

---

## Czym są wzorce projektowe?

3

## Przypisywanie odpowiedzialności obiektom – podstawowe zadanie fazy projektowej

---

- Projektowanie obiektowe jest kolejnym zadaniem, które jest realizowane po fazie zebrania wymagań oraz utworzeniu modelu wiedzy dziedzinowej, polegające na dodaniu metod do klas oraz zdefiniowaniu komunikatów przesyłanych pomiędzy obiektami w celu zrealizowania zebranych wymagań.
- Zadanie to jest kluczowe w procesie tworzenia systemów obiektowych, jednakże przypisanie metod do obiektów oraz określenie tego jak ma być realizowana interakcja pomiędzy obiektami jest niezwykle trudne.
- Przypisywanie odpowiedzialności obiektom może zostać opisane przy pomocy zestawu reguł tzw. GRASP patterns.

4

## Odpowiedzialność obiektów, a metody (1)

---

- Odpowiedzialność to cel, zobowiązanie lub wymagane możliwości obiektu lub klasy. Odpowiedzialność jest określana jako zestaw usług realizowanych przez obiekt lub klasę.
- Odpowiedzialność obiektu za działanie (doing):
  - Realizowanie czegoś samodzielnie (tworzenie innych obiektów, wykonywanie obliczeń),
  - Inicjowanie akcji realizowanych przez inne obiekty,
  - Kontrolowanie i koordynowanie akcji innych obiektów.
- Odpowiedzialne obiektu za wiedzę (knowing)
  - Posiadanie wiedzy o danych prywatnych,
  - Posiadanie wiedzy o powiązanych obiektach,
  - Posiadanie wiedzy o faktach, które mogą zostać wywnioskowane lub obliczone.

5

## Odpowiedzialność obiektów, a metody (2)

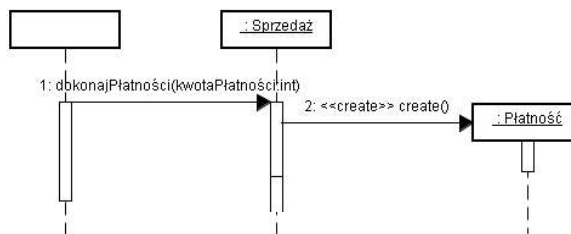
---

- Przykład:
  - Odpowiedzialność obiektu za działanie: „obiekt *Sprzedaz* jest odpowiedzialny za tworzenie obiektu *ElementPozycjiSprzedazy*”,
  - Odpowiedzialność obiektu za wiedzę: „obiekt *Sprzedaz* jest odpowiedzialny za posiadanie wiedzy o sumie sprzedaży”.
- Odpowiedzialność to nie to samo co metody, ale metody są implementowane po to aby realizować odpowiedzialność obiektów.
- Odpowiedzialności obiektów są implementowane przy pomocy metod, które działają samodzielnie, bądź metod które współpracują z innymi obiektami!
- Dana odpowiedzialność może być realizowane przy pomocy bardzo różnej liczby obiektów (od kilku do kilku tysięcy)

6

## Odpowiedzialność, a diagramy interakcji

- Tworzenie diagramów interakcji wspiera etap identyfikowania odpowiedzialności obiektów.
- Diagramy interakcji ilustrują wybory analityka podczas przypisywania odpowiedzialności poszczególnym obiektom.



- Przykład ilustruje przypisanie obiektowi *Sprzedaż* odpowiedzialności polegającej na tworzeniu obiektu *Platność*. Odpowiedzialność jest realizowana podczas wywołania metody *dokonajPlatności* i jest realizowane przez tą metodę.

7

## Wzorce projektowe (patterns)

- Doświadczenia projektantów systemów obiektowych zapisywane są w skodyfikowany sposób przy pomocy tzw. wzorców projektowych (patterns).
- W technologii obiektowej wzorce projektowe opisują pewien znany problem oraz jego rozwiązanie w taki sposób aby można było tą wiedzę wykorzystać w nowej sytuacji:
  - Nazwa wzorca projektowego**
  - Rozwiązanie**
  - Problem**
- Wzorce projektowe opisują wiedzę sprawdzoną i powszechnie akceptowaną. Dla każdego znającego reguły projektowania obiektowego wiedza przekazywana przy pomocy wzorców projektowych wydawała się będzie znajoma i podstawowa.
- Każdy wzorec projektowy powinien posiadać sugestywną nazwę:
  - Powinna ona oddawać główne cechy wzorca, który opisuje,
  - Powinna ułatwiać komunikację podczas projektowania systemu.

8

## Wzorce projektowe GRASP

---

- Wzorce projektowe GRASP (General Responsibility Assignment Software Patterns) pozwalają na przypisywanie odpowiedzialności obiektom.
- Podstawowe wzorce projektowe typu GRASP (Larman):
  - Expert,
  - Creator,
  - High Cohesion,
  - Low Coupling,
  - Controller.

---

**Wzorce projektowe określające zasady  
przypisywania odpowiedzialności obiektom**

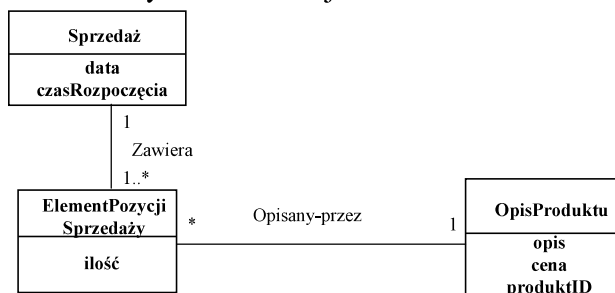
## Expert: założenia

- **Rozwiązanie:** przypisuj odpowiedzialność do obiektów, które pełnią rolę ekspertów posiadających określoną informację (klasy posiadające informację niezbędną do wypełnienia pewnej odpowiedzialności).
- **Problem:** Jak jest podstawowa zasada przypisywania odpowiedzialności obiektom?
- Podstawowy wzorzec projektowy pozwalający na przypisywanie obiektom odpowiedzialności w najbardziej oczywistych sytuacjach.
- Wzorzec zaleca, aby zacząć przypisywać obiektowi odpowiedzialność po wcześniejszym czytelnym ustaleniu czego dana odpowiedzialność dotyczy.
- Zasada postępowania przy poszukiwaniu klas, którym można przypisać poszukiwaną odpowiedzialność:
  - W pierwszej kolejności poszukujemy kandydatów w modelu projektowym,
  - W drugiej kolejności poszukujemy kandydatów w modelu wiedzy dziedzicznej i wprowadzamy te klasy do modelu projektowego.

11

## Expert: przykład (1)

- W programie terminal wielozadaniowy POS poszukujemy klasy, która powinna znać „wartość sprzedaży razem”.
- Odpowiedzialność dotyczy określenia „kto powinien być odpowiedzialny za posiadanie informacji o wartości sprzedaży razem”.
- Przyjmujemy, że rozpoczynamy proces projektowania i nie dostępne są jeszcze, żadne klasy projektowe. Dostępny jest za to następujący model wiedzy dziedzicznej:



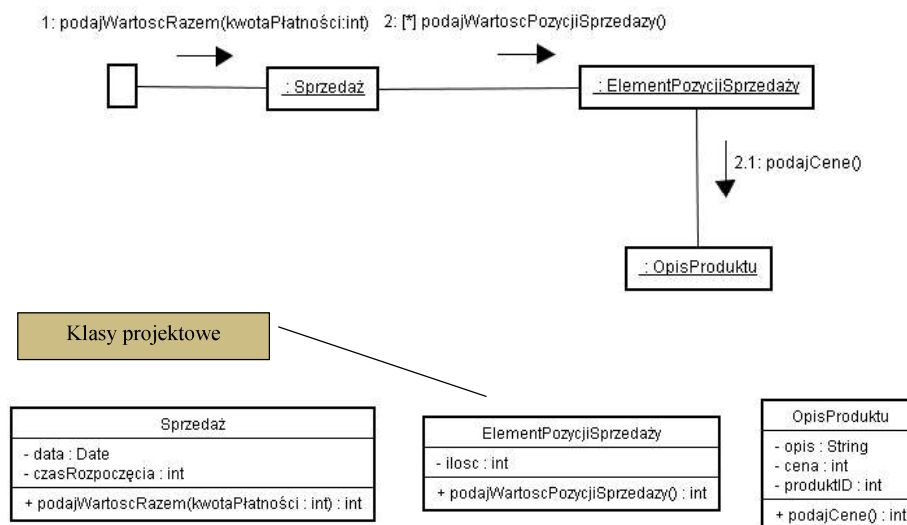
12

## Expert: przykład (2)

- Przy założeniu, że rozpoczynamy proces projektowy i nie zawarte są w nim żadne klasy projektowe:
  - Kandydatem, który powinien posiadać odpowiedzialność jest *Sprzedaż*, a wprowadzoną metodą *podajWartoscRazem*,
  - W celu wyznaczenia wartości sprzedaży razem konieczna jest znajomość wartości poszczególnych sprzedanych produktów – kandydatem na przypisanie tej odpowiedzialności jest klasa *ElementPozycjiSprzedaży*, a wprowadzoną metodą *podajWartoscPozycjiSprzedaży*,
  - W celu obliczenia wartości poszczególnych sprzedanych produktów konieczna jest znajomość ceny produktów – kandydatem na przypisanie tej odpowiedzialności jest klasa *Opis produktu*, a wprowadzoną metodą *podajCene*.

13

## Expert: przykład (3)



14

## Expert: przykład (4)

---

- W procesie poszukiwania odpowiedzialności „wartość sprzedaży razem” uzyskano trzy odpowiedzialności, przypisane do trzech różnych klas:

Klasa projektowa	Odpowiedzialność
Sprzedaż	Zna wartość sprzedaży razem
ElementPozycjiSprzedaży	Zna wartość pozycji sprzedaży
OpisProduktu	Zna cenę produktu

- Odpowiedzialności zostały uzyskane w procesie budowy diagramów interakcji.
- Wzorzec projektowy Expert pozwolił na przypisanie odpowiedzialności do tego obiektu, który posiadana daną informację, niezbędną do spełnienia danej odpowiedzialności.

15

## Expert: dyskusja

---

- Expert to elementarny i powszechnie używany wzorzec projektowy odzwierciedlający powszechną intuicję, że obiekty realizują czynności związane z informacją, którą posiadają.
- Często zdarza się, że odpowiedzialność dotyczy informacji, która jest rozproszona pomiędzy różnymi obiektami – w konsekwencji wielu ekspertów posiadających wiedzę cząstkową będzie współpracowało (przesyłając sobie komunikaty) w celu dostarczenia pełnej informacji (w przykładzie współpracują trzy klasy – trzech ekspertów posiadających informację cząstkową).
- Wzorzec ten prowadzi do projektów, w których obiekty (softwarowe) realizują pewne czynności, podczas gdy ich protoplaści w świecie rzeczywistym tego nie czynią (reguła „Do It Myself”). Np. w świecie rzeczywistym obiekt *Sprzedaż* nie jest w stanie samodzielnie obliczyć wartości razem. Ktoś inny (sprzedawca, kasa) oblicza wartość sprzedaży razem! W modelu obiektowym obiekty softwarowe są „ożywione” i dlatego realizują czynności dla informacji, którą posiadają.

16

## Expert: korzyści

---

- Korzyści:
  - Enkapsulacja informacji jest zachowana dzięki temu, że obiekty wykorzystują swoją własną informację w celu realizowania zadań,
  - Zachowanie jest rozproszone pomiędzy różne klasy, pozwala to tworzyć przejrzyste modele.
- Inne nazwy na wzorzec projektowy Expert:
  - „Information Expert”
  - „Do It Myself”
  - „Place responsibilities with data”
  - „That which knows, does”
  - „Put services with the Attributes They work On”

17

## Creator: założenia

---

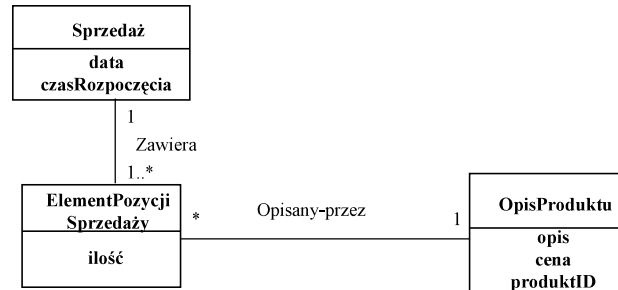
- **Rozwiązanie:** Klasa B powinna posiadać odpowiedzialność tworzenia instancji klasy A jeżeli zachodzi jedno z poniższych:
  - B agreguje obiekty klasy A,
  - B zawiera obiekty klasy A,
  - B w bardzo ścisły sposób używa obiektów A,
  - B posiada dane inicjalizujące, które zostaną przekazane do obiektu A gdy ten jest tworzony.
- **Problem:** Kto powinien być odpowiedzialny za tworzenie nowych instancji obiektów danej klasy?

18



## Creator: przykład (1)

- Kto w programie wielozadaniowy POS powinien tworzyć instancje klasy *ElementPozycjiSprzedaży*?
- Analizujemy odpowiedni fragment modelu wiedzy dziedzinowej:

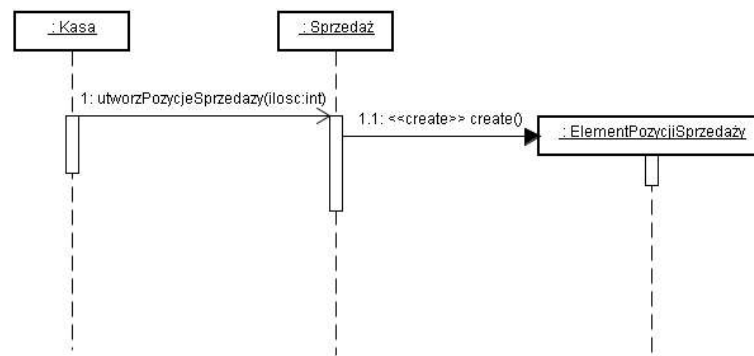


- Ponieważ *Sprzedaż* agreguje wiele obiektów *ElementPozycjiSprzedaży*, wzorec Creator sugeruje tą właśnie klasę jako kandydata przy przypisywaniu odpowiedzialności tworzenia instancji klasy *ElementPozycjiSprzedaży*.

19

## Creator: przykład (2)

- Przypisanie odpowiedzialności wymaga przypisania klasie *Sprzedaż* metody *utwórzPozycjęSprzedaży*.
- Odpowiedzialność ta została uzyskana w procesie budowy diagramu interakcji:



20

## Low Coupling: założenia (1)

---

- **Rozwiązanie:** Odpowiedzialność powinna być przypisywana w taki sposób, aby powiązanie obiektów (coupling) było niskie.
- **Problem:** W jaki sposób utrzymać małą zależność obiektów, niski współczynnik negatywnych skutków wprowadzania zmian i zwiększać powtórne użycie?
- Powiązanie (coupling) wskazuje jak silnie jeden element jest związany z innym elementem, posiada wiedzę o innym elemencie lub zależy od innego elementu.

21

## Low Coupling: założenia (2)

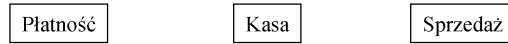
---

- Element (klasa, komponent, system) posiadający niski współczynnik powiązania (low coupling) nie zależy od zbyt wielu innych elementów. Co oznacza „zbyt wiele” zależy od rozpatrywanego kontekstu.
- Klasa z wysokim współczynnikiem powiązania zależy od zbyt wielu innych klas. Podejście to jest niekorzystne ze względu na:
  - Zmiany w powiązanych klasach wymuszają zmiany w rozpatrywanej klasie,
  - Trudności w zrozumieniu roli rozpatrywanej klasy,
  - Trudności z powtórny użyciu – powtórne użycie zakłada każdorazowo obecność klas zależnych.

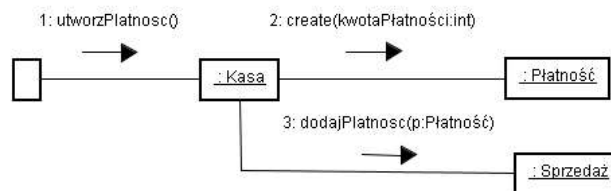
22

## Low Coupling: przykład (1)

- W programie wielozadaniowy POS musimy rozstrzygnąć, kto jest odpowiedzialny za tworzenie instancji klasy *Płatność*?
- Analizujemy odpowiedni fragment z modelu wiedzy dziedzinowej:



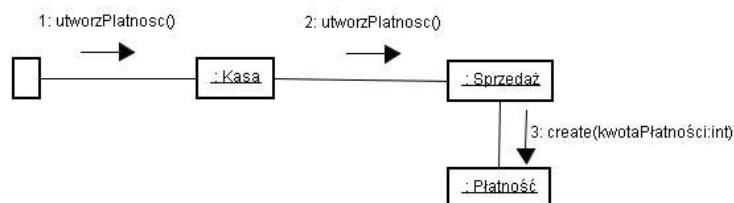
- W związku z faktem, że w świecie rzeczywistym kasa zapisuje informacje o płatnościach, wzorzec projektowy Creator sugeruje klasę *Kasa* jako kandydata do przypisania odpowiedzialności tworzenia instancji klasy *Płatność*. Następnie utworzona instancja może zostać przesłana (jako parametr) przy pomocy metody *dodajPłatność* do obiektu *Sprzedaż*.



23

## Low Coupling: przykład (2)

- Przypisanie odpowiedzialności w ten sposób wiąże klasę *Kasa* z klasą *Płatność*!?
- Rozwiązaniem alternatywnym jest tworzenie klasy *Płatność* bezpośrednio przez klasę *Sprzedaż*:



- Z punktu widzenia wzorca projektowego Low Coupling powyższe rozwiązanie jest poprawne ze względu na fakt utrzymania niskiego współczynnika powiązania.
- Przykład pokazuje wykluczanie się dwóch wzorców: Creator i Low Coupling!

24

## Low Coupling: powiązania, a języki programowania

---

- W języka C++, Java, C# spotykanymi formami powiązań w kierunku od Typu A do Typu B są:
  - A posiada atrybut, który wskazuje na instancję B,
  - Instancja A wywołuje usługi instancji B,
  - X posiada metody, które odwołują się do instancji B. Jest to parametr lub zmienna lokalna typu B, bądź obiekt zwracany przez daną metodę jest typu B,
  - X jest podklasą Y,
  - Y jest interfejsem, a A implementuje ten interfejs.

25

## Low Coupling: dyskusja (1)

---

- Wzorzec projektowy Low Coupling powinien być brany pod uwagę każdorazowo podczas podejmowaniu decyzji dotyczących odpowiedzialności danej klasy! Jest więc wzorcem oceniającym wyniki uzyskane po zastosowaniu innych wzorców!
- Wykorzystywanie wzorca pozwala na tworzenie klas bardziej niezależnych, zredukowany jest współczynnik negatywnych skutków wprowadzania zmian.
- Wzorzec powinien być wykorzystywany razem z innymi wzorcami jako wzorzec silnie wpływający na podejmowane decyzje dotyczące przypisywania odpowiedzialności.
- Nie ma jednoznacznych zaleceń dotyczących tego, kiedy współczynnik powiązania jest zbyt wysoki. Wszystko zależy od kontekstu, decyzje są każdorazowo podejmowane arbitralnie przez projektanta.

26

## Low Coupling: dyskusja (2)

---

- Ekstremalnym przypadkiem użycia wzorca Low Coupling jest sytuacja w której klasy nie są w ogóle powiązane. Nie jest to rozwiązanie korzystne ponieważ:
  - Łamana jest zasada, że obiekty komunikują się poprzez przesyłanie komunikatów,
  - Tworzone są pojedyncze obiekty, które są odpowiedzialne za wykonywanie całej pracy, pozostałe obiekty spełniają jedynie rolę zwykłych repozytoriów danych.
- Wzorec ten jest jednym z fundamentalnych wzorców projektowych. Jego znaczenie jest podnoszone od początku istnienia analizy i projektowania obiektowego.