

Instrukcje i wskazówki do przygotowania diagramu klas

(IO 2024/2025)

Wyznaczenie zakresu

- Otworzyć model przypadków użycia (PU) przygotowany w poprzednim etapie (przy czym, jeśli były uwagi osoby prowadzącej do PU, to lepiej je uwzględnić w pierwszej kolejności, przed rozpoczęciem modelowania klas).
- Spróbować zidentyfikować klasy i ich właściwości oraz związki wynikające z PU i przełożyć to na elementy diagramu klas.

Uwaga: Standardowo przyjmuje się, że diagram klas powinien pokrywać wszystko, co jest potrzebne do realizacji PU. Gdyby jednak zespół miał poczucie, że wynikowy diagram klas będzie olbrzymi (albo w drugą stronę – bardzo mały), należy taką sprawę skonsultować z osobą prowadzącą laboratorium.

Modelowanie diagramu klas

Klasy

- Na diagramie należy uwzględnić klasy reprezentujące byty z dziedziny problemowej, nie zaś dotyczące wewnętrznej budowy systemu (baza danych, ekrany czy okna dialogowe, stan sesji połączenia internetowego etc.). Powinny to być jednak tylko te byty, które będą potem reprezentowane w systemie.
- Najlepszym sposobem identyfikacji klas jest przegląd PU i „wychwycenie” bytów informacyjnych, które są prezentowane użytkownikowi lub przez niego wprowadzane czy też modyfikowane. Np. opis interakcji typu: „Klient przegląda produkty dostępne w sklepie” sugeruje uwzględnienie klasy Produkt; opis „Kierownik zmiany przegląda listę aktualnych zadań i przypisuje je pracownikom” implikuje, że zapewne potrzebne będą klasy Zadanie i Pracownik.
- Jak widać z poprzedniego punktu, z przykładowych opisów nie wynikała konieczność dodania na diagramie klas Klienta czy Kierownika Zmiany. Możliwe jednak, że takie klasy wynikałyby z innych PU (np. „Klient zakłada konto w systemie”; „Do każdej zmiany przypisywany jest kierownik”) lub wymagania, żeby w systemie była śladowość tego, kto co zrobił.
- Poza PU można się dodatkowo wspomagać opisem dziedziny z Wizji Systemu, ale należy zachować ostrożność – nie każdy rzeczownik, który pojawi się w opisie, będzie bytem informacyjnym reprezentowanym w systemie.

Atrybuty i operacje

- Jeżeli PU opracowywane były zgodnie ze wskazówkami (tzn. odpowiednio szczegółowo), to prawdopodobnie ich lektura, poza samymi klasami, pozwoli zidentyfikować większość atrybutów tych klas np. „Klient wybiera funkcję przeglądania produktów dostępnych w sklepie. System prezentuje mu listę aktualnie sprzedawanych produktów, dla każdego podając jego nazwę, cenę, dostępność (dostępny/na wyczerpaniu/chwilowo niedostępny), nazwę producenta, ...” – większość potencjalnych atrybutów jest bezpośrednio podana.
- Warto przyjąć, że atrybuty powinny być typów prostych czy też nie bardziej złożonych niż np. data lub waluta. Nie wynika to z właściwości podejścia obiektowego – pozwala ono na stosowanie

atrybutów będących np. listami, tablicami czy innymi obiektami. Chodzi jednak o uniknięcie ryzyka, że zespół pominie istotne byty, które powinny być zamodelowane jako klasy. Na przykład w systemie dla sklepu internetowego ktoś mógłby zamodelować klasę Klient i umieścić w niej atrybut „zamówienia” – i byłoby to traktowane jako błąd, gdyż Zamówienie powinno być osobną klasą opisaną przez własne atrybuty, a w dodatku powiązaną z innymi klasami np. Produkt, Płatność, Pozycja Zamówienia itp., które w konsekwencji pewnie również zostałyby pominięte.

- Operacje będą oceniane w mniejszym stopniu, ponieważ trudno jest bez uwzględnienia modeli dynamicznych (stanowiących temat następnego etapu) zidentyfikować wszystkie operacje. Zespół powinien jednak wykonać pewien wysiłek, żeby uwzględnić chociaż najważniejsze operacje odpowiadające funkcjom opisywanym w PU. Na pewno diagram klas nie powinien wyglądać jak diagram ERD czyli z samymi atrybutami.
- Ważną kwestią jest to, że operacje powinny być definiowane w klasie, na której działają. Częstym błędem jest przypisywanie operacji do klas reprezentujących ludzi – użytkowników systemu na zasadzie „listy dostępnych funkcji”. Dla wcześniejszych przykładów: „Klient przegląda produkty dostępne w sklepie” sugeruje operację typu pobierzOpis() dla klasy Produkt, nie przeglądaj() w klasie Klient! „Kierownik zmiany przegląda listę aktualnych zadań i przypisuje je pracownikom” pewnie będzie się przekładać na operacje pobierzDane() i przypiszWykonawcę() definiowane w klasie Zadanie.

Związki

- Związki między klasami będą zapewne w większości asocjacjami, dodatkowo z możliwością użycia generalizacji i agregacji (słabych i silnych – kompozycji). Związki zależności (dependency) na analitycznych diagramach klas nie występują.
- Generalizację stosuje się, gdy można wyróżnić ogólniejsze i bardziej specjalizowane byty oraz jest sens ich rozróżnienia (właściwości, które można przypisać do klasy ogólniejszej - współdzielone atrybuty, operacje lub związki z innymi klasami).
- Agregację i kompozycję stosuje się, gdy opisując byty z dziedziny problemowej można powiedzieć, że „coś jest częścią czegoś” lub „coś się składa z czegoś” np. pozycja faktury jest częścią faktury, czy też klasa szkolna składa się z uczniów. Nie należy myśleć w kategoriach wewnętrznego zapisu danych w systemie np. „adres klienta będzie zapisywany w jego rekordzie, więc dajmy kompozycję, że Klient zawiera Adres” – ponieważ nikt raczej nie powie, że adres zamieszkania jest integralną częścią jego/jej osoby...
- Agregacje i kompozycje zwykle nie mają nazw (bo samo zastosowanie związku tego typu implikuje nazwę „składa się z” lub podobną), powinny jednak mieć wyspecyfikowane licznosci!
- Kompozycja to silniejsza agregacja, opatrzona dodatkowymi warunkami: część musi zawsze należeć do 1 całości (nie do zera, nie do wielu), część nie zmienia swojej przynależności do całości, w przypadku usunięcia całości część także jest usuwana. Czyli przykładami mogą być Budynek-Mieszkanie czy Faktura-Pozycja Faktury, ale na pewno nie: Klasa Szkolna-Uczeń.
- Asocjacje powinny być opisane nazwami (precyzującymi, o co chodzi w danym związku) oraz licznosciami. Alternatywnie, zamiast nazwy asocjacji można stosować nazwy ról każdej z połączonych klas.
- Kluczowe jest uchwycenie za pomocą asocjacji związków istotnych z punktu widzenia systemu i jego użytkowników np. to jest zamówienia złożone przez tego klienta, w ramach tej zmiany w fabryce mają pracować następujący pracownicy, ten nauczyciel jest wychowawcą tej klasy. Jest to dużo ważniejsze niż często spotykane na studenckich diagramach klas asocjacje typu: przegląda, ogląda, ma dostęp, zarządza etc.

Opisy

- W przeciwieństwie do PU, których główną wartość stanowiły opisy, diagramy klas zawierają zwykle tylko to, co jest widoczne na samym diagramie.
- Mogą być jednak sytuacje, gdy z samej nazwy klasy lub atrybutu trudno wywnioskować, co ma reprezentować. Na przykład: klasa Przedmiot w systemie dla szkoły może oznaczać „Fizykę”, „Fizykę dla klas drugich licealnych”, „Fizykę dla klasy II A w roku szkolnym 2024/25” itd. i w takiej sytuacji może być warto dodać zdanie czy dwa opisu.
- Jeśli chodzi o atrybuty, szczególnie warto wyjaśniać znaczenie atrybutów takich jak: „stan”, „status”, „rodzaj”, „kategoria” itp. Tak więc np. dla klasy Pracownik nie ma sensu dawać przy atrybucie „imię” opisu „to jest imię pracownika”, ale już np. atrybut „status” może odnosić się do zajmowanego stanowiska, aktualności zatrudnienia, oceny pracy, przepracowanych nadgodzin i wielu innych rzeczy. Najlepiej jest wówczas wymienić w opisie atrybutu typu „status” wszystkie możliwe wartości, jakie może on przyjmować.