

Temat	UML. Diagram pakietów	Czas
9	Omówienie i praca na lekcji	2h
Wymaga	Diagram klas	

1. Omówienie diagramu
2. Zamodelowanie systemu
3. Praca samodzielna / zadanie domowe

Omówienie diagramu

Diagram pakietów reprezentuje widok konstrukcji systemu i jest potrzebny, gdy rozmiar tego systemu lub liczba jego klas jest na tyle duża, że bez sensownej organizacji tych klas praca nad systemem może być chaosem. Jeżeli system jest mały, to diagram pakietów może wskazywać na chęć stworzenia biblioteki lub modułu, który będzie częścią większego systemu.

Poprzednio poznane diagramy klas są podstawą do powstania diagramu pakietów. Dzięki pakietom możliwe jest wymuszenie struktury organizacji klas w logicznie powiązane ze sobą grupy. Z kolei pakiety można łączyć w większe pakiety. Większość języków programowania obiektowego implementuje ideę pakietów UML pozwalającą na organizację klas:

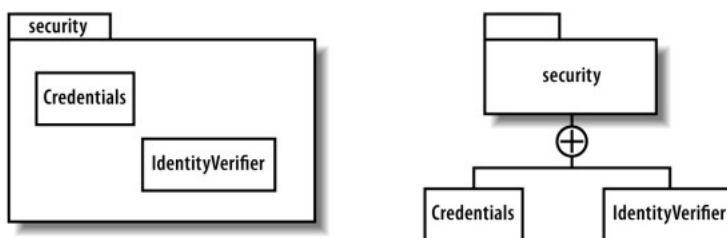
- *package* (pakiety) – w językach Java i Python,
- *namespace* (przestrzenie nazw) – w PHP i C#.

Elementami notacji graficznej diagramu są Pakiet, Klasa umieszczona w pakiecie oraz linie prezentujące zależności i połączenia pomiędzy pakietami. Możliwe przedstawienie pakietu:

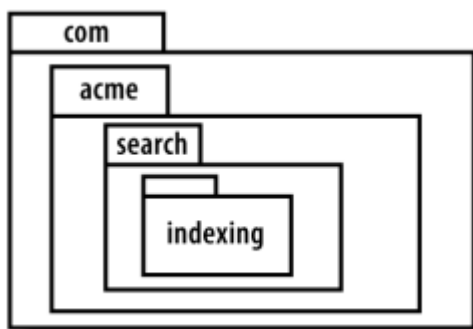


Możliwą zawartością pakietów są grupowane klasy (a) lub inne, mniejsze pakiety (b):

a) grupowanie klas (tworzenie pakietu)



b) grupowanie pakietów (tworzenie programu)



Przykład grupowania pakietów idealnie prezentuje podejście obiektowe języka Java, które zakłada, że każda klasa w programie od początku jest częścią jakiegoś pakietu, a pakiety można grupować w kolejne pakiety. Przykład użycia w Java:

```
import com.acme.search.*;
```

wskazuje, że chcemy użyć dowolnej klasy z pakietu Search, który znajduje się w com.acme. Lub zgodnie z konwencją firma-autor-projekt: **zse.gpetri.teleport**.

Przykłady z życia wzięte:

- a) pakiet narzędzi linuxowych **net-tools**, który zawiera takie programy jak: **ifconfig**, **arp**, **route** czy **netstat**,
- b) pakiet programów do przetwarzania systemu plików,
- c) pakiet programów do zarządzania plikami.

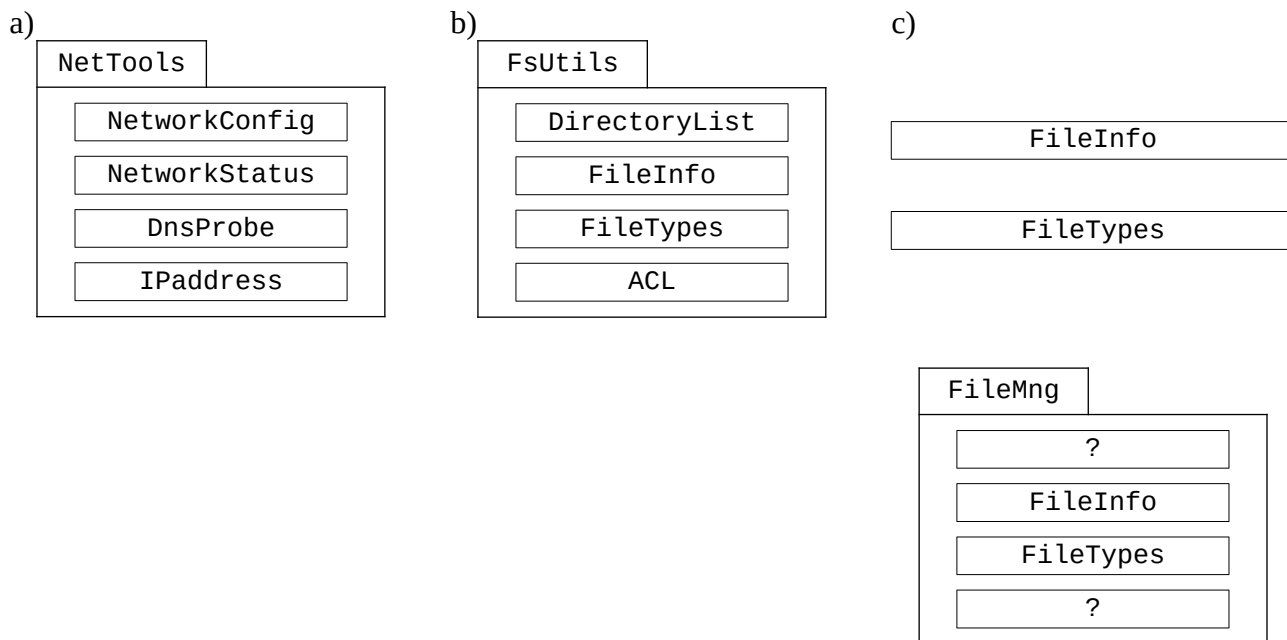
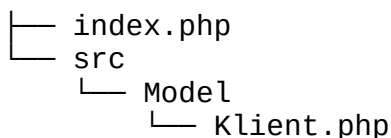


Diagram pakietów rozwiązuje następujące problemy:

- chaos klas w systemie - organizuje klasy w logiczne zbiory,
- powtarzające się nazwy klas – rozdziela klasy o tych samych nazwach do różnych pakietów,
- konflikty nazw klas i własności kodu w projektach z wieloma programistami – pozwala nazwać pakiety zgodnie z hierarchiczną strukturą organizacyjną.

Struktura plików i katalogów:



Składnia w języku PHP:

```
namespace Sklep\Model;

class Klient
{
    public function metoda() {}
}

index.php
require 'src/Model/Klient.php';
$kl = new Sklep\Model\Klient('Jan');
echo $customer->metoda();
```

Zadania

1. Utwórz projekt w IDEA IntelliJ w języku Java, którego nazwa to: ProjektPakietowy, a struktura **pakietów** i **klas** to:
com.tools.File, **com.tools.Dir**, **com.apps.Mng**, **com.apps.Edit**
2. Utwórz katalog na dysku, a w nim klasy w języku PHP takie jak w przykładzie powyżej.
3. Przygotuj projekt systemu w postaci katalogów i plików na systemie plików dysku
4. Pogrupuj klasy w pakiety wybranego systemu z dostarczonej listy.