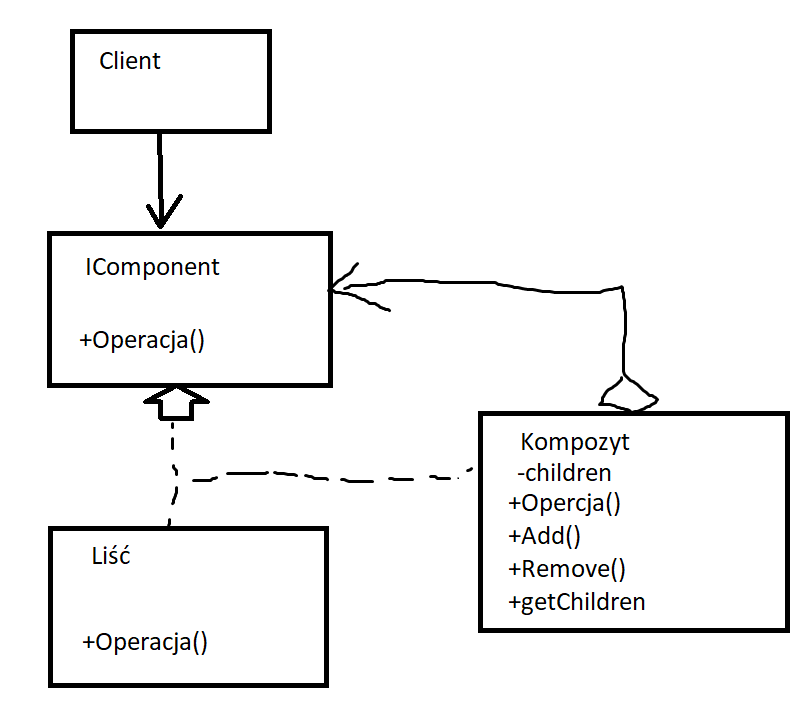
KOMPOZYT

1. Kompozyt – wzorzec projektowy tworzący z obiektów elementy połączone, jak gałęzie drzewa, a następnie traktuje je, jak osobne obiekty. Głównym celem kompozytu jest tworzrzenie hierarchii obiektów, jakby w formie drzewa.

3.



Jest to najbardziej typowy przykład Kompozytu (sam rysowałem :D) i tak można to podsumować:

1. Struktura kompozytu:

Klient na samej górze jest połaczony z interfejsem IComponent, dzięki któremu może działać w każdej gałęzi drzewa, tak jak z tą firmą jest szefem który może działa z każdym z pracowników.

IComponent – interfejs w tym kompozycie, który odgrywa kluczową rolę to on odbiera różne operacje i te proste od liścia i tych których jest więcej od Kompozytu, ale to on jest głównym łącznikiem bloków między sobą i tworzy te gałęzie.

Liść pokazuje podstawowy element drzewa. Robi jakąś operacje i łączy się z IComponent i Kompozyt. Nie ma elementów pod sobą. Robi operacje zlecone przez kompozyt

Kompozyt – on ma elementy pod samym sobą i przy otrzymaniu zadania przekazuje je w odpowiedni sposób do Liścia, po czym przetwarza wyniki pośrednie i zwraca ostateczny wynik klientowi. Jest połączony ze swoimi podelementami oraz z IComponent.

4. Prosty przykład uzycia: Restauracja jest właściciel, który dba o jedzenie, naczynia, ale to kelnerka jest interfejsem, a kompozytem są kucharze.

5. Wniosek: Kompozyt pozwala na dobre rozgałęzienie kodu, który może dużo zrobić. Zalety: struktura drzewiasta – duży potencjał, rozszerzalność, bez zmiany kodu można dodawać i usuwać komponenty, Wady: Używanie go ma sens tylko kiedy chcemy stworzyć coś w formie drzewa.

6. Przykład z życia: porównanie do firmy, masz dyrektora kóry ma pod sobą pracowników, nad którymi ma kontrolę, a sam jest pracownikiem, który jest pod szefem. Tu właśnie działa kompozyt, który jest odpowiedzialny za ustawienie tej hierarchii.

Korzystałem z tych stron:

<https://refactoring.guru/pl/design-patterns/composite>

https://java.edu.pl/inne/designPatterns/8.composite.php