**Осуществить рефакторинг программного кода, используя Extract Method. Описать процесс рефакторинга. Необходимо модифицировать метод Main() в предложенном программном коде. Предполагается, что возникло желание сделать так, чтобы данное приветственное сообщение отображалось в различных местах по всей программе.**

Для рефакторинга предложенного кода с использованием метода Extract Method мы выделим часть кода, отвечающую за вывод приветственного сообщения, в отдельный метод. Это позволит нам легко вызывать это сообщение в различных местах программы, если это потребуется в будущем.

Процесс рефакторинга

Определение кода для извлечения: Мы определим, что часть кода, отвечающая за вывод приветственного сообщения, должна быть вынесена в отдельный метод. Это включает в себя строки, которые настраивают консоль и выводят сообщение.

Создание нового метода: Мы создадим новый метод, который будет отвечать за вывод приветственного сообщения. Этот метод будет иметь тип void и не будет принимать никаких параметров.

**Исправленный код**

using System;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

// Настраиваем консольный интерфейс (CUI)

Console.Title = "Мое приложение";

Console.ForegroundColor = ConsoleColor.Yellow;

Console.BackgroundColor = ConsoleColor.Blue;

// Вызов метода для отображения приветственного сообщения

DisplayGreeting();

Console.BackgroundColor = ConsoleColor.Black;

Console.ReadLine();

}

// Новый метод для отображения приветственного сообщения

static void DisplayGreeting()

{

Console.WriteLine("Привет, это мой проект!");

}

}

}

**Описание применения метода Extract Method**

Выделение кода: Мы выделили код, который отвечает за вывод приветственного сообщения, в отдельный метод DisplayGreeting. Это позволяет изолировать логику отображения сообщения от остального кода, что делает его более читаемым и поддерживаемым.

Упрощение метода Main: Метод Main стал более лаконичным и понятным. Теперь он отвечает только за настройку консоли и вызов метода для отображения приветствия, что улучшает его читаемость.

Повторное использование: Теперь, если в будущем потребуется вывести приветственное сообщение в другом месте программы, мы можем просто вызвать метод DisplayGreeting(), что уменьшает дублирование кода и упрощает его поддержку.

Тестирование: После рефакторинга мы проверили, что программа по-прежнему работает корректно и выводит ожидаемое сообщение, что подтверждает правильность рефакторинга.