1. **Основы работы с файлами**

**Пространство имен:**

using System.IO; // Подключите для работы с файлами

**Классы для работы с файлами:**

* File — статические методы для чтения/записи.
* StreamWriter и StreamReader — потоковая запись и чтение.
* FileStream — низкоуровневый доступ.

2. **Запись в файл**

**Метод 1: Использование**File.WriteAllText

string path = "notes.txt";

string text = "Привет, мир!";

File.WriteAllText(path, text); // Создаст или перезапишет файл

**Метод 2: Потоковая запись через**StreamWriter

using (StreamWriter writer = new StreamWriter(path, append: true)) // append: true — дописывает в конец

{

writer.WriteLine("Первая строка");

writer.WriteLine("Вторая строка");

} // using автоматически закрывает поток

3. **Чтение из файла**

**Метод 1:**File.ReadAllText

csharp

Copy

string content = File.ReadAllText(path);

Console.WriteLine(content);

**Метод 2: Потоковое чтение через**StreamReader

using (StreamReader reader = new StreamReader(path))

{

string line;

while ((line = reader.ReadLine()) != null)

{

Console.WriteLine(line);

}

}

4. **Обработка исключений**

**Типичные ошибки:**

* Файл не существует.
* Нет прав доступа.
* Неверный путь.

**Пример с**try-catch**:**

try

{

string content = File.ReadAllText("missing\_file.txt");

}

catch (FileNotFoundException)

{

Console.WriteLine("Файл не найден!");

}

catch (IOException ex)

{

Console.WriteLine($"Ошибка ввода-вывода: {ex.Message}");

}

**Практическое задание**

**Задача 1: Дневник настроения**

Создайте программу, которая:

1. Запрашивает у пользователя текущее настроение.
2. Сохраняет его в файл mood\_diary.txt с датой и временем.
3. При запуске показывает историю настроений.

**Пример кода:**

string path = "mood\_diary.txt";

Console.Write("Опишите ваше настроение: ");

string mood = Console.ReadLine();

string entry = $"{DateTime.Now}: {mood}";

File.AppendAllText(path, entry + Environment.NewLine); // Добавляет запись

Console.WriteLine("История настроений:\n" + File.ReadAllText(path));

**Задача 2: Учет задач (To-Do List)**

Модифицируйте список задач из урока 4, добавив сохранение в файл:

List<string> tasks = new List<string>();

string tasksFile = "tasks.txt";

// Загрузка задач из файла при запуске

if (File.Exists(tasksFile))

{

tasks = File.ReadAllLines(tasksFile).ToList();

}

while (true)

{

// ... (логика добавления/удаления задач)

// При выходе сохраняем задачи:

File.WriteAllLines(tasksFile, tasks);

}

**Задача 3: Шифровщик текста**

Напишите программу, которая:

1. Читает текст из файла input.txt.
2. «Шифрует» его (например, заменяет каждый символ на следующий в Unicode).
3. Сохраняет результат в output.txt.

**Пример:**

string inputPath = "input.txt";

string outputPath = "output.txt";

string text = File.ReadAllText(inputPath);

string encryptedText = "";

foreach (char c in text)

{

encryptedText += (char)(c + 1); // Простой шифр

}

File.WriteAllText(outputPath, encryptedText);