Wzorce projektowe: **Adapter, Singleton, Dekorator.**

Wzorzec projektowy **Adapter** (ang. Adapter Design Pattern) jest wzorcem strukturalnym, którego celem jest umożliwienie współpracy dwóch różnych interfejsów. **Adapter** pozwala na dostosowanie interfejsu jednej klasy do oczekiwań innej klasy poprzez przedstawienie tego interfejsu w inny sposób.

Wzorzec **Adapter** jest używany, gdy chcemy użyć istniejącej klasy, ale jej interfejs nie spełnia naszych oczekiwań. Dzięki **Adapterowi** możemy stworzyć nową klasę, która będzie działać zgodnie z oczekiwaniami, a jednocześnie będzie wykorzystywać funkcjonalność istniejącej klasy.

Wzorzec **Singleton** zapewnia, że dana klasa posiada tylko jedną instancję, która jest tworzona tylko raz. Klient może uzyskać dostęp do tej instancji poprzez jedno, globalne miejsce dostępu. Dzięki temu możemy mieć pewność, że dana klasa posiada tylko jedną instancję i jest ona udostępniona w sposób kontrolowany.

Wzorzec **Singleton** jest często używany w aplikacjach, gdy chcemy mieć dostęp do wspólnych danych lub obiektów z różnych miejsc w aplikacji. Może być również używany, gdy chcemy zapewnić, że dana klasa posiada tylko jedną instancję i jest ona udostępniona w sposób kontrolowany.

Wzorzec projektowy **dekorator** jest strukturalnym wzorcem projektowym, który pozwala na dynamiczne dodawanie nowych funkcjonalności do obiektów w obiektowym systemie bez modyfikowania ich kodu źródłowego. Wzorzec ten jest często używany do rozszerzania zachowania obiektu bez konieczności tworzenia nowych podklasy.

Wzorzec **dekoratora** opiera się na koncepcji "opakowywania" obiektu w inny obiekt, który dodaje do niego nowe zachowanie. W ten sposób można "dekorować" obiekt za pomocą różnych dekoratorów, które dostarczają różne funkcjonalności, a także mogą być łączone ze sobą w dowolny sposób.

**Mateusz Cieśla**

Z508 9831