**Курс:** Шаблоны проектирования приложений

**Тема:** Модуль 03 Базовые принципы проектирования

*Цель задания:*

Произведите корректную (правильную) по вашему мнению реализацию с применением принципа DRY:

**Использование параметризованных методов**

public void LogError(string message)

{

Console.WriteLine($"ERROR: {message}");

}

public void LogWarning(string message)

{

Console.WriteLine($"WARNING: {message}");

}

public void LogInfo(string message)

{

Console.WriteLine($"INFO: {message}");

}

**Использование общих конфигурационных настроек**

public class DatabaseService

{

public void Connect()

{

string connectionString = "Server=myServer;Database=myDb;User Id=myUser;Password=myPass;";

// Логика подключения к базе данных

}

}

public class LoggingService

{

public void Log(string message)

{

string connectionString = "Server=myServer;Database=myDb;User Id=myUser;Password=myPass;";

// Логика записи лога в базу данных

}

}

Произведите корректную (правильную) по вашему мнению реализацию с применением принципа KISS:

**Избегание ненужного вложения кода**

public void ProcessNumbers(int[] numbers)

{

if (numbers != null)

{

if (numbers.Length > 0)

{

foreach (var number in numbers)

{

if (number > 0)

{

Console.WriteLine(number);

}

}

}

}

}

**Избегание ненужного использования LINQ**

public void PrintPositiveNumbers(int[] numbers)

{

var positiveNumbers = numbers.Where(n => n > 0).OrderBy(n => n).ToList();

foreach (var number in positiveNumbers)

{

Console.WriteLine(number);

}

}

**Избегание избыточного использования исключений**

public int Divide(int a, int b)

{

try

{

return a / b;

}

catch (DivideByZeroException)

{

return 0;

}

}

Произведите корректную (правильную) по вашему мнению реализацию с применением принципа YAGNI:

**Создание многофункционального класса**

public class User

{

public string Name { get; set; }

public string Email { get; set; }

public string Address { get; set; }

public void SaveToDatabase()

{

// Код для сохранения пользователя в базу данных

}

public void SendEmail()

{

// Код для отправки электронного письма пользователю

}

public void PrintAddressLabel()

{

// Код для печати адресного ярлыка

}

}

**Добавление ненужных настроек или конфигураций**

public class FileReader

{

public string ReadFile(string filePath, bool useBuffer = true, int bufferSize = 1024)

{

// Логика чтения файла с возможностью использования буфера

if (useBuffer)

{

// Чтение с буфером

}

else

{

// Чтение без буфера

}

return "file content";

}

}

**Добавление ненужных методов и функций**

public class ReportGenerator

{

public void GeneratePdfReport()

{

// Генерация PDF отчета

}

public void GenerateExcelReport()

{

// Генерация Excel отчета

}

public void GenerateHtmlReport()

{

// Генерация HTML отчета

}

}