

Paradygmaty Programowania

Laboratorium 2

Dekompozycja problemu programistycznego z wykorzystaniem schematów blokowych i notacji UML

Sprawozdanie proszę wysłać przez platformę Teams lub na adres mailowy podany przez prowadzącego zajęcia (wyłącznie w przypadku problemów z oprogramowaniem Teams). Sprawozdanie powinno zawierać: **imię i nazwisko, numer albumu, datę i formułę**: *Oświadczam, że niniejsza praca stanowiąca podstawę do uznania osiągnięcia efektów uczenia się z przedmiotu Paradygmaty Programowania została wykonana przez mnie samodzielnie.*

Cel ćwiczenia

Celem ćwiczenia jest:

- zaznajomienie studentów z podstawami notacji UML jako formą komunikacji oraz
- praktyka dekompozycji problemów programistycznych i przedstawienia ich w postaci diagramów UML

Przebieg ćwiczenia

Ćwiczenie przebiega według następującego harmonogramu:

- krótki test sprawdzający wstępne przygotowanie studentów
- omówienie narzędzia z którego korzysta się w trakcie zajęć
- przedstawienie problemu programistycznego
- wspomagane przez prowadzących rozwiązywanie zadania

Przygotowanie do zajęć

W celu przygotowania się do zajęć wystarczy przejrzeć materiały do wykładów 5 i 6, ze zwróceniem szczególnej uwagi na:

- pojęcie klasy i obiektu
- relacje jakie mogą zachodzić pomiędzy poszczególnymi klasami i sposób ich reprezentacji na diagramach UML (statyczny klas i dynamiczny sekwencji)

Przykładowe zadanie

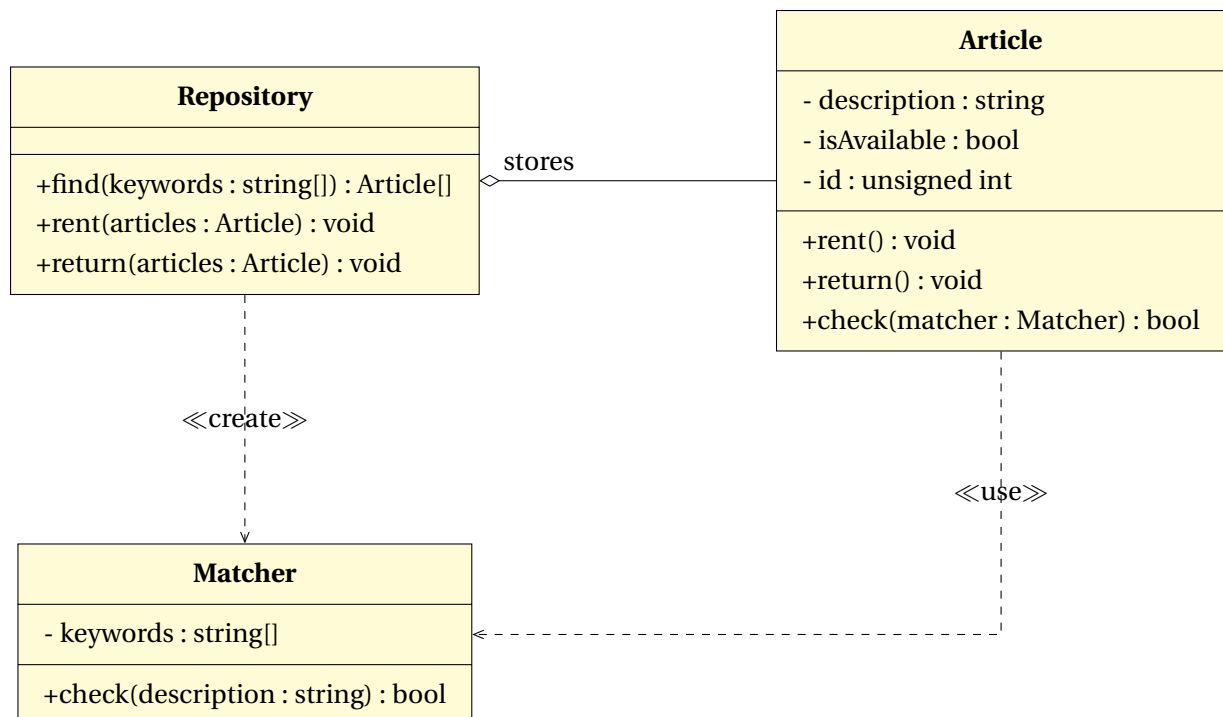
Sklep internetowy

Dany jest elektroniczny magazyn sprzętu sportowego. W rzeczywistym magazynie znajdują się artykuły sportowe. Każdy z artykułów posiada dokładny opis, oraz indywidualny numer identyfikacyjny. W systemie może istnieć wiele produktów tego samego typu, różniących się jedynie numerem.

Proces zamawiania produktów z magazynu przebiega następująco:

1. Użytkownik wpisuje informację czego potrzebuje.
2. System przeszukuje listę elementów dostępnych w magazynie.
3. System zwraca listę elementów spełniających kryteria zapytania.
4. Użytkownik zaznacza te elementy, które chce pobrać z magazynu.
5. Lista potrzebnego sprzętu zwracana jest do systemu.
6. Sprzęt jest pobierany, a poszczególnym elementom zmieniany jest stan - dostępny na niedostępny.

Zaproponuj strukturę klas pozwalających na realizację procesu zamówienia. Przedstaw proces zamawiania w postaci diagramu sekwencji.



Na powyższym diagramie przedstawiono prostą strukturę klas umożliwiającą realizację zadania. Obiekt klasy repozytorium przechowuje artykuły w postaci zbioru elementów. Jego interfejs umożliwia wykonanie podstawowych operacji na elementach: takich jak znalezienie elementów pasujących do wzorca, wypożyczenie oraz zwrot. Warto zwrócić uwagę, że

diagram nie odpowiada na pytanie czy pojedynczy artykuł, może należeć jednocześnie do jednego repozytorium czy do wielu różnych repozytoriów. Na diagramie nie zawarto również informacji o sposobie dodawania nowych elementów i usuwania zużytych. Te aspekty systemu nie są modelowane, gdyż nie dotyczą one procesu realizacji zamówienia.

Obiekt repozytorium odpowiedzialny jest również, za utworzenie odpowiedniego obiektu pozwalającego na sprawdzenie czy dany artykuł pasuje do zapytania.

Odpowiedzialnością obiektu reprezentującego artykuł jest przechowywanie informacji o tym czy dany artykuł jest dostępny czy został już wypożyczony. Indywidualny w ramach systemu numer pozwala jednoznacznie identyfikować przedmiot. Metody *rent* oraz *return* pozwalają na zmianę statusu obiektom reprezentujących repozytorium (repozytoria). Klasa artykułu, wykorzystując do tego obiekt dopasowujący, sprawdza również czy jej opis pasuje do zapytania.

Diagram sekwencji przedstawia proces wypożyczania sprzętu opisany w treści zadania. Oprócz przedstawionej sekwencji poszczególnych operacji diagram zawiera kilka istotnych szczegółów:

- Lista pasujących do zapytania artykułów jest tworzona za każdym razem gdy pojawia się zapytanie.
- Zmiana statusu konkretnego artykułu odbywa się za pośrednictwem repozytorium.
- System nie sprawdza czy dany artykuł nie został już wypożyczony.

Warto zwrócić uwagę, że nie wszystkie szczegóły istotne z punktu widzenia implementacji znajdują swoje odzwierciedlenie na diagramie. Na przykład brakuje informacji o tym czy zwracana lista pasujących artykułów jest listą referencji czy kopii danych obiektów. Na diagramie nie zawarto również informacji o tym w jaki sposób użytkownik decyduje czy dany artykuł jest mu potrzebny czy też nie. Te oraz wiele innych informacji mogą być doprecyzowane na innych diagramach klas lub sekwencji.

Podsumowanie

Język UML traktowany jako narzędzie tworzenia specyfikacji pozwala na efektywne komunikowanie się pomiędzy osobami wchodzącymi w skład zespołu wytwórczego. Osobom nie będącym programistami łatwiej jest zrozumieć proste diagramy i wychwycić błędy modelu i nieścisłości w specyfikacji dla tworzonego oprogramowania. Z kolei programistom łatwiej jest przełożyć na język, w którym jest implementowana aplikacja statyczne i dynamiczne diagramy UML niż opis tekstowy zawarty w specyfikacji.

Ostatnia aktualizacja: 6 kwietnia 2021

