

Klasy i komponenty

- Klasy, atrybuty, metody
- Diagramy klas
- Komponenty i interfejsy
- Diagramy komponentów

Języki formalne i kompilatory, © by Michał Śmiałek

Po co nam klasy i komponenty

Podział kodu kompilatora na logiczne jednostki.

Reprezentacja struktur danych występujących w kompilatorze.

Hermetyzacja danych – zapobieganie „makaronowi” w kodzie.

Grupowanie algorytmów i powiązanie ich ze strukturami danych.

Realizacja zasady abstrakcji – grupowanie pojęć i zamykanie je w większe skrzynki.

Reprezentacja pojęć w dziedzinie problemu (np. elementów kompilatora).

WAŻNE: tutaj klasy i komponenty są wizualną reprezentacją kodu i umożliwiają koncentrację na strukturze kodu, a nie na szczegółach kodu.

Języki formalne i kompilatory

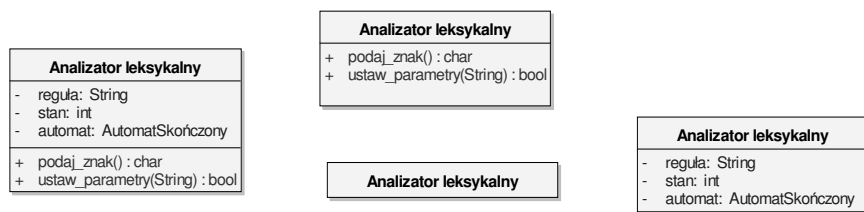
© by Michał Śmiałek

Klasa w języku UML

Klasa w UML reprezentowana jest za pomocą zwykłego prostokąta podzielonego na trzy części (rubryki).

Pierwsza rubryka zawiera nazwę klasy, druga - listę atrybutów, a trzecia - listę operacji. W zależności od stopnia szczegółowości diagramu, części z atrybutami i operacjami mogą być ukryte.

Klasa może występować na wielu diagramach klas. Na każdym diagramie mogą być pokazane tylko niektóre jej szczegóły.



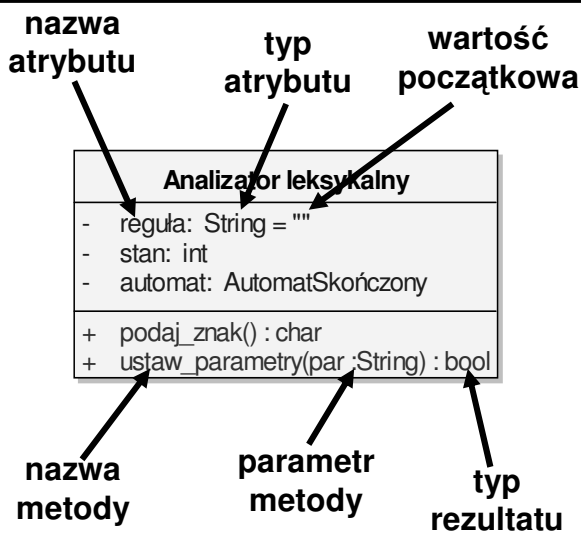
Języki formalne i kompilatory

© by Michał Śmiałek

Atrybuty i metody klasy

Atrybut to element (danych), który jest zawarty w ten sam sposób w każdym obiekcie klasy i jest reprezentowany w każdym obiekcie przez osobną wartość.

Atrybut opisywany jest co najmniej przez swoją nazwę. Dodatkowo, dla atrybutu można określić jego typ oraz wartość początkową.



Języki formalne i kompilatory

© by Michał Śmiałek

Ograniczanie dostępu do elementów klasy

Każdy element definicji klasy ma określony poziom zabezpieczenia (widoczność):

- zabroniony (private) - do tego elementu ma dostęp tylko sama klasa oraz klasy zaprzyjaźnione;
- ograniczony (protected) - do tego elementu ma dostęp tylko sama klasa oraz klasy pochodne;
- publiczny (public) - wszyscy klienci klasy mają dostęp do tego elementu klasy.

AnalizatorLeksykalny

```

- reguła: String = ""
- stan: int
- automat: AutomatSkończony

+ podaj_znak() : char
# ustaw_parametry(par :String) : bool
                    
```

Poziom dostępu do klasy określony jest przy pomocy odpowiedniego symbolu.

Języki formalne i kompilatory © by Michał Śmiałek

Związki klas

Związki klas pokazują powiązania między obiektami klas (patrz np. atrybuty typu innej klasy w Javie).

Związek oznacza, że w trakcie działania systemu (runtime) istnieje wymiana komunikatów między obiektami klas.

Powiązanie między obiektami klas może trwać przez cały czas ich życia, lub tylko przez pewien okres.

Związek oznaczany jest linią uzupełnioną o: nazwę związku, rolę klas, nawigowalność, krotność i uporządkowanie.

```

classDiagram
    class AnalizatorLeksykalny {
        +źródło_symboli
    }
    class AnalizatorSkładniowy {
        +pobieracz_symboli
    }
    AnalizatorLeksykalny "1" -- "1" AnalizatorSkładniowy : Pobieranie
                    
```

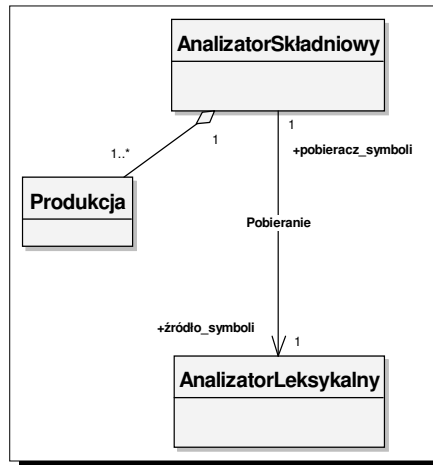
Języki formalne i kompilatory © by Michał Śmiałek

Rodzaje i opis związków klas

Ukierunkowanie (nawigowalność) związku oznacza, że z obiektów jednej z klas można bezpośrednio osiągnąć obiekty drugiej.

Krotność określa liczbę obiektów, jakie mogą występować w danym związku. Krotność podaje się jako zakres lub pojedynczą wartość, '**' oznacza 'wiele'.

Relacje agregacji oznaczają, że obiekty jednej klasy są w całości zawarte w obiektach innej klasy.



Języki formalne i kompilatory

© by Michał Śmiałek

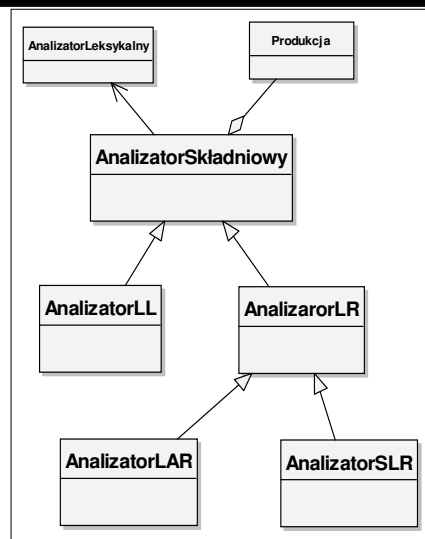
Dziedziczenie

Jedna klasa może dziedziczyć od innej jej cechy (atrybuty i metody).

Klasa dziedzicząca może dodać własne cechy lub zmienić sposób działania istniejących usług.

Klasa dziedzicząca musi określić jedynie to czym się różni od swego „przodka”.

Klasy dziedziczą również związki z innymi klasami!



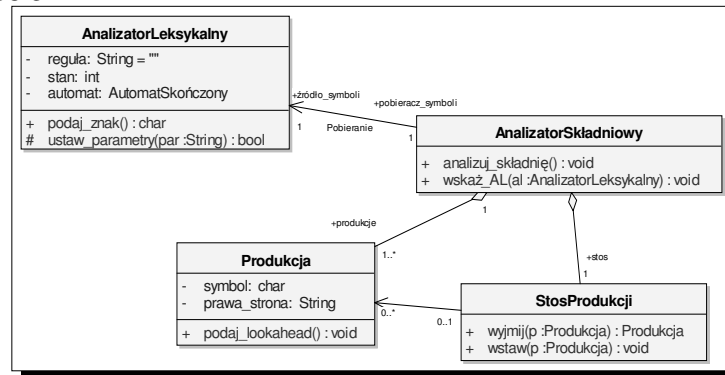
Języki formalne i kompilatory

© by Michał Śmiałek

Diagramy klas

Powiązania między wieloma klasami uwidaczniamy na diagramie klas, który określa statyczną strukturę systemu.

Diagram klas może zawierać tylko część klas - np. klasy w jednym pakiecie.

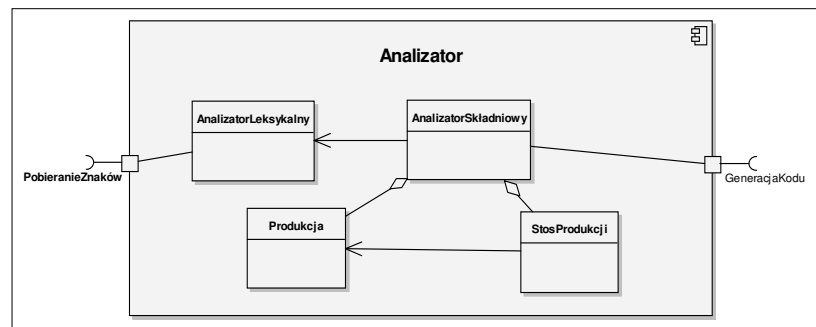


Języki formalne i kompilatory

© by Michał Śmiałek

Komponenty

Komponent to element systemu realizujący określony fragment jego zadań, komunikujący się z innymi elementami za pośrednictwem interfejsów. Składniki komponentu mogą być opisane za pomocą diagramów klas.



Języki formalne i kompilatory

© by Michał Śmiałek

