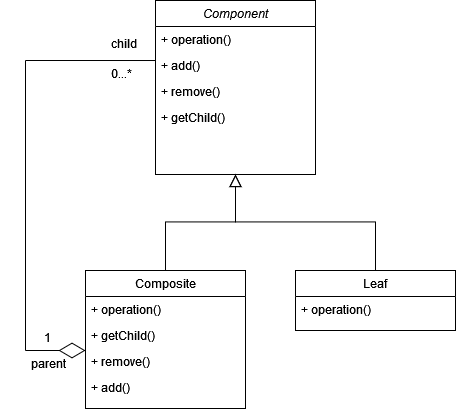
**Wzorce projektowe wykorzystane w projekcie – definicje ogólne**

**Kompozyt**

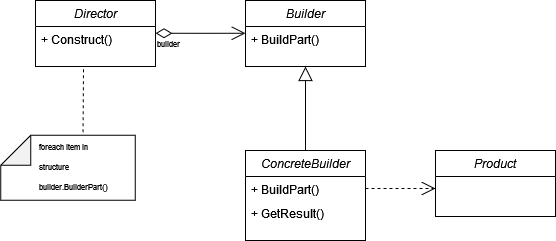
**Kompozyt** umożliwia stworzenie w projekcie struktury drzewa z konkretnych obiektów a następnie daje możliwość potraktowania tej struktury jako pojedynczego obiektu. Największą zaletą tego wzorca jest możliwość sumowania wyników działania wszystkich elementów struktury.

W tym rozwiązaniu nie musimy interesować się konkretną klasą obiektów w strukturze drzewa, ani też tym czy obiekt jest złożony czy prosty. Istnieje tylko jeden interfejs i gdy wywoła się metodę obiekty będą wzajemnie przekazywać ją sobie dalej.



**Budowniczy**

**Budowniczy** pozwala, przy wykorzystaniu tego samego kodu konstrukcyjnego, produkować proste i złożone obiekty w poszczególnych etapach. Aby stworzyć kompletny obiekt wykonuje się ciąg etapów, przy założeniu że nie trzeba ich wszystkich wywoływać. Można przez to zminimalizować ilość kroków do tych wymaganych. Zastosowanie wzorca daje możliwość utworzenia wielu klas budowniczych, implementujących te same kroki ale w zupełnie odmienny sposób, powstają wtedy różniące się od siebie obiekty.



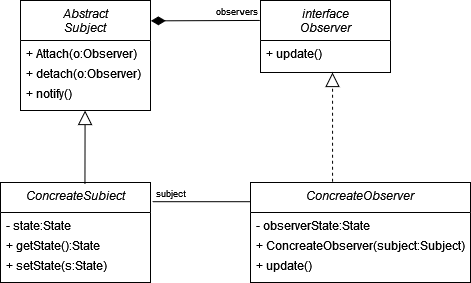
**Singleton**

**Singleton** bardzo poręczne i chętnie wykorzystywane globalne zmienne są nie najlepszym rozwiązaniem. Ich zastosowanie wiąże się z awarią programu, dlatego że istnieje spore ryzyko nadpisania ich zawartości. Aby zapobiec takiej sytuacji stosuje się Singetona, który zapewnia ochronę przed wpływem innego kodu, jednocześnie mamy dalej dostęp do zmienne w przestrzeni globalnej



**Obserwator**

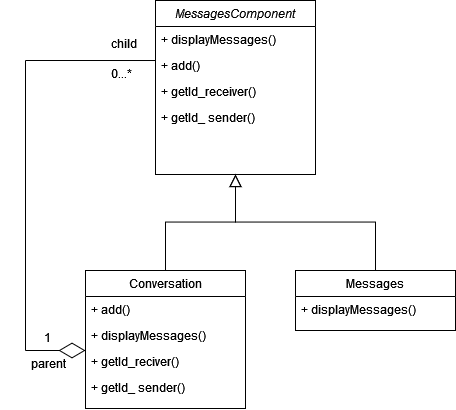
**Obserwator** to wzorzec behawioralny. Zasadniczo składa się z dwóch interfejsów, jeden z nich obserwuje a drugi jest przez niego obserwowany. Obiekt posiadający stan który chcemy obserwować nazywa się najczęściej podmiotem, jest odpowiedzialny jednocześnie za publikację informacji dotyczących zmiany stanu, który był aktualnie obserwowany. Zasada działania opiera się na istnieniu tablicy przechowującej listę odniesień oraz metod, dzięki którym mamy możliwość dodawać lub usuwać elementy z tej tablicy.



**Charakterystyka wzorców wykorzystanych w projekcie**

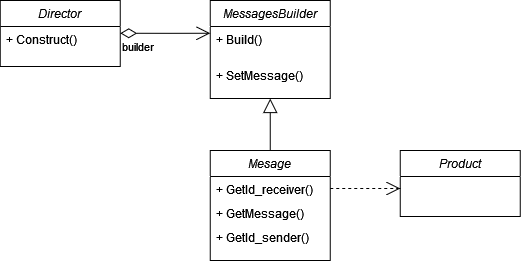
**Kompozyt**

Agregacja wiadomości w jedną zbiorczą konwersacje



**Budowniczy**

Odpowiada za bezpieczną budowę obiektów wiadomości



**Singleton**

Klasa Database odpowiada tutaj za zmiany



**Obserwator**

Monitoruje konwersację w której uczestniczysz, sprawdza w bazie czy ktoś napisał nową wiadomość jeśli tak to ją wyświetla

