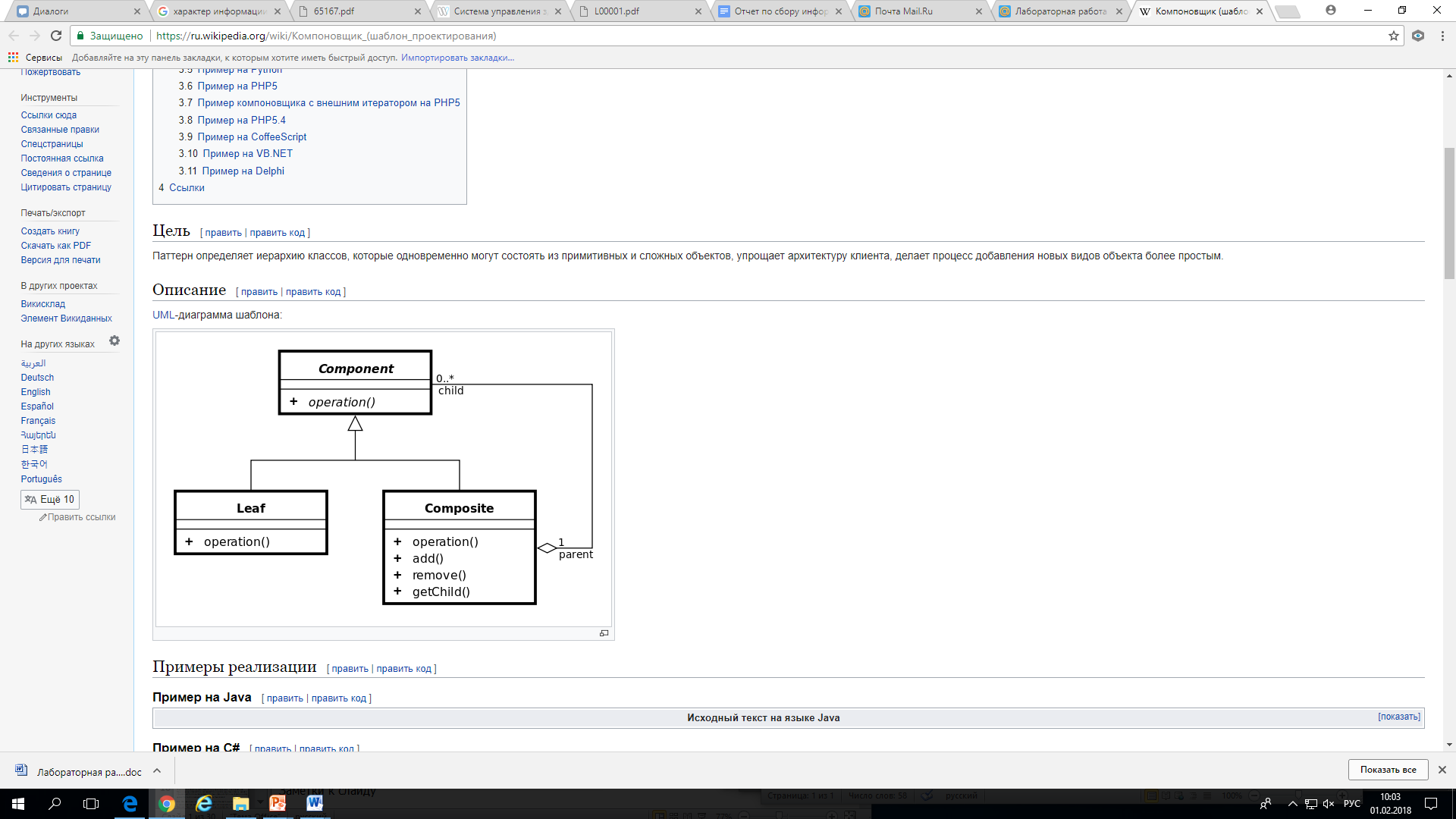
1. Сформулируйте назначение и цель применения паттерна

[**структурный**](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%BD%D1%8B%D0%B5_%D1%88%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD%D1%8B_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)[**шаблон проектирования**](https://ru.wikipedia.org/wiki/%D0%A8%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)**, объединяющий объекты в древовидную структуру для представления иерархии от частного к целому. Компоновщик позволяет клиентам обращаться к отдельным объектам и к группам объектов одинаково.**

**Паттерн определяет иерархию классов, которые одновременно могут состоять из примитивных и сложных объектов, упрощает архитектуру клиента, делает процесс добавления новых видов объекта более простым.**

1. Постройте UML-диаграмму паттерна и дайте пояснения основным элементам



1. Укажите области применения

**Применяется там, где есть иерархия из простых элементов.**

1. Особенности паттерна

**Применяется, когда необходим однообразный интерфейс. (Пример: объект части меню является самим меню)**

1. Приведите примеры реализации

class Program

{

    static void Main(string[] args)

    {

        Component fileSystem = new Directory("Файловая система");

        // определяем новый диск

        Component diskC = new Directory("Диск С");

        // новые файлы

        Component pngFile = new File("12345.png");

        Component docxFile = new File("Document.docx");

        // добавляем файлы на диск С

        diskC.Add(pngFile);

        diskC.Add(docxFile);

        // добавляем диск С в файловую систему

        fileSystem.Add(diskC);

        // выводим все данные

        fileSystem.Print();

        Console.WriteLine();

        // удаляем с диска С файл

        diskC.Remove(pngFile);

        // создаем новую папку

        Component docsFolder = new Directory("Мои Документы");

        // добавляем в нее файлы

        Component txtFile = new File("readme.txt");

        Component csFile = new File("Program.cs");

        docsFolder.Add(txtFile);

        docsFolder.Add(csFile);

        diskC.Add(docsFolder);

        fileSystem.Print();

        Console.Read();

    }

}

abstract class Component

{

    protected string name;

    public Component(string name)

    {

        this.name = name;

    }

    public virtual void Add(Component component){}

    public virtual void Remove(Component component) { }

    public virtual void Print()

    {

        Console.WriteLine(name);

    }

}

class Directory :Component

{

    private List<Component> components = new List<Component>();

    public Directory(string name)

        : base(name)

    {

    }

    public override void Add(Component component)

    {

        components.Add(component);

    }

    public override void Remove(Component component)

    {

        components.Remove(component);

    }

    public override void Print()

    {

        Console.WriteLine("Узел " + name);

        Console.WriteLine("Подузлы:");

        for(int i=0; i<components.Count;i++)

        {

            components[i].Print();

        }

    }

}

class File : Component

{

    public File(string name)

            : base(name)

    {}

}

1. Составьте список источников

<https://metanit.com/sharp/patterns/4.4.php>

https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%BE%D0%BD%D0%BE%D0%B2%D1%89%D0%B8%D0%BA\_(%D1%88%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD\_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)