**4 ЛАБОРАТОРНАЯ РАБОТА №4**

**«Исследование типов данных, определяемых пользователем.**

**Наследование и интерфейсы.»**

**4.1 Цель работы**

Изучить концепцию интерфейсов в языке C#, их назначение и основные принципы использования. Освоить реализацию интерфейсов в классах и структурах, а также применение полиморфизма для обеспечения гибкости и расширяемости программного кода.

**4.2 Индивидуальный вариант**

Разработать класс посетителя отеля. С помощью интерфейсов определить поля (идентификатор, ФИО, номер комнаты), а также методы «Сделать заказ» и «Выезд из отеля».

**4.3 Ход выполнения работы**

В начале лабораторной работы были изучены методические указания. Далее были написаны интерфейсы и классы по варианту задания, код которых содержится в листингах 4.1­–4..

Листинг 4.1 – Интерфейс идентифицируемой сущности

public interface IIdentifieble

{

Guid Id { get; }

}

Листинг 4.2 – Интерфейс именуемой сущности

public interface INameble

{

string FIO { get; set; }

}

Листинг 4.3 – Интерфейс сущности посетителя отеля

public interface IResident

{

int? Room { get; }

void SetRoom(int room);

void ResetRoom();

}

Листинг 4.4 – Класс человека

public class Person(string fio) : IIdentifieble, INameble

{

public Guid Id { get; } = Guid.NewGuid();

public string FIO { get; set; } = fio;

}

Листинг 4.5 – Класс посетителя отеля

public class Resident(string fio) : Person(fio), IResident

{

public int? Room { get; private set; }

public void SetRoom(int room)

{

Room = room;

Console.WriteLine($"Resident {FIO} is in room {Room} now");

}

public void ResetRoom()

{

Room = null;

Console.WriteLine($"Resident {FIO} is homeless now");

}

}

Далее была написана программа, которая тестирует вышеописанные классы и интерфейсы, что показано на листинге 4.6.

Листинг 4.6 – Код основной программы

var p1 = new Person("Sasha");

var r1 = new Resident("Dima");

r1.SetRoom(1);

var r2 = new Resident("Grisha");

r2.SetRoom(2);

r1.ResetRoom();

r2.ResetRoom();

Console.WriteLine($"Is Person Identifieble: {p1 is IIdentifieble}, is Nameble: {p1 is INameble}, is Resident: {p1 is IResident}");

Console.WriteLine($"Is Resident Identifieble: {r1 is IIdentifieble}, is Nameble: {r1 is INameble}, is Person: {r1 is Person}");

Далее программа была протестирована, что показано на рисунке 4.1.

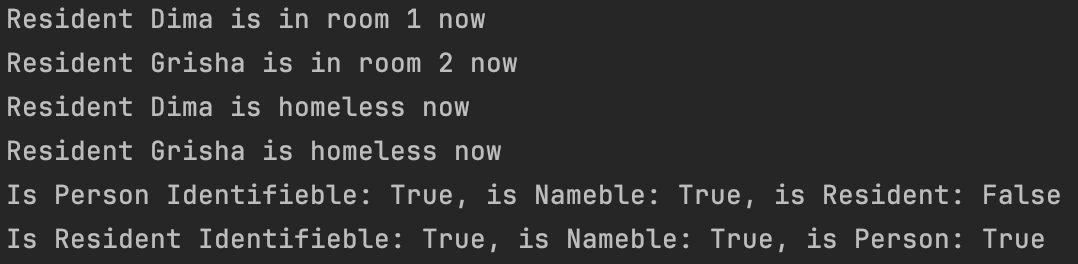


Рисунок 4.1 – Тест программы

**Выводы**

В начале выполнения работы были изучены методические указания. Была изучена концепция интерфейсов в языке C#, их назначение и основные принципы использования. Также была освоена реализация интерфейсов в классах и структурах, а также применение полиморфизма для обеспечения гибкости и расширяемости программного кода.