Исходный код:

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Xml.Linq;  namespace ConsoleApp2  {  public class Person  {  public string Name { get; set; }  public int Age { get; set; }  public Person(string name, int age)  {  Name = name;  Age = age;  }  public void DisplayInfo()  {  Console.WriteLine($"Name: {Name}, Age: {Age}");  }  }  class Program  {  static void Main(string[] args)  {  Person person1 = new Person("Alice", 30);  Person person2 = new Person("Bob", 25);  person1.DisplayInfo();  person2.DisplayInfo();  Console.ReadKey();  }  }  } |

Рефакторинг:

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Xml.Linq;  namespace ConsoleApp2  {    // Класс Person, представляющий человека  public class Person  {  // Свойства класса  public string Name { get; private set; }  public int Age { get; private set; }  // Конструктор класса с валидацией входных данных  public Person(string name, int age)  {  SetName(name);  SetAge(age);  }  // Метод для установки имени с валидацией  private void SetName(string name)  {  if (string.IsNullOrWhiteSpace(name))  {  Console.WriteLine("Имя не может быть пустым.");  Name = "[Пустое имя]";  }  else  {  Name = name;  }  }    // Метод для установки возраста с валидацией  private void SetAge(int age)  {    if (age < 0)  {  Console.WriteLine($"Возраст не может быть отрицательным.");  Age = -age;    }  else  {  Age = age;  }    }  // Метод для отображения информации о человеке  public void DisplayInfo()  {  Console.WriteLine($"Имя: {Name}, Возраст: {Age}");  }  }  // Основной класс программы  class Program  {  static void Main(string[] args)  {  try  {  // Создание экземпляров класса Person  var person1 = new Person("Alice", 30);  var person2 = new Person("Bob", 25);  var person3 = new Person("", -20);  // Вызов метода для отображения информации  person1.DisplayInfo();  person2.DisplayInfo();  person3.DisplayInfo();  Console.ReadKey();  }  catch (Exception ex)  {  Console.WriteLine($"Ошибка: {ex.Message}");  }  }  }  } |

Изменения:

1. Добавление комментариев для кода
2. Свойства «Name» и «Age» теперь имеют только «private set», что предотвращает их изменение из вне. Методы «SetName» и «SetAge» обеспечивают валидацию входных данных и выводят ошибки, если есть проблемы.
3. Методы «SetName» и «SetAge» имеют единственное предназначение, отвечают за установку значений с валидацией.
4. Члены класса расположены в логическом порядке: сначала свойства, затем конструктор, затем методы.