Ćwiczenie 4 i 5

Tworzenie modelu obiektowego z wykorzystaniem notacji *UML* w środowisku *Object-Oriented Model* pakietu *Power Designer 9.5*

W ćwiczeniu będziemy tworzyć bazujący na metodyce UML model obiektowy, opisujący wybrane aspekty funkcjonowania księgarni internetowej:

"Księgarnia umożliwia zakup książek poprzez stronę WWW. Klient łączy się ze stroną księgarni i wprowadza informacje dotyczące szukanej przez niego pozycji książkowej. System następnie wyszukuje określonej (lub określonych) pozycji i dostarcza klientowi informację na temat wyniku przeszukiwania. Jeśli klient jest zadowolony z odpowiedzi, wypełnia formularz, który następnie jest przesyłany do przedstawiciela handlowego księgarni. Obsługuje on jego żądanie i przesyła mu e-mailem potwierdzenie zamówienia".

W trakcie ćwiczeń utworzymy proste przykłady następujących diagramów:

- przypadków użycia (use case diagram),
- kolaboracji (collaboration diagram),
- sekwencji (sequence diagram),
- stanów (statechart diagram),
- aktywności (activity diagram),
- klas i obiektów (class diagram, object diagram),
- komponentów (component diagram),
- wdrożeniowy (deployment diagram).

1. Wstępne zapoznanie się ze środowiskiem *OOM* pakietu *Power Designer*. Definiowanie opcji i preferencji wyświetlania modelu.

- Uruchomić program Power Designer (menu Start/Programy/Sybase/PowerDesigner9)
- Z górnego menu wybrać polecenie *File/New* a następnie *Object Oriented Model* oraz *OK*. Pojawi się okno dialogowe, w którym możemy wybrać z listy rozwijalnej nazwę obiektowego języka programowania oraz typ diagramu, który będzie tworzony jako pierwszy.
- Wybrać Java z pierwszej listy oraz Use Case Diagram z drugiej listy jako pierwszy będziemy tworzyć diagram przypadków użycia. Przełącznik radiowy powinien być ustawiony na opcję Share.

Object language:	Java	• □
	 Share: Use the shared Object Language definition 	
	C Copy: Create a copy of the Object Language definition in model	
<u>F</u> irst diagram:	Use Case Diagram	•

- Kliknąć *OK* pojawi się puste okno diagramu przypadków użycia, paleta narzędzi (*Palette*), okno przeglądarki obiektów (*Browser*) oraz okno wyjściowe (*Output*).
- Wybrać z górnego menu polecenie *Model/Model Properties*. W polu edycyjnym *Name* wpisać *Tutorial OOM* to będzie nazwa naszego modelu. Kliknąć *OK*.
- Rozwinąć węzeł modelu w oknie przeglądarki (*Browser*). Prawym przyciskiem myszy kliknąć gałąź diagramu przypadków użycia (*UseCaseDiagram_1*) i z menu kontekstowego wybrać *Properties* pojawi się arkusz właściwości diagramu.
- W polu *Name* wpisać *Księgarnia internetowa*, kliknąć *OK*.
- Z górnego menu wybrać polecenie *File/Save As* i zapisać model pod swoją nazwą.
- Z górnego menu wybrać polecenie *Tools/Model Options*, a następnie zaznaczyć następujące opcje: *Name/code case sensitive* oraz *Preview editable* w grupie *All Objects*, następnie *Attribute: int, Operation: int, Parameter: int* w grupie *Default data types* oraz *Data type* w grupie *Domain/Attribute*. Wszystkie pozostałe opcje nie powinny być zaznaczone.
- Kliknąć przycisk *Set As Default*, a następnie *OK*.

2. Tworzenie diagramu przypadków użycia (use case diagram)

Diagram przypadków użycia jest modelem wymagań odnośnie funkcjonalności systemu, widzianej przez użytkowników zewnętrznych, zwanych aktorami. Przypadki użycia opisują, co tacy użytkownicy mogą zrobić z systemem, w jaki sposób go **używać** (stąd nazwa - *przypadek użycia*).

Aktor – abstrakcyjny użytkownik systemu, reprezentujący grupę rzeczywistych użytkowników lub partnerów systemu o podobnych funkcjach i sposobie komunikacji z systemem. Przykłady aktorów: *użytkownik, klient, magazynier*.

Przypadek użycia – ciąg interakcji pomiędzy aktorem a systemem oraz transakcji (niepodzielnych operacji) w systemie, dostarczający aktorowi rezultatu o mierzalnej wartości. Przykłady przypadków użycia: *kupowanie biletów, ustawianie opcji wyświetlania*.

2.1. Definiowanie opcji wyświetlania modelu przypadków użycia

- Z górnego menu wybrać polecenie *Tools/Display Preferences* pojawi się okno dialogowe preferencji wyświetlania modelu.
- Wybrać podgałąź *Association* w gałęzi *Object View* i usunąć zaznaczenie opcji *Show name*, *Show stereotype* oraz *Show direction*. Opcja *Show direction* nie powinna być zaznaczona, zgodnie z notacją *UML*.
- Kliknąć przycisk *Set As Default*, a następnie *OK*.

2.2. Tworzenie obiektów w diagramie przypadków użycia

• Kliknąć symbol *Use Case* (*przypadek użycia*) - niebieski owal znajdujący się na palecie narzędzi.

- Kliknąć na diagramie w miejscu kliknięcia pojawi się symbol przypadku użycia. W podobny sposób utworzyć trzy dodatkowe symbole przypadków użycia. Kliknąć prawym przyciskiem myszy, aby zwolnić symbol przypadku użycia.
- Kliknąć w symbol *Case_1*, a następnie w środek tego symbolu i wpisać nazwę przypadku użycia *Poszukiwanie produktu*. Następnie kliknąć w dowolnym miejscu diagramu w celu zatwierdzenia nowej nazwy.
- W podobny sposób ustalić nazwy pozostałych przypadków użycia *Przeszukiwanie katalogu, Obsługa sprzedaży, Kupowanie produktu.* Diagram powinien wyglądać jak poniższym rysunku.



- Zapisać model polecenie *File/Save*.
- Kliknąć symbol *Actor* na palecie narzędzi. Następnie kliknąć na diagramie, po lewej stronie symbolu *Poszukiwanie produktu* pojawi się symbol aktora.
- Utworzyć dwa dodatkowe symbole aktorów po prawej stronie symboli przypadków użycia.
- Kliknąć prawym przyciskiem myszy, a następnie podwójnie kliknąć symbol Actor_1 pojawi się arkusz właściwości aktora. W polu Name wpisać Klient.
- W podobny wprowadzić nazwy dwóch pozostałych aktorów *Kierownik* oraz *Przedstawiciel handlowy.* Diagram powinien wyglądać jak poniższym rysunku:



2.3. Tworzenie powiązań pomiędzy obiektami diagramu

Związki pomiędzy aktorami a przypadkami użycia mogą być rysowane w dwóch kierunkach, w zależności od tego czy aktor jest aktorem pierwotnym (*primary actor*) czy też aktorem wtórnym (*secondary actor*). Aktor pierwotny to aktor inicjujący określone działanie opisane przypadkiem użycia, aktor wtórny – bierze udział w działaniu nie będąc jego inicjatorem, pomaga w jego wykonaniu.

- w przypadku aktora pierwotnego powiązanie skierowane jest od aktora do przypadku użycia,
- w przypadku aktora wtórnego powiązanie jest skierowane od przypadku użycia do aktora.

W celu utworzenia powiązań między obiektami:

- Kliknąć symbol Association (powiązanie) na palecie narzędzi.
- Kliknąć na symbolu *Klienta*, trzymając wciśnięty lewy przycisk myszy przesunąć kursor nad symbol przypadku użycia *Poszukiwanie produktu* i następnie zwolnić przycisk myszy. Pojawi się symbol powiązania pomiędzy aktorem, którym jest *Klient* i przypadkiem użycia *Poszukiwanie produktu*.
- Kliknąć prawym przyciskiem myszy a następnie kliknąć podwójnie w symbol utworzonego przed chwilą powiązania. Pojawi się arkusz właściwości powiązania w jego polu *Name* wpisać *Szukanie produktu*. Kliknąć *OK*.

	D // /		<i>,</i> .	1		• •
•	Powtorzyc	powyzsze	czvnnosci w	celu utworzenia	nastepulacych	powiazan:
					I JUJ	F

Od	Do	Nazwa powiązania	Typ aktora
Kierownik	Przeszukiwanie	Przeszukiwanie	Pierwotny
	katalogu		
Obsługa sprzedaży	Kierownik	Statystyka	Wtórny
Przedstawiciel	Obsługa sprzedaży	Udzielanie	Pierwotny
handlowy		informacji	
Kupowanie produktu	Przedstawiciel	Składanie	Wtórny
	handlowy	zamówienia	
Klient	Kupowanie	Zamówienie	Pierwotny
	produktu	produktu	

Diagram powinien wyglądać jak poniższym rysunku:



2.4. Tworzenie zależności pomiędzy przypadkami użycia

- Kliknąć symbol *Dependency* (*zależność*) na palecie narzędzi. Połączyć przypadki użycia *Poszukiwanie produktu* i *Przeszukiwanie katalogu*. Pomiędzy obydwoma przypadkami użycia pojawi się symbol zależności. Kliknąć prawym przyciskiem myszy, aby zwolnić kursor.
- Kliknąć podwójnie utworzony przed chwilą symbol zależności pojawi się jego arkusz właściwości. W polu *Name* wpisać *wymaga*, zaś w polu *Stereotype* (typ zależności) wpisać *używa*. Przypadek użycia *Poszukiwanie produktu* używa innego przypadku użycia, jakim jest *Przeszukiwanie katalogu*.

Typy zależności łączących przypadki użycia:

- *uses* (*używa*) przypadek użycia *B* używa przypadku *A*, gdy przypadek *A* jest integralną częścią przypadku *B*. Na przykład przypadek użycia *sprzedawanie sprzętu* używa przypadku użycia *wystawianie gwarancji*.
- *extends* (*rozszerza*) przypadek użycia *B* rozszerza przypadek użycia *A*, jeśli *B* jest jakąś bardziej skomplikowaną, opcjonalną sytuacją wyjątkową, która zachodzi tylko w specjalnych warunkach. Na przykład przypadek użycia *wpisanie klienta do ewidencji* rozszerza przypadek użycia *sprzedawanie sprzętu* (zakładamy, że najczęściej klient się już w tej ewidencji znajduje).
 - Powtórzyć poprzednie kroki w celu stworzenia zależności łączącej przypadki użycia Obsługa sprzedaży oraz Przeszukiwanie katalogu. W polu Name wpisać wykorzystuje zaś w polu Stereotype (typ zależności) wpisać używa. Diagram powinien wyglądać jak poniżej:



2.5. Tworzenie klasy implementującej przypadek użycia

- Kliknąć podwójnie symbol przypadku użycia *Przeszukiwanie katalogu* pojawi się okno jego właściwości.
- Kliknąć zakładkę *Implementation Classes*, a następnie ikonę *Create a New Class* (trzecia od lewej) pojawi się arkusz właściwości nowej klasy.
- W polu *Name* wpisać *Katalog*, a następnie *OK*. Na stronie *Implementation Classes* pojawi się nowa klasa *Katalog*. Kliknąć *OK*.

- Rozwinąć kategorię *Classes* w oknie przeglądarki (*Browser*). Kliknąć podwójnie w gałęzi klasy *Katalog* pojawi się arkusz jej właściwości.
- Kliknąć zakładkę *Dependencies* (*zależności*) widzimy tutaj nazwę przypadku użycia *Przeszukiwanie katalogu*, który jest implementowany przez klasę *Katalog*. Kliknąć *OK*.
- Zapisać model na dysku.

3. Tworzenie diagramu kolaboracji (collaboration diagram)

Istotą **diagramu kolaboracji (współpracy**), podobnie jak omawianego później diagramu sekwencji, jest przedstawienie przepływu komunikatów pomiędzy obiektami. Diagram kolaboracji opisuje w jaki sposób system osiąga to, co zostało opisane przez diagram przypadków użycia.

Kolaboracja (współpraca) między obiektami obejmuje dwa aspekty:

- **statyczną strukturę** uczestniczących w niej obiektów (związki, atrybuty i operacje) jest to tzw. kontekst kolaboracji,
- **sekwencję komunikatów** wymienianych pomiędzy obiektami w celu realizacji konkretnego zadania.

W poniższym przypadku rozpatrujemy zamodelowany wcześniej przypadek użycia *Poszukiwanie produktu*. Diagram kolaboracji pokazywał będzie klienta łączącego się z internetem i wybierającego z katalogu na stronie księgarni żądaną pozycję książkową.

3.1. Definiowanie opcji wyświetlania diagramu kolaboracji

- Kliknąć prawym przyciskiem myszy gałąź naszego modelu w oknie przeglądarki (*Browser*) a następnie wybrać opcję *New/Collaboration Diagram*.
- W polu *Name* arkusza właściwości diagramu wpisać *Przeszukiwanie katalogu*.
- Z górnego menu wybrać polecenie *Tools/Display Preferences*. W oknie preferencji wyświetlania zaznaczyć podgałąź *Message* znajdującą się w gałęzi *Object View* a następnie zaznaczyć wszystkie jej opcje: *Show name*, *Replace by operation name* oraz *Show condition prefix*.
- Kliknąć przycisk *Set As Default*, a następnie *OK*.

3.2. Tworzenie obiektów w diagramie kolaboracji

Tworzenie diagramu kolaboracji rozpoczniemy od narysowania klienta łączącego się z siecią w celu zamówienia książki, a następnie dodamy obiekty wchodzące w skład systemu.

- Rozwinąć gałąź *Actors* w oknie przeglądarki (*Browser*), zaznaczyć symbol *Klient* i przeciągnąć go do lewej części pustego okna diagramu kolaboracji.
- Kliknąć symbol *Object* znajdujący się na palecie narzędzi, a następnie kliknąć na diagramie z prawej strony symbolu *Klienta* w miejscu kliknięcia pojawi się symbol obiektu.
- Dodać do diagramu jeszcze jeden obiekt po prawej stronie pierwszego obiektu. Kliknąć prawym przyciskiem myszy, aby zwolnić symbol obiektu. Diagram powinien wyglądać następująco:



• Kliknąć na symbolu pierwszego obiektu, a następnie na jego nazwie. W polu nazwy wpisać *Zapotrzebowanie*. W podobny sposób zmienić nazwę drugiego obiektu na *Katalog*.

3.3. Tworzenie powiązań oraz komunikatów

Powiązanie (*instance link*) jest odpowiednikami **asocjacji** z diagramu klas. W przypadku diagramów kolaboracji reprezentuje ono związek zachodzący pomiędzy dwoma obiektami lub pomiędzy obiektem a aktorem. Na diagramie zaznaczane jest w postaci ciągłej linii łączącej dane elementy.

Komunikat (*message*) jest informacją wysyłaną od nadawcy do odbiorcy. Na diagramie rysowany jest w formie strzałki skierowanej od nadawcy do odbiorcy.

- Kliknąć symbol *Message* (*komunikat*) znajdujący się na palecie narzędzi, a następnie połączyć *Klienta* z obiektem *Zapotrzebowanie* pomiędzy nimi pojawi się symbol powiązania **oraz** symbol komunikatu.
- W podobny sposób połączyć obiekt *Zapotrzebowanie* z obiektem *Katalog*.
- Utworzyć dwa kolejne komunikaty pomiędzy obiektami, skierowane w kierunku przeciwnym niż opisane powyżej. Diagram powinien wyglądać następująco:



- Kliknąć podwójnie symbol pierwszego komunikatu. W polu *Name* jego arkusza właściwości wpisać nazwę *Wprowadź szukane pozycje*. Upewnić się, że numer komunikatu (*Sequence number*) ma wartość 1. Kliknąć *OK*.
- W podobny sposób zmodyfikować nazwy kolejnych komunikatów:

Nadawca	Odbiorca	Nazwa komunikatu	Nr komunikatu
Zapotrzebowanie	Katalog	Wyślij zapotrzebowanie	2
Katalog	Zapotrzebowanie	Zwróć rezultat	3
Zapotrzebowanie	Klient	Zwróć listę dostępnych pozycji	4

• Diagram powinien wyglądać jak poniżej:



3.4. Tworzenie powiązania pomiędzy obiektem a klasą

W niniejszym punkcie powiążemy obiekt *Zapotrzebowanie* z klasą *Zapotrzebowanie*, która zostanie utworzona z diagramu kolaboracji. W tym celu:

- Kliknąć podwójnie w symbol obiektu *Zapotrzebowanie* pojawi się arkusz jego właściwości.
- Kliknąć ikonę *Create (utwórz klasę)* znajdującą się obok listy rozwijalnej *Class* pojawi się arkusz właściwości nowej klasy.
- W polu *Name* wpisać nazwę nowej klasy *Zapotrzebowanie*, następnie kliknąć *OK* w każdym z okien dialogowych. Utworzyliśmy właśnie związek instancjacji pomiędzy obiektem *Zapotrzebowanie* a klasą o tej samej nazwie. Symbol obiektu *Zapotrzebowanie* zawiera teraz nazwę obiektu, po niej dwukropek, a następnie nazwę klasy do której ten obiekt należy.
- Rozwinąć kategorię *Classes* w oknie przeglądarki (*Browser*). Kliknąć podwójnie w gałęzi klasy *Zapotrzebowanie* pojawi się arkusz jej właściwości.
- Kliknąć zakładkę *Dependencies* (*zależności*) widzimy tutaj nazwę obiektu *Zapotrzebowanie*, który należy do tej klasy. Kliknąć *OK*.
- Zapisać model na dysku.

4. Tworzenie diagramu sekwencji (sequence diagram)

Diagram sekwencji jest drugim, obok omawianego w poprzednim punkcie diagramu kolaboracji, wprowadzonym w UML typem diagramu interakcji. Diagram sekwencji posiada dwa wymiary: pionowy - reprezentujący czas, oraz poziomy - reprezentujący obiekty. Pokazuje on uporządkowaną w czasie sekwencję komunikatów wymienianych pomiędzy obiektami.

4.1. Definiowanie opcji wyświetlania diagramu sekwencji

- Kliknąć prawym przyciskiem myszy na tle jednego z utworzonych poprzednio diagramów. Z menu kontekstowego wybrać polecenie *Diagram/New Diagram/Sequence Diagram* pojawi się okno dialogowe nowego diagramu sekwencji.
- W polu *Name* wpisać *Obsługa zamówienia*, kliknąć *OK*.
- Kliknąć prawym przyciskiem myszy na tle nowo utworzonego okna diagramu sekwencji. Z menu kontekstowego wybrać opcję *Display Preferences* otworzy się okno preferencji wyświetlania.
- Zaznaczyć podgałąź *Message* znajdującą się w gałęzi *Object View* a następnie odznaczyć opcję *Show activation attachment*. Wszystkie pozostałe opcje powinny być zaznaczone.
- Kliknąć przycisk *Set As Default,* a następnie *OK*.

4.2. Tworzenie obiektów w diagramie sekwencji

- Z okna przeglądarki (*Browser*) przeciągnąć symbol aktora *Klient* oraz symbol klasy *Katalog* i umieścić je na diagramie.
- Korzystając z palety narzędzi dodać do diagramu trzy dodatkowe obiekty i rozmieścić jak na poniższym rysunku:



• Kliknąć podwójnie w każdy z nowych symboli obiektów i wprowadzić ich nazwy, kolejno od lewej strony diagramu: *Koszyk*, (*Katalog*), *Weryfikacja*, *Zamówienie*.

4.3. Tworzenie komunikatów w diagramie sekwencji

Uwaga: ponieważ w diagramie sekwencji istotna jest chronologia, komunikaty powinny być rysowane kolejno jeden pod drugim, zgodnie z ich wzajemnym położeniem na diagramie.

- Kliknąć symbol *Message* (*komunikat*) na palecie narzędzi, następnie kliknąć "linię życia" (*lifeline*) prowadzącą od *Klienta* i trzymając wciśnięty lewy klawisz myszy przeciągnąć symbol wiadomości do "linii życia" prowadzącej od obiektu *Koszyk*.
- Kliknąć prawym przyciskiem myszy aby zwolnić symbol wiadomości, a następnie kliknąć podwójnie w dodany przed chwilą do diagramu symbol wiadomości. W arkuszu jego właściwości wpisać nazwę *Włóż do koszyka*, następnie kliknąć *OK*.
- W podobny sposób utworzyć komunikat pomiędzy obiektem *Koszyk* a obiektem *Katalog* i nazwać go *Sprawdź czy pozycja dostępna*.
- Kliknąć symbol *Return Message* (*komunikat powrotny*) na palecie narzędzi, dodać do diagramu komunikat od obiektu *Katalog* do obiektu *Koszyk* i nazwać go *Informacja o dostępności*.
- Utworzyć kolejny komunikat (*Message*) łączący *Klienta* z *Koszykiem* i nazwać go *Kupuj zawartość koszyka*. Diagram powinien wyglądać jak na poniższym rysunku:



Utworzymy teraz komunikat który zainicjuje tworzenie obiektu.

- Kliknąć symbol *Message* na palecie narzędzi. Poprowadzić skośnie linię łączącą "linię życia" prowadzącą od *Koszyka* z (uwaga!) symbolem obiektu *Weryfikacja*. Symbol obiektu przesunął się w dół obiekt *Weryfikacja* będzie tworzony w tej chwili czasu.
- Kliknąć prawym przyciskiem myszy, następnie kliknąć podwójnie symbol komunikatu dodany przed chwilą do diagramu. W arkuszu właściwości wpisać nazwę komunikatu *Weryfikuj klienta*.
- Kliknąć zakładkę *Detail* zaznaczony jest element *Create* listy rozwijalnej *Action*.
- Kliknąć *OK* diagram powinien wyglądać jak na rysunku:



- Kliknąć symbol *Return Message* (*komunikat powrotny*) na palecie narzędzi, dodać do diagramu komunikat prowadzący od "linii życia" obiektu *Weryfikacja* do "linii życia" obiektu *Koszyk* i nazwać go *Wynik weryfikacji klienta*.
- Utworzyć komunikat wysyłany przez obiekt Koszyk, tworzący obiekt Zamówienie i nazwać go Wysłanie zamówienia.

4.4. Dodawanie metody do klasy w oparciu o diagram sekwencji

- Kliknąć podwójnie symbol komunikatu *Sprawdź czy dostępna*, następnie kliknąć zakładkę *Detail*.
- Kliknąć ikonę *Create* znajdującą się obok listy rozwijalnej *Operation* pojawi się arkusz właściwości operacji.
- Domyślną nazwą operacji jest nazwa komunikatu. W polu *Name* wpisać nową nazwę operacji *Zwróć listę pozycji*. Kliknąć *OK* w każdym z okien dialogowych. Na diagramie w miejscu nazwy komunikatu wyświetla się teraz nazwa operacji.
- Zapisać model na dysku. Ostatecznie diagram powinien wyglądać jak na rysunku:



5. Tworzenie diagramu stanów (statechart diagram)

Diagram stanów reprezentuje stany pewnego procesu oraz przejścia pomiędzy tymi stanami. W notacji *UML* diagramy stanów są raczej klasycznymi diagramami przepływu sterowania niż diagramami stanów znanymi np. z teorii automatów.

W tym punkcie zaprojektujemy diagram przepływu stanów dotyczący klasy *Weryfikacja*, która pojawiła się w wyniku poprzednio przeprowadzonej analizy. Model opisywać będzie proces weryfikacji tożsamości klienta, który chce zakupić książki włożone do wirtualnego "koszyka" w sklepie internetowym.

5.1. Definiowanie opcji wyświetlania diagramu stanów

- Kliknąć prawym przyciskiem myszy na tle jednego z utworzonych poprzednio diagramów. Z menu kontekstowego wybrać polecenie *Diagram/New Diagram/Statechart Diagram* – pojawi się okno dialogowe nowego diagramu.
- W polu *Name* wpisać *Procedura weryfikacji*, kliknąć *OK*.
- Kliknąć prawym przyciskiem myszy na tle nowo utworzonego okna diagramu sekwencji. Z menu kontekstowego wybrać opcję *Display Preferences* otworzy się okno preferencji wyświetlania.
- Zaznaczyć podgałąź *State* znajdującą się w gałęzi *Object View* a następnie zaznaczyć opcję *Show Actions/All* w oknie po prawej stronie. Wszystkie pozostałe opcje powinny być nieaktywne.
- Kliknąć przycisk *Set As Default*, a następnie *OK*.

5.2. Tworzenie stanów

- Kliknąć symbol *Start* znajdujący się na palecie narzędzi, a następnie kliknąć z prawej strony okna diagramu symbol startu pojawi się na diagramie.
- W podobny sposób umieścić na diagramie dwa symbole stanów (*State*) diagram powinien wyglądać jak poniżej:

State_1	State_2	

 Zwolnić prawym przyciskiem myszy symbol stanu i kliknąć podwójnie na symbolu *State_1* dodanym przed chwilą do diagramu. Wpisać jego nazwę – *Edycja*. W podobny sposób ustalić nazwę drugiego stanu – *Zatwierdzanie*.

5.3. Tworzenie akcji i zdarzeń

- Kliknąć podwójnie symbol stanu *Edycja*, a następnie zakładkę Actions
- Kliknąć ikonę *Insert a Row* (*wstaw wiersz*) w polu *Name* wpisać *zainicjuj timer*. W kolumnie *Trigger Event* z listy rozwijalnej wybrać *entry*. Każda akcja zwiera właściwość *Trigger Event* zawierającą opis zdarzenia inicjującego akcję.
- Powtórzyć powyższe kroki w celu dodania do stanu *Zatwierdzanie* następujących akcji i zdarzeń:

Action	Trigger Event
wyświetl komunikat	entry
zapytaj serwer	do
ukryj komunikat	exit

• Diagram powinien wyglądać jak poniżej:



5.4. Tworzenie przejść między stanami

• Kliknąć symbol *End* na palecie narzędzi. Rozmieścić symbole końca jak na poniższym rysunku:

	Zatwierdzanie	
Edycja	entry/ wyswietl komunikat	2
entry/ zainicjuj timer	do/ zapytaj serwer)
	(💼)	

• Kliknąć symbol *Transition* (*przejście*) na palecie narzędzi. Połączyć symbol startu na diagramie z symbolem stanu *Edycja*. W podobny sposób połączyć pozostałe elementy w sposób pokazany na poniższym rysunku:



- Kliknąć podwójnie symbol przejścia łączącego *Edycję* z *Zatwierdzaniem* pojawi się arkusz właściwości przejścia.
- Kliknąć zakładkę *Condition* (*warunek*) i w polu *Alias* wpisać *Zatwierdzanie*. Kliknąć *OK* warunek wyświetlany jest w nawiasach kwadratowych.
- W podobny sposób powiązać warunki z pozostałymi przejściami, tak, aby diagram wyglądał jak na poniższym rysunku:



• Zapisać model na dysku.

6. Tworzenie diagramu aktywności (activity diagram)

Diagram aktywności (*czynności*) jest szczególnym przypadkiem diagramu stanów. Jest on modelem przedstawiającym przepływy sterujące (*control flows*) pomiędzy zdarzeniami zachodzącymi w systemie (zwanymi *aktywnościami, czynnościami*), począwszy od punktu startowego aż do kilku potencjalnych punktów końcowych. Pojawiają się w nim (aczkolwiek w dość prymitywnej formie) elementy synchronizacji równoległych procesów.

Tworzony przez nas w ćwiczeniu 1. model procesów biznesowych był właśnie rodzajem takiego diagramu aktywności. W tym punkcie zaprojektujemy diagram aktywności i utworzymy wymagane obiekty w celu opisania szczegółów zachodzących w przypadku użycia *Kupowanie produktu*.

6.1. Tworzenie aktywności

- Kliknąć prawym przyciskiem myszy węzeł modelu w oknie przeglądarki (*Browser*), a następnie wybrać opcję *New/Activity Diagram*. W polu *Name* okna właściwości diagramu wpisać *Proces płatności*. Kliknąć *OK*.
- Kliknąć symbol *Start* na palecie narzędzi, a następnie kliknąć w górnej części diagramu w miejscu kliknięcia pojawi się symbol startu.
- W podobny sposób umieścić na diagramie pięć symboli *Activity* (*aktywności*) jak na rysunku poniżej:



- Kliknąć prawym przyciskiem myszy aby zwolnić symbol aktywności, a następnie podwójnie kliknąć symbol *Activity_1* na diagramie. W arkuszu właściwości wpisać nazwę aktywności *Obsłuż płatność*.
- W podobny sposób ustal nazwy kolejnych aktywności: Zgłoś błąd, Zarejestruj błąd płatności, Aktualizuj konto klienta, Zarejestruj udaną płatność.



- Kliknąć symbol *Decision* (*decyzja*) na palecie narzędzi i umieścić go bezpośrednio pod symbolem aktywności *Obsłuż płatność*. Jeśli symbol jest zbyt duży – zmniejszyć jego rozmiar.
- Kliknąć symbol *Synchronization* (*synchronizacja*) na palecie narzędzi i umieścić go bezpośrednio pod symbolami aktywności *Zarejestruj błąd płatności* oraz *Zarejestruj udaną płatność*.
- Kliknąć symbol *Object state (stan obiektu)* na palecie narzędzi i umieścić go pomiędzy symbolami aktywności *Aktualizuj konto klienta* oraz *Zarejestruj udaną płatność*.

Stan obiektu jest połączeniem obiektu jako instancji klasy oraz stanu. Reprezentuje obiekt znajdujący się w określonym stanie.

- Kliknąć podwójnie symbol *Object state* umieszczony przed chwilą na diagramie. W polu *Name* jego arkusza właściwości wpisać *Zaktualizowane*.
- Kliknąć ikonę *Create* znajdującą się obok listy rozwijalnej *Object*. Usunąć domyślną nazwę obiektu w polu *Name* arkusza właściwości obiektu.
- Kliknąć ikonę *Create* znajdującą się obok listy rozwijalnej *Class*. W polu *Name* arkusza właściwości klasy wpisać *Konto klienta*. Kliknąć *OK* w każdym z okien dialogowych.

Utworzyliśmy nowy obiekt i nową klasę. Obiekt nie ma nazwy, ale jest instancją klasy *Konto klienta* i znajduje się w stanie, który zdefiniowaliśmy jako *Zaktualizowane*. Diagram powinien wyglądać jak na poniższym rysunku:

 Obsłuż płatność

 Zgłoś błąd

 Aktualizuj konto klienta

 [Zarejestruj błąd płatności]

 Zarejestruj udaną płatność

6.2. Tworzenie przejść pomiędzy aktywnościami

- Kliknąć symbol *Transition* na palecie narzędzi. Połączyć symbol startu z symbolem aktywności *Obsłuż platność.*
- W podobny sposób połączyć pozostałe elementy diagramu jak na poniższym rysunku:



- Kliknąć symbol *End* znajdujący się na palecie narzędzi, a następnie kliknąć na diagramie poniżej symbolu synchronizacji symbol końca pojawi się na diagramie.
- Kliknąć symbol *Transition* na palecie narzędzi i połączyć symbol synchronizacji na diagramie z utworzonym przed chwilą symbolem końca. Kliknąć prawym przyciskiem myszy, aby zwolnić symbol.
- Diagram powinien wyglądać jak poniżej:



6.3. Dekompozycja aktywności

Jeśli aktywność opisuje czynność dość złożoną, może zostać zdekomponowana (rozłożona). W poniższym przykładzie zdekomponujemy aktywność *Obsłuż płatność* w celu pokazania nowych aktywności, dokładniej opisujących jej działanie.

- Trzymając wciśnięty klawisz CTRL kliknąć podwójnie na symbolu aktywności *Obsłuż płatność* otworzy się nowe okno diagramu podrzędnego.
- Kliknąć symbol *Start* na palecie narzędzi, a następnie kliknąć w górnej części nowego okna diagramu podrzędnego.
- W podobny sposób dodać do diagramu nowe elementy i ustalić ich nazwy, tak, aby diagram miał postać jak na poniższym rysunku:



- Kliknąć prawym przyciskiem myszy w oknie diagramu, z menu kontekstowego wybrać opcję *Diagram/Go Up One Level/Proces Płatności* wrócimy w ten sposób do głównego okna diagramu aktywności. Zauważmy, że symbol aktywności *Obsłuż płatność* ma w prawym dolnym rogu ikonę symbolizującą aktywność złożoną.
- Zapisać model na dysku.

7. Tworzenie diagramu klas (class diagram)

Diagram klas jest centralnym pojęciem we wszystkich znanych metodykach obiektowych, także w UML. Opisuje on statyczną strukturę modelu, obejmując zbiór klas, pakietów, interfejsów oraz powiązań między nimi zachodzących.

W porównaniu z diagramami związków encji, diagramy klas wprowadzają m.in. metody (funkcje) przypisane do określonych klas. Diagram klas pokazuje klasy w postaci pewnych oznaczeń graficzno-językowych powiązanych w sieć zależnościami należącymi do trzech kategorii:

Asocjacja (association) – dowolny związek pomiędzy klasami, który ma znaczenie dla modelowania,

Agregacja (*aggregation*) – szczególny przypadek asocjacji, odwzorowujący stosunek całośćczęść pomiędzy obiektami z modelowanej dziedziny przedmiotowej,

Dziedziczenie (inheritance) – związek generalizacji-specjalizacji pomiędzy klasami.

W niniejszym punkcie utworzymy prosty diagram klas wykorzystując dwie metody: budowę od podstaw z elementów składowych oraz rewersowanie kodu napisanego w języku *Java*. Następnie w oparciu o nasz diagram klas wygenerujemy fizyczny model danych (*PDM*), dzięki czemu możliwa będzie jego implementacja w formie relacyjnej bazy danych. W końcowej części z diagramu klas utworzymy diagram obiektów.

7.1. Definiowanie opcji wyświetlania diagramu klas

- Kliknąć prawym przyciskiem myszy gałąź modelu w oknie przeglądarki (*Browser*), a następnie wybrać z menu kontekstowego opcję *New/Class Diagram*. W polu *Name* okna właściwości diagramu wpisać *Fakturowanie*. Kliknąć *OK*.
- Kliknąć prawym przyciskiem myszy w oknie diagramu i z menu kontekstowego wybrać opcję *Display Preferences*. W oknie preferencji wyświetlania wybrać gałąź *Class* i zaznaczyć następujące opcje: *Show attributes – All, Show operations – All, Show stereotype, Show constraints, Show inner classifiers*. Kliknąć przycisk *Set As Default*.
- Wybrać gałąź Attribute i zaznaczyć następujące opcje: Show visibility Use markers, Show stereotype, Show data type i Show initial value. Kliknąć przycisk Set As Default.
- Wybrać gałąź Association i zaznaczyć następujące opcje: Show multiplicity string, Show constraints, Show role names, Show ordering. Kliknąć przycisk Set As Default, a następnie OK.

7.2. Dodawanie klas do diagramu

- Kliknąć symbol *Class* na palecie narzędzi i utworzyć na diagramie dwa symbole klas, jeden obok drugiego.
- Kliknąć podwójnie pierwszy symbol klasy otworzy się arkusz jego właściwości. W polu *Name* wpisać *Klient*, kliknąć przycisk *Zastosuj*.
- Kliknąć zakładkę *Attributes*, a następnie ikonę *Insert a row*. Kliknąć podwójnie strzałkę na początku linii, a następnie *Tak* aby potwierdzić chęć utworzenia obiektu. Otworzy się arkusz właściwości atrybutu.
- W polu *Name* wpisać nazwę atrybutu: *IDKlienta*.
- Z listy rozwijalnej *Data Type* wybrać nazwę typu: *long*.
- Kliknąć zakładkę *Detail* a następnie zaznaczyć opcję *Primary identifier* atrybut *IDKlienta* będzie głównym identyfikatorem klasy *Klient*
- Kliknąć *OK* aby powrócić do strony *Attributes*. Dodać dwa kolejne atrybuty: *nazwa* oraz *adres*. Ich typem niech będzie łańcuch znakowy *Javy java.lang.String*. Tym razem nie zaznaczać opcji *Primary identifier –* atrybuty te nie będą identyfikatorami !
- Kliknąć *OK* w każdym z okien dialogowych.
- Powtórzyć powyższe kroki w celu ustalenia właściwości drugiej klasy:

Nazwa klasy	Nazwa atrybutu	Typ atrybutu	Główny identyfikator
Zamowienie	IDZamowienia	long	tak
	data_zamowienia	Java.util.Date	nie

• Diagram klas powinien wyglądać następująco:

Klient		
+ IDKlienta : long + nazwa : java.lang.String + adres : java.lang.String	1	<u>+</u> <u>IDZamowienia</u> <u>: long</u> + data_zamowienia : java.util.Dat

7.3. Tworzenie asocjacji (powiązań) między klasami

Typem asocjacji czyli powiązania między klasami, które utworzymy w tym punkcie będzie tzw. *kompozycja* (*composition*) – jest ona mocniejszą formą *agregacji*. Klasycznym przypadkiem związku kompozycji jest np. związek zachodzący pomiędzy *zamówieniem* i *pozycją zamówienia*: pozycja zamówienia nie występuje oddzielnie (poza zamówieniem), nie podlega przenoszeniu od jednego zamówienia do innego i znika w momencie kasowania zamówienia.

- Kliknąć symbol asocjacji (*Associacion*) znajdujący się na palecie narzędzi. Przeciągnąć linię od *Klienta* do *Zamowienia*. Zwolnić symbol asocjacji klikając prawym przyciskiem myszy.
- Kliknąć podwójnie symbol asocjacji na diagramie, a następnie zakładkę *Detail*. Usunąć zaznaczenie opcji *Navigable* w panelu obiektu *Zamowienie*.
- Zaznaczyć opcję *Aggregation/Composition* i wybrać przełącznik radiowy *Composition.* Kliknąć *OK.* Diagram powinien wyglądać jak na rysunku:



7.4. Rewersowanie pliku Javy

- Pobrać ze strony <u>http://po.opole.pl/~bartecki/projektowanie/faktura.java</u> plik zawierający definicję klasy *Faktura*, i zapisać go w katalogu roboczym.
- Zapoznać się z jego zawartością.
- Z górnego menu wybrać polecenie *Language/Reverse Engineer Java*, a następnie kliknąć zakładkę *Options*.
- Zaznaczyć opcje *Create symbols* i *Create associations*. Wszystkie pozostałe opcje nie powinny być zaznaczone.
- Kliknąć zakładkę *Selection* i z listy rozwijalnej *Reverse* wybrać *Java files*. Kliknąć przycisk *Add* i w oknie wyboru plików wybrać plik *faktura.java*.
- Kliknąć przycisk Otwórz, a następnie OK. Pojawi się okno dialogowe Merge Models. Panel po lewej stronie zawiera model źródłowy (From Model) oraz model docelowy (To Model). Model źródłowy zawiera elementy rewersowanego pliku Javy – aby je zobaczyć należy kliknąć gałąź Classes. Widzimy tu klasę Faktura, jej atrybuty i metody.
- Kliknąć *OK* na diagramie pojawi się symbol rewersowanej klasy *Faktura*.
- Z górnego menu wybrać polecenie *Tools/Rebuild Data Type Links* w oknie dialogowym które się pojawi wszystkie opcje powinny być zaznaczone.
- Kliknąć *OK.* Pomiędzy klasami *Faktura* i *Zamówienie* automatycznie utworzona zostanie asocjacja, której źródłem jest atrybut klasy *Faktura* o nazwie *IDZamowienia*, należący do klasy *Zamówienie*.
- Diagram powinien mieć postać jak na poniższym rysunku:



7.5. Konwersja do modelu fizycznego (PDM)

- Z górnego menu wybrać polecenie *Tools/Generate Physical Data Model*. Zaznaczyć opcję *Generate New Physical Data Model*, w polu *Name* wpisać *Tutorial2*.
- Kliknąć zakładkę *Detail.* Zaznaczyć opcje *Save generation dependencies* oraz *O/R Mapping.* Wszystkie pozostałe opcje nie powinny być zaznaczone.
- Kliknąć *OK* w nowym oknie diagramu pojawi się fizyczny model danych, wygenerowany na podstawie naszego diagramu klas:



- Trzy klasy z modelu obiektowego zostały przekształcone do tabel w modelu fizycznym, zaś dwie asocjacje w powiązania między tabelami.
- Zapisać utworzony model w pliku *Tutorial2.pdm*.
- Powrócić do diagramu klas, np. wybierając odpowiednie okno z górnego menu *Window* lub klikając na odpowiedniej gałęzi w oknie przeglądarki (*Browser*).
- Kliknąć podwójnie symbol klasy *Klient* pojawi się jej arkusz właściwości. Kliknąć zakładkę *Mapping*, a następnie *Class Sources* u dołu strony. Kolumny *Name* oraz *Code* wskazują na nazwę oraz kod odpowiedniej tabeli w modelu fizycznym, powstałej w wyniku "mapowania" z klasy *Klient*.
- Kliknąć zakładkę *Attributes Mapping* u dołu strony widzimy nazwy atrybutów klasy i przyporządkowane im nazwy kolumn w odpowiedniej tabeli modelu fizycznego.
- Kliknąć *OK*, zapisać model na dysku.

7.6. Tworzenie diagramu obiektów na podstawie diagramu klas

- Kliknąć prawym przyciskiem myszy w oknie diagramu, z menu kontekstowego wybrać opcję *Diagram/New Diagram/Object Diagram.*
- W polu *Name* okna dialogowego wpisać *Zamowienie*, kliknąć *OK*.
- Rozwinąć gałąź *Associations* w oknie przeglądarki (*Browser*), przeciągnąć do nowego okna dialogowego symbol asocjacji utworzonej pomiędzy klasami *Klient* i *Zamowienie* (*Association_1*).
- Na diagramie wyświetlone zostaną dwa nowo utworzone obiekty będące instancjami klas, odpowiednio *Klient* oraz *Zamowienie*, a także powiązanie pomiędzy tymi obiektami:



• Zapisać model na dysku.

8. Tworzenie diagramu komponentów (component diagram)

Diagram komponentów jest obok diagramu wdrożeniowego, który zostanie omówiony w następnym punkcie, typem **diagramu implementacyjnego**. Diagramy implementacyjne pokazują niektóre aspekty implementacji systemu informatycznego, włączając w to strukturę kodu źródłowego oraz strukturę kodu czasu wykonania (*run-time code structure*).

Komponent reprezentuje fizyczny fragment implementacji systemu, np. kod programu (źródłowy, binarny, wykonywalny), skrypt, plik wsadowy. Jest on niezależnym fragmentem oprogramowania powstałym w określonym celu, ale nie dla konkretnej aplikacji. Komponenty umożliwiają budowę oprogramowania na zasadzie "czarnej skrzynki".

Diagram komponentów pokazuje zależności pomiędzy komponentami oprogramowania. Diagram komponentów jest przedstawiany jako graf, gdzie węzłami są komponenty, zaś strzałki (z linią przerywaną) prowadzą do klienta pewnej informacji od jej dostawcy.

Interfejs – zewnętrzna charakterystyka obiektu, widoczna dla jego użytkownika. Interfejs reprezentuje operacje, które można wykonać na obiekcie, oraz atrybuty, których wartości są dostępne z zewnątrz.

Implementacja – określa wewnętrzne aspekty obiektu danego typu. Składa się z reprezentacji stanu obiektu oraz zbioru metod (funkcji działających na obiektach).

Oddzielenie interfejsu od implementacji jest istotą hermetyzacji.

EJB (*Enterprise JavaBean*) – architektura komponentów oprogramowania firmy *Sun Microsystems*, przeznaczona do wytwarzania aplikacji biznesowych opartych na komponentach. Aplikacje napisane przy użyciu *EJB* są skalowalne, przenaszalne i bezpieczne pod względem wieloużytkowości (*definicja według firmy Sun Microsystems*).

8.1. Tworzenie EJB z wykorzystaniem asystenta (Enterprise Java Bean Wizard)

- Kliknąć prawym przyciskiem myszy symbol klasy *Faktura* na utworzonym wcześniej diagramie klas. Z menu kontekstowego wybrać *Create Enterprise Java Bean* pojawi się okno asystenta (*Enterprise Java Bean Wizard*).
- W polu *Name* wpisać *Faktura*, kliknąć przycisk *Dalej*. Z listy rozwijalnej *Component Type* wybrać *EJB CMP Entity Bean*, kliknąć *Dalej*.
- Z listy rozwijalnej *Bean Class* wybrać nazwę klasy *FakturaBean*. W kolejnych oknach wybierać *Dalej*.
- Gdy pojawi się okno jak poniżej, zaznaczyć opcje *Create Symbol in New Diagram* oraz *Create Class Diagram for component classifiers*.



• Kliknąć *Zakończ* – nasz klasa *Faktura* automatycznie stanie się klasą *EJB Bean*. Jej nazwa zostanie zmieniona na *FakturaBean*, dodane zostaną dodatkowe atrybuty oraz metody:

	< <ejbentity>></ejbentity>			
FakturaBean				
		{abstract}		
+ IDFaktury	: long			
+ dataFaktury	: java.util.Date			
+ IDZamowienia	: Zamowienie			
 ejbContext 	: EntityContext			
+	drukuj ()	: void		
+	wyslij ()	: void		
+	zwrocIDFaktury ()	: long		
+	ustawIDFaktury (long nowyIDFaktury)	: void		
+	zwrocDateFaktury ()	: java.util.Date		
+	ustawdateFaktury (java.util.Date nowaDataFaktury)	: void		
+	zwrocZamowienie ()	: Zamowienie		
+	ustawZamowienie (Zamowienie noweZamowienie)	: void		
+ < <constructor>></constructor>	FakturaBean ()			
+	ejbActivate ()	: void		
+	ejbLoad ()	: void		
+	ejbPassivate ()	: void		
+	ejbRemove ()	: void		
+	ejbStore ()	: void		
+	setEntityContext (EntityContext ctx)	: void		
+	unsetEntityContext ()	: void		

• W oknie przeglądarki (*Browser*) dodana zostanie gałąź diagramu komponentów oraz nowa gałąź diagramu klas. Kliknąć podwójnie gałąź diagramu komponentów w oknie przeglądarki. Diagram komponentów pojawi się w nowym oknie:



• Z górnego menu wybrać element *Tools/Display Preferences* i zaznaczyć opcję *Show Interfaces Symbols* w gałęzi *Component*. Kliknąć *Set As Default* a następnie *OK* – na diagramie wyświetlone zostaną symbole interfejsów naszego komponentu:



• Kliknąć podwójnie element *Class Diagram for Component Faktura*, znajdujący się w oknie przeglądarki (*Browser*) – w oknie pojawi się diagram klas jak na rysunku:



• Na diagramie widzimy symbol klasy *FakturaBean*, tzw. klasę klucza głównego (*Primary Key Class*) *FakturaPK* oraz interfejsy: zdalny, zdalny domowy, lokalny oraz lokalny domowy. Krótkie wyjaśnienie tych pojęć przedstawiono w poniższej tabeli:

Pojęcie	Opis
Zdalny interfejs domowy	Definiuje metody i operacje używane w perspektywie
(Remote home interface)	zdalnego klienta. Dziedziczy po interfejsie
	javax.ejb.EJBHome.
Zdalny interfejs	Dostarcza perspektywę zdalnego klienta.
(Remote interface)	
Lokalny interfejs domowy	Definiuje metody i operacje używane w perspektywie
(Local home interface)	klienta lokalnego. Dziedziczy po interfejsie
	javax.ejb.EJBLocal-Home.
Lokalny interfejs	Pozwala na bezpośredni dostęp do elementów klasy.
(Local interface)	
Klasa Bean	Klasa implementująca metody biznesowe bean.
(Bean Class)	
Klasa klucza głównego	Klasa realizująca wskaźnik do bazy danych.
(Primary Key Class)	

• Klasa *FakturaBean* jest połączona z pozostałymi elementami związkami zależności (*dependency*).

8.2. Dodawanie metod klasy Bean

- Kliknąć podwójnie symbol klasy *FakturaBean* w oknie diagramu, następnie kliknąć zakładkę *Operations* pojawi się lista operacji.
- Kliknąć przycisk *Add* i wybrać *EJB Create Method*. W ten sposób metoda *ejbCreate* została dodana do klasy *FakturaBean* oraz do lokalnego interfejsu domowego, zaś tylko do klasy *FakturaBean* dodana została metoda *ejbPostCreate*.
- Kliknąć *OK* w celu zamknięcia arkusza właściwości.
- Sprawdzić symbol interfejsu *FakturaLocalHome* na diagramie metoda *create* (*utwórz*) została dodana do interfejsu.

8.3. Przeglądanie klas oraz interfejsów EJB

Z arkusza właściwości komponentów możemy przeglądać klasy oraz interfejsy EJB.

- Kliknąć podwójnie gałąź diagramu komponentów w oknie przeglądarki (*Browser*) a następnie kliknąć podwójnie symbol komponentu *Faktura*.
- Kliknąć zakładkę *EJB* strona *EJB* wyszczególnia wszystkie klasy oraz interfejsy bieżącego obiektu.
- Kliknąć ikonę *Properties* (*właściwości*) z prawej strony listy rozwijalnej *Remote home interface* pojawi się arkusz właściwości tego interfejsu. W ten sposób możemy przeglądać oraz modyfikować właściwości każdego elementu *EJB*.
- Kliknąć *OK* w każdym z okien dialogowych, zapisać model na dysku.

9. Tworzenie diagramu wdrożeniowego (deployment diagram)

Diagramy wdrożeniowe, podobnie jak diagramy komponentów, należą do grupy diagramów implementacyjnych. Pokazują one konfigurację elementów czasu wykonania: komponentów sprzętowych, komponentów oprogramowania, procesów oraz związanych z nimi obiektów.

- Kliknąć prawym przyciskiem myszy w oknie dowolnego z diagramów i z menu kontekstowego wybrać opcję *Diagram/New Diagram/Deployment Diagram*.
- W polu *Name* arkusza właściwości diagramu wpisać *Wdrożeniowy*. Kliknąć *OK*.
- Kliknąć podwójnie gałąź diagramu komponentów w oknie przeglądarki (*Browser*) na diagramie komponentów znajduje się komponent *Faktura*.
- Kliknąć prawym przyciskiem myszy symbol komponentu, z menu kontekstowego wybrać *Deploy Component to Node*.
- Wybrać opcję *New Node*, kliknąć *OK*. Nowa gałąź, zawierająca nową instancję komponentu pojawiła się w oknie przeglądarki (*Browser*).
- Kliknąć podwójnie gałąź diagramu wdrożeniowego w oknie przeglądarki (*Browser*), następnie kliknąć prawym przyciskiem myszy na diagramie i wybrać z menu kontekstowego opcję *Diagram/Show symbols*. Zaznaczyć utworzoną przed chwilą gałąź, kliknąć *OK*.

• Na diagramie pojawi się symbol gałęzi (*Node*) zawierający wewnątrz instancję komponentu. Ta instancja jest instancją komponentu *EJB Entity Bean* o nazwie *Faktura*. Gałąź odpowiada serwerowi aplikacji, wewnątrz której wykonywane są instancje komponentu:



- Trzymając wciśnięty klawisz CTRL kliknąć podwójnie symbol gałęzi (*Node_1*) na diagramie.
- Przeciągnąć symbol utworzonej przed chwilą instancji komponentu z okna przeglądarki (*Browser Nodes / Node_1 / Component Instances / ComponentInstance_1*) do nowego okna diagramu.
- W diagramie pojawi się symbol instancji komponentu. Teraz można dodawać dane z arkusza właściwości instancji komponentu.
- Można także dodawać dowolne pliki, jak na przykład pliki tekstowe zawierające informację o procesie wdrożeniowym.
- Kliknąć prawym przyciskiem myszy, z menu kontekstowego wybrać opcję *Diagram/Go Up One Level* a następnie *Wdrożeniowy*, aby powrócić do okna macierzystego.
- Zapisać model na dysku.

UWAGA: Dla porównania i analizy można przejrzeć także podręcznikowy model *oomafter.oom* (w języku angielskim), znajdujący się w katalogu *Examples/Tutorial*.

Zachęcam także do zapoznania się z następującymi modelami przykładowymi z katalogu *Examples*:

- *DesginUseCaseTutorial.oom z*awiera diagram przypadków użycia oraz diagramy sekwencji modelujące... opisany w niniejszym podręczniku przykład tworzenia diagramu przypadków użycia.
- *demo.oom* zawiera model w formie diagramu klas, obejmujący urządzenia peryferyjne, przyłączane do portu równoległego komputera.
- *demo2.oom* zawiera modele wybranych aspektów użytkowania pakietu *Power Designer*. Są tutaj modele przypadków użycia oraz diagramy sekwencji dla: sprawdzania i przechowywania modelu oraz generacji i rewersowania bazy danych.

Krzysztof Bartecki