

Pytania do egzaminu inżynierskiego z Informatyki – Rafał Bańka

Programowanie zaawansowane

1. Co to są wzorce projektowe. Jaka jest ich klasyfikacja. Omów jeden wybrany wzorec.

Wzorec projektowy to rozwiązanie problemów często napotykanym przy projektowaniu oprogramowania. Ideą wzorca projektowego jest przedstawienie gotowego „planu”, który można dostosować, aby rozwiązać powtarzający się problem w kodzie. Wzorec nie jest konkretnym fragmentem kodu, ale ogólną koncepcją pozwalającą rozwiązać dany problem.

Wzorce projektowe dzielą się na:

Wzorce kreacyjne wprowadzają elastyczniejsze mechanizmy tworzenia obiektów i pozwalają na ponowne wykorzystanie istniejącego kodu.

Wzorce strukturalne wyjaśniają jak składać obiekty i klasy w większe struktury, zachowując przy tym elastyczność i efektywność struktur.

Wzorce behawioralne, które zajmują się efektywną komunikacją i podziałem obowiązków pomiędzy obiektami.

Przykładowym wzorcem projektowym jest adapter. Jest to strukturalny wzorec projektowy pozwalający na współdziałanie ze sobą obiektów o niekompatybilnych interfejsach. Przykładowym problemem użycia adaptera jest plik w formacie XML pobierający dane, którego dane mają zostać poddane analizie przez bibliotekę bazującą na danych z pliku JSON. Aby zwalczyć ten problem tworzony jest adapter, obiekt konwertujący interfejs jednego z obiektów w taki sposób, że drugi obiekt go zrozumie.

2. Scharakteryzuj kontenery sekwencyjne, skojarzeniowe, haszujące. Podaj kilka przykładów kosztów O wykonywania operacji na danych w tych kontenerach.

Kontener to struktura służąca do przechowywania danych i posiadająca ściśle zdefiniowany interfejs.

Kontenery sekwencyjne są kolekcjami uporządkowanymi, w których każdy element posiada określoną pozycję. Pozycja ta zależy od momentu oraz miejsca wstawienia, jest jednak niezależna od wartości elementu. Przykładowe koszty wykonywania operacji to:

tablica (dostęp – $O(1)$),

lista (wstawianie – $O(1)$, dostęp – $O(n)$, usuwanie – $O(1)$),

wektor (wstawianie – $O(1)$ dla końca wektora $O(n)$ dla początku i środka wektora, dostęp – $O(1)$, usuwanie – $O(n)$).

Kontenery skojarzeniowe (asocjacyjne) są kolekcjami sortowanymi, w których aktualna pozycja elementu zależy od jego wartości (albo klucza w przypadku kontenerów operujących na parach klucz-wartość), zgodnie z określonym kryterium sortowania. Przykładowe koszty wykonywania operacji to:

zbiór (wstawianie – $O(\log n)$, dostęp – $O(\log n)$, usuwanie – $O(\log n)$),

mapa (wstawianie – $O(\log n)$, dostęp – $O(\log n)$, usuwanie – $O(\log n)$)

Kontenery haszujące (nieporządkujące) to kolekcje nieporządkujące i niezachowujące pozycji elementów, bo ich zadaniem głównym jest ustalanie, czy (a nie gdzie) element znajduje się w kolekcji. Elementy nie zachowują więc uporządkowania ani względem kolejności wstawiania, ani względem wartości – jedno i drugie może w czasie życia kontenera ulegać zmianie. Przykładowe koszty wykonywania operacji to:

nieuporządkowany zbiór (wstawianie – $O(1)$, dostęp – $O(1)$, usuwanie – $O(1)$),
nieuporządkowana mapa (wstawianie – $O(1)$, dostęp – $O(1)$, usuwanie – $O(1)$)

3. Na jakiej zasadzie działają systemy kontroli wersji. Czym różni się centralny i rozproszony system kontroli wersji. Scharakteryzuj zalety repozytorium Git.

System kontroli wersji śledzi wszystkie zmiany dokonywane na pliku (lub plikach) i umożliwia przywołanie dowolnej wcześniejszej wersji.

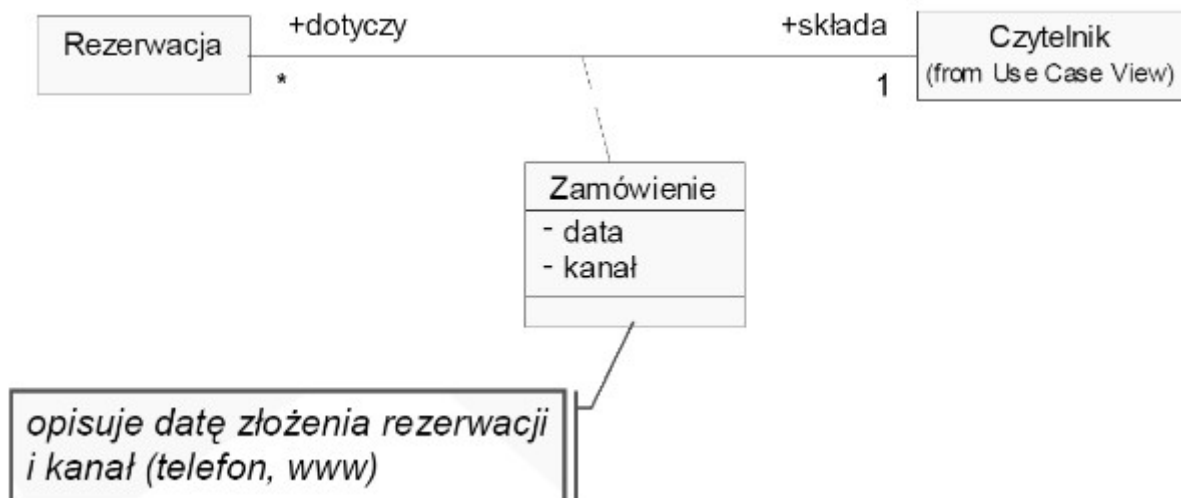
Centralny system kontroli wersji przechowuje repozytorium na serwerze, a w przypadku rozproszonego systemu kontroli wersji repozytorium przechowywane jest na serwerze i komputerze użytkownika.

Zaletami repozytorium Git są:

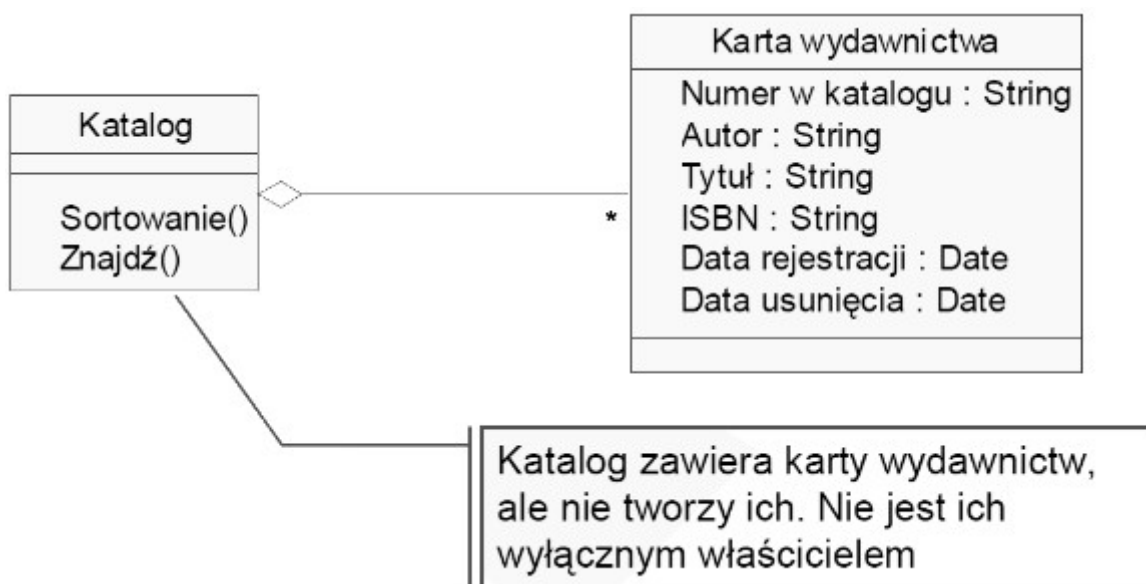
- pozwała na jednoczesną pracę na tym samym kodzie przez kilka osób,
- umożliwia transferowanie i łączenie zmian z różnych gałęzi w jednym projekcie,
- pozwała na pracę offline we własnym repozytorium

4. Omów zależności pomiędzy obiektami (asocjacje, agregacja, dziedziczenie).
Narysuj przykładowe diagramy UML obrazujące te relacje.

Asocjacja reprezentuje czasowe powiązanie pomiędzy obiektami dwóch klas. Obiekty związane asocjacją są od siebie niezależne. Przykład asocjacji:



Agregacja częściowa reprezentuje relację typu całość-część, w której część może należeć do kilku całości, a całość nie zarządza czasem istnienia części. Przykład agregacji częściowej:



Agregacja całkowita jest relacją typu całość-część, w której całość jest wyłącznym właścicielem części, tworzy je i zarządza nimi. Przykład agregacji całkowitej:

