**Цель работы:** научить применять структурные шаблоны для создания программной системы.

**Вариант 25**

**Выполнение заданий:**

Разработать библиотеку классов, которая содержит указанные классы (таблица 22.1), задействованные в шаблоне Адаптер.

Для адаптера объектов атрибуты класса должны быть реализованы как автоматические свойства, а для шаблона Адаптер классов – как защищённые поля.

using System;

namespace MultySidedSolution

{

interface ITarget

{

double CalculateDp(int T0, int dT);

void ModifMass(double dm);

String GetData();

}

class GasBalloon

{

public double volume { get; set; }

public double mass { get; set; }

public double molar { get; set; }

const double R = 8.31;

public GasBalloon(double volume, double mass, double molar)

{

this.volume = volume;

this.mass = mass;

this.molar = molar;

}

public double GetPressure(double T)

{

return (mass \* R \* T) / (molar \* volume);

}

public double AmountOfMatter()

{

return mass / molar;

}

public override string ToString()

{

return $"Объём [{volume} м^3] | Масса газа [{mass} кг] | Молярная масса [{molar} кг/моль]";

}

}

class Adapter : ITarget

{

GasBalloon adapted;

public Adapter(GasBalloon gb)

{

adapted = gb;

}

public double CalculateDp(int T0, int dT)

{

return adapted.GetPressure(dT) - adapted.GetPressure(T0);

}

public void ModifMass(double dm)

{

adapted.mass += dm;

}

public String GetData()

{

return adapted.ToString();

}

}

class Program

{

static void Main(string[] args)

{

var gb = new GasBalloon(60, 5, 0.0064);

Console.WriteLine(gb.ToString());

ITarget target = new Adapter(gb);

target.ModifMass(10);

Console.WriteLine(target.GetData());

Console.ReadKey();

}

}

}



[Ссылка на файл](MultySidedSolution/bin/Debug/netcoreapp3.1/MultySidedSolution.exe)

*Ответы на контрольные вопросы:*

1. Какие шаблоны проектирования называют структурными?

Ответ: Эти паттерны отвечают за построение удобных в поддержке иерархий классов.

1. Каково назначение структурного шаблона Адаптер?

Ответ: Добавляет возможность взаимодействия объектов с разными интерфейсами.

1. В чём заключается различие между Адаптером объектов и Адаптером классов?

Ответ: Адаптер объекта применяет для адаптации одного интерфейса к другому композицию объектов адаптируемого класса

Отличие реализации шаблона Адаптер класса будет заключаться только в коде класса Adapter

1. Какая проблема решается с помощью шаблона Фасад?

Ответ: Как обеспечить унифицированный интерфейс с набором разрозненных реализаций или интерфейсов, например, с подсистемой, если нежелательно высокое связывание с этой подсистемой или реализация подсистемы может измениться

1. В чём заключается решение, предлагаемое в шаблоне Фасад?

Ответ: Определить одну точку взаимодействия с подсистемой — фасадный объект, обеспечивающий общий интерфейс с подсистемой, и возложить на него обязанность по взаимодействию с её компонентами

1. К каким последствиям приводит использование шаблона Фасад?

Ответ: клиенты изолируются от классов (компонентов) подсистемы, что уменьшает число объектов, с которыми клиенты взаимодействуют, и упрощает работу с подсистемой; снижается степень связанности между клиентами и подсистемой, что позволяет изменять классы подсистемы, не затрагивая при этом клиентов.