

Kompozyt

ang. (Composite pattern)

Kompozyt jest to strukturalny wzorzec projektowy (**ang. Structural design pattern**), pozwalający na traktowanie grupy obiektów jakby były obiektem o tym samym typie. Kompozyt jest budowany w struktury drzewiaste.

Przykładem Kompozytu używanego w życiu jest hierarchia w wojsku. Dywizje dzielą się na brygady, potem na drużyny i na poszczególnych członków

Zalety:

- Łatwość rozszerzania o następne instancje
- Klient nie ma dostępu do samej struktury i pozycji obiektów

Wady:

- Im bardziej rozrośnięty staje się interfejs tym ciężiej jest go utrzymać.
- Dochodzi do większych skomplikowań kiedy obiekty w kompozycie bardzo się od siebie różnią

Wniosek:

- Kompozyt pozwala na budowanie struktur obiektów o różnych typach.

Kompozyt można znaleźć np. w budowie gałęzi Gita.

Przykładowy kod:

```
abstract class Component{  
    protected readonly List<Component> _components = new  
    List<Component>();  
    public abstract void ShowDetail();  
    public void Add(Component comp) { _components.Add(comp); }
```

```

        public void Remove(Component comp){
_components.Remove(comp); } }
class Composite : Component{
    private readonly string _name;
    public Composite(string name){ _name = name; }
    public override void ShowDetail(){ } }
class Leaf :Component{
    private readonly string _leaf;
    public Leaf(string leaf) { _leaf = leaf; }
    public override void ShowDetail() { } }
static void Main(string[] args){
    Component Comp = new Composite("");
    Component Comp1 = new Composite("");
    Component leaf1 = new Leaf("");
    Comp.Add(Comp1);
    Comp1.Add(leaf1);
    Comp.ShowDetail();
}
}

```

UML:

