. **Shape.cs**

using System;

namespace OOP\_Laba6

{

abstract class Sorting

{

abstract public void sort(ref int[] arr);

public void swap(ref int a, ref int b)

{

int temp = a;

a = b;

b = temp;

}

}

}

1. **Quick.cs**

using System;

namespace OOP\_Laba6

{

class Quick: Sorting

{

public override void sort(ref int[] arr)

{

quick\_sort(ref arr, 0, arr.Length - 1);

}

private void quick\_sort(ref int[] arr, int minIndex, int maxIndex)

{

if (minIndex >= maxIndex)

{

return;

}

var pivotIndex = partition(ref arr, minIndex, maxIndex);

quick\_sort(ref arr, minIndex, pivotIndex - 1);

quick\_sort(ref arr, pivotIndex + 1, maxIndex);

}

private int partition(ref int[] arr, int minIndex, int maxIndex)

{

var pivot = minIndex - 1;

for (var i = minIndex; i < maxIndex; i++)

{

if (arr[i] < arr[maxIndex])

{

pivot++;

swap(ref arr[pivot], ref arr[i]);

}

}

pivot++;

swap(ref arr[pivot], ref arr[maxIndex]);

return pivot;

}

}

}

1. **Choice.cs**

using System;

namespace OOP\_Laba6

{

class Choice: Sorting

{

public override void sort(ref int[] arr)

{

int size = arr.Length;

for (int i = 0; i < size - 1; ++i)

{

int i\_min = i;

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

{

if (arr[j] < arr[i\_min])

{

i\_min = j;

}

}

swap(ref arr[i], ref arr[i\_min]);

}

}

}

}

1. **PrinfFile.cs**

using System;

using System.IO;

using System.Text;

namespace OOP\_Laba6

{

class PrintFile

{

static public void print\_file(ref int[] arr)

{

string path = @"result.txt";

using (StreamWriter fout = new StreamWriter(path, false))

{

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

{

fout.Write($"{arr[i]}\n");

}

}

}

}

}

1. **Parser.cs**

using System;

namespace OOP\_Laba6

{

class Parser

{

static public int parse(string data)

{

return Convert.ToInt32(data);

}

}

}