**9 Работа с файлами**

Задание 1. Запись данных в несколько файлов в зависимости от возраста.

Классы:

1. Person – модель (Name, Age).

2. PersonFileReader – читает данные.

3. PersonProcessor – группировка.

Методы:

* ReadPeople() – загружает людей.
* GroupByAge() – группирует по возрасту (моложе 18, от 18 до 40, старше 40).

Листинг задачи:

namespace Task3

{

public class Person

{

public string Name { get; set; }

public int Age { get; set; }

public Person(string name, int age)

{

Name = name;

Age = age;

}

public override string ToString()

{

return $"{Name}, {Age}";

}

}

}

namespace Task3

{

public class PersonFileReader

{

private readonly string \_filePath;

public PersonFileReader(string filePath)

{

\_filePath = filePath;

}

public List<Person> ReadPeople()

{

List<Person> people = new List<Person>();

try

{

if (File.Exists(\_filePath))

{

string[] lines = File.ReadAllLines(\_filePath);

foreach (string line in lines)

{

string[] parts = line.Split(',');

if (parts.Length == 2)

{

string name = parts[0].Trim();

if (int.TryParse(parts[1].Trim(), out int age))

{

people.Add(new Person(name, age));

}

else

{

Console.WriteLine($"Некорректный возраст в строке: {line}");

}

}

else

{

Console.WriteLine($"Некорректный формат строки: {line}");

}

}

}

else

{

Console.WriteLine($"Файл не найден: {\_filePath}");

}

}

catch (Exception ex)

{

Console.WriteLine($"Ошибка при чтении файла: {ex.Message}");

}

return people;

}

}

}

namespace Task3

{

public class PersonProcessor

{

public Dictionary<string, List<Person>> GroupByAge(List<Person> people)

{

Dictionary<string, List<Person>> groups = new Dictionary<string, List<Person>>();

groups["моложе 18"] = people.Where(p => p.Age < 18).ToList();

groups["от 18 до 40"] = people.Where(p => p.Age >= 18 && p.Age <= 40).ToList();

groups["старше 40"] = people.Where(p => p.Age > 40).ToList();

return groups;

}

}

} using Task3;

class Program

{

static void Main(string[] args)

{

string filePath = "C:/Users/37529/Desktop/практика C#/Новик\_09/Task2/bin/Debug/net8.0/file.data";

PersonFileReader reader = new PersonFileReader(filePath);

List<Person> people = reader.ReadPeople();

PersonProcessor processor = new PersonProcessor();

Dictionary<string, List<Person>> ageGroups = processor.GroupByAge(people);

foreach (var group in ageGroups)

{

Console.WriteLine($"\n{group.Key}:");

foreach (Person person in group.Value)

{

Console.WriteLine($" {person.Name}, {person.Age}");

}

}

Console.ReadKey();

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Петр,25; Анна,17; Дмитрий,40; Елена,16; Борис,17 | моложе 18: Анна,17; Елена,16; Борис,17  от 18 до 40: Петр,25; Дмитрий,40  старше 40: |

Анализ результатов:

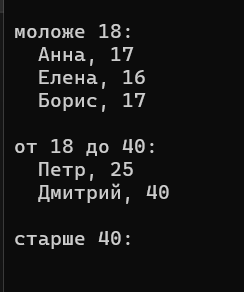


Рисунок 1.1 – Результат работы программы