

Wprowadzenie do kompilatorów

- Czy ja kiedykolwiek napiszę jakiś kompilator?
- Jakie zadania ma do wykonania kompilator?
- Czy „język formalny” to rodzaj języka programowania?
- Co to jest UML?

Języki formalne i kompilatory, © by Michał Śmiałek

Czy ja kiedykolwiek napiszę jakiś kompilator?

PRAWDOPODOBNIENIE NIE!

Ale...

Duża część oprogramowania wykorzystuje wiele elementów teorii kompilacji, np.:

- gramatyki regularne używane są do wyszukiwania tekstu;
- automaty skończone są powszechnie wykorzystywane podczas projektowania oprogramowania;
- języki formalne mogą być użyte do opisu danych wprowadzanych przez użytkownika.

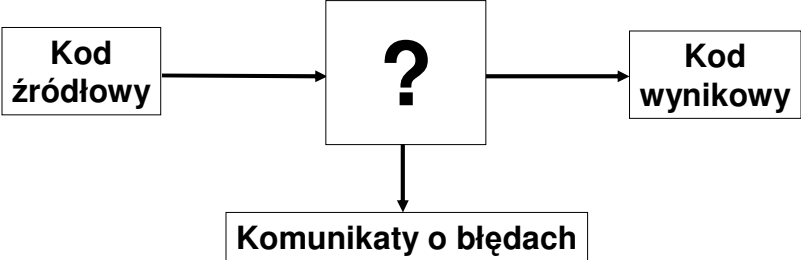
Kompilator jest dobrym przykładem dużego systemu oprogramowania.

Techniki stosowane do konstrukcji kompilatorów stosowane są też podczas konstrukcji innych systemów.

Języki formalne i kompilatory

© by Michał Śmiałek

Jakie zadania ma do wykonania kompilator?



Kompilator przetwarza kod zapisany w języku źródłowym na kod zapisany w języku wynikowym zgłaszając odpowiednie komunikaty.

Dla użytkownika kompilator stanowi „czarną skrzynkę” działającą w sposób automatyczny. Niniejszy wykład ma za zadanie pokazać, jak ta skrzynka wygląda od środka (jak zbudować taką skrzynkę).

Języki formalne i kompilatory © by Michał Śmiałek

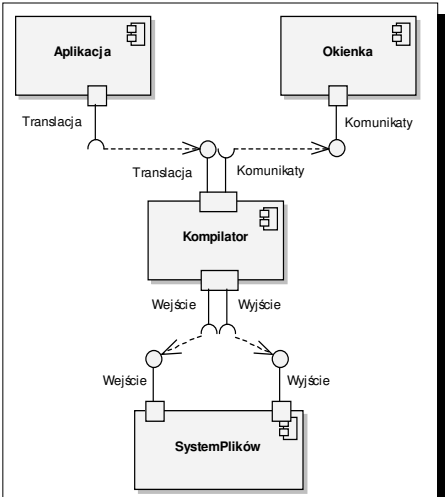
Z kim komunikuje się kompilator?

Kompilator przyjmuje od jakiejś Aplikacji komunikaty wykonania kompilacji (Translacja).

Kompilator wysyła do Okienek polecenia pokazania komunikatu (Komunikaty).

Kompilator wysyła do SystemuPlików polecenia pobrania z pliku kodu źródłowego (Wejście).

Kompilator wysyła do SystemuPlików polecenia zapisania do pliku kodu wynikowego (Wyjście).

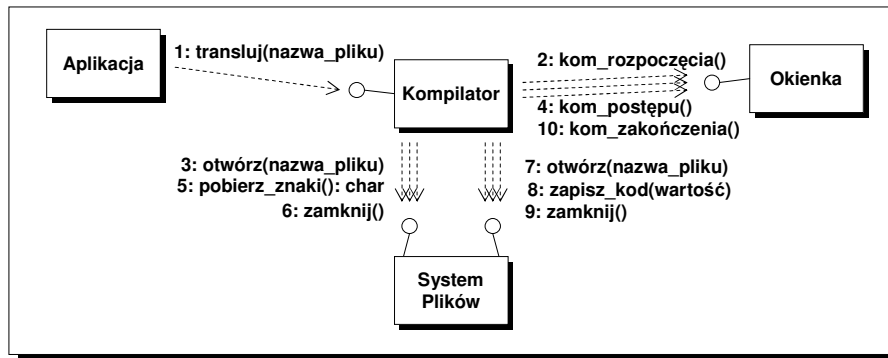


Języki formalne i kompilatory © by Michał Śmiałek

Jak się zachowuje kompilator?

Kompilator dostaje od Aplikacji komunikat z żądaniem dokonania translacji.

Kompilator wykonuje translację współpracując z Systemem Plików i Okienkami.

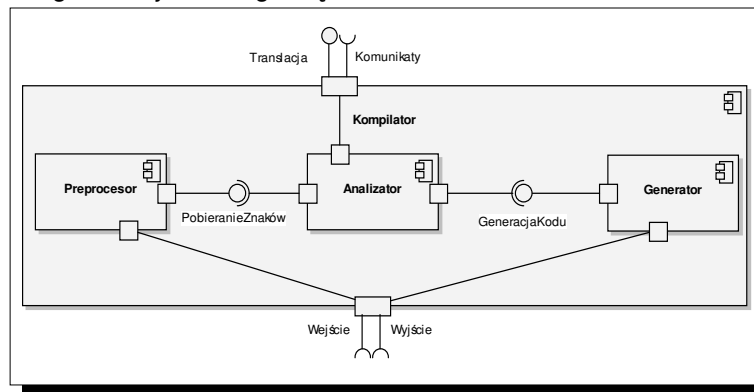


Języki formalne i kompilatory

© by Michał Śmiałek

Z czego się składa kompilator?

- Preprocesor służy do wstępnego przygotowania tekstu źródłowego.
- Analizator służy do analizy kodu źródłowego.
- Generator służy do generacji kodu wynikowego.
- Czego brakuje: obsługa błędów.

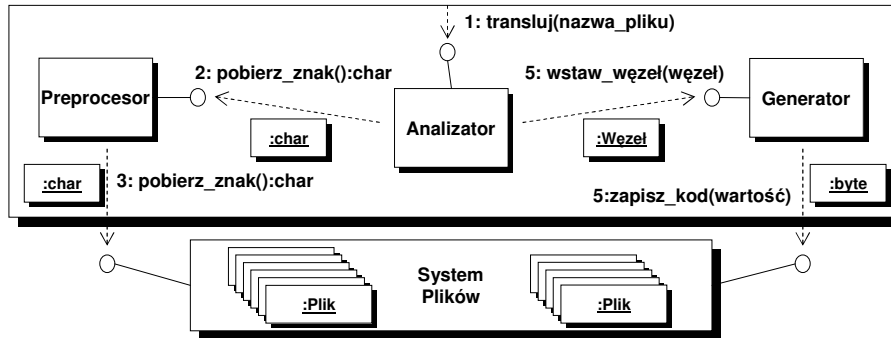


Języki formalne i kompilatory

© by Michał Śmiałek

Jak działa kompilator? (sterowanie)

- Analizator zarządza kompilacją dokonując analizy kodu źródłowego.
- Tekst źródłowy Analizator pobiera z Preprocesora, który go wstępnie przygotowuje.
- Dane do generacji kodu przekazywane są Generatorowi, który na podstawie tych danych generuje kod.



Języki formalne i kompilatory

© by Michał Śmiałek

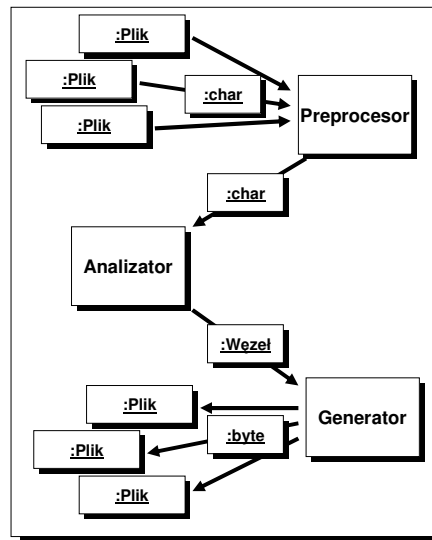
Jak działa kompilator? (przepływ danych)

Preprocesor odczytuje znaki z różnych plików i przekształca w jeden ciągły strumień znaków.

Analizator odczytuje znaki ze strumienia znaków i sprawdza zgodność z językiem. Analizator w odpowiednich momentach podczas analizy tworzy węzły dla generatora.

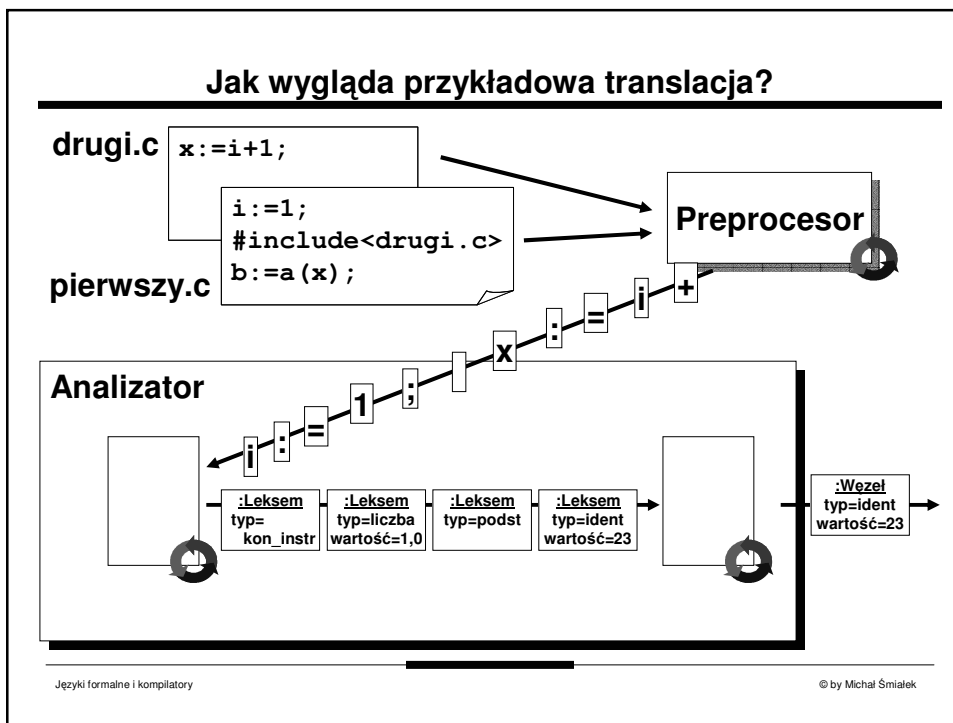
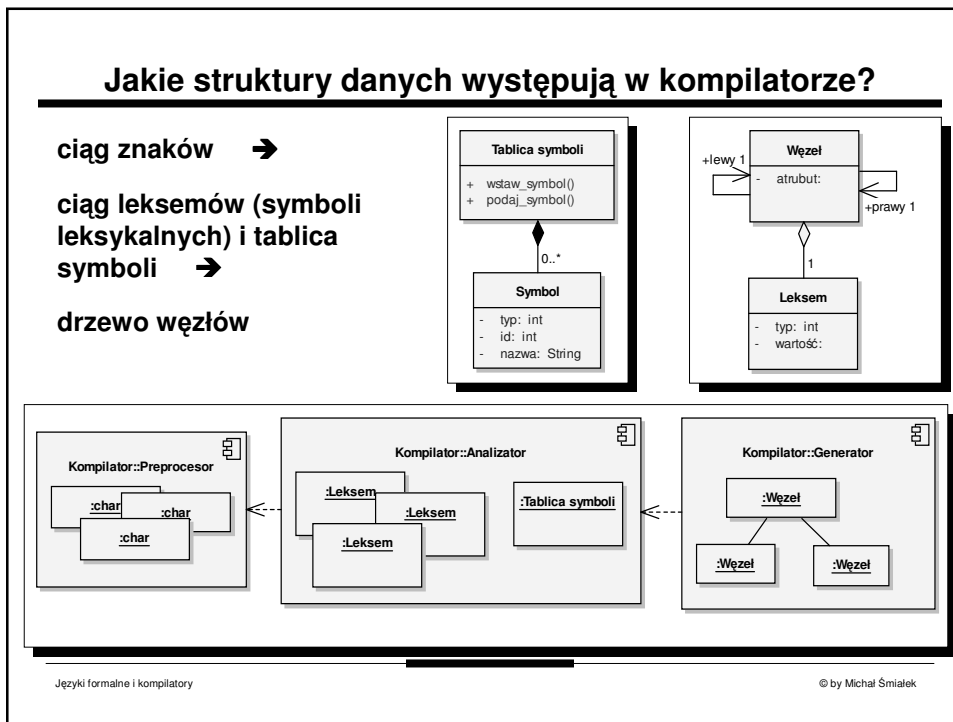
Generator otrzymuje węzły i wstawia je w odpowiednie struktury danych.

Po zbudowaniu odpowiednich struktur, generator tworzy ciąg wartości i wstawia je do plików.



Języki formalne i kompilatory

© by Michał Śmiałek



Czy język formalny to taki rodzaj języka programowania?

OCZYWIŚCIE, NIE! Język formalny (gramatyka formalna) to sposób zapisu struktury programu (składni języka źródłowego) używany m.in. podczas konstrukcji analizatorów.

Języki formalne i kompilatory © by Michał Śmiałek

Jak projektować kompilatory?

Diagramy z poprzednich slajdów pokazują strukturę kompilatora w formie współpracujących ze sobą komponentów.

Każdy komponent może stanowić klasę w rozumieniu obiektowym.

Kompilator jest jednym z przykładów aplikacji, które można zbudować (zanalizować i zaprojektować) w sposób obiektowy.

Projektowanie kompilatora powinno odbywać się w sposób graficzny.

Graficznym „językiem” do analizy i projektowania systemów jest UML.

UML jest w tej chwili standardem modelowania obiektowego.

Języki formalne i kompilatory © by Michał Śmiałek

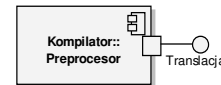
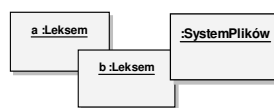
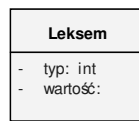
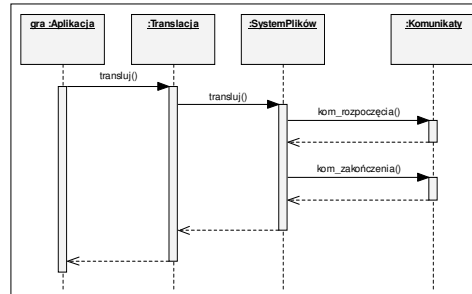
Jak wygląda język UML?

Podstawowym elementem jest klasa.

Komponenty to też klasy.

Komponenty i klasy mogą posiadać interfejsy.

Obiekty klas występują na diagramach pokazujących dynamikę.



Języki formalne i kompilatory

© by Michał Śmiałek

Co dalej?

Jak zaprojektować kompilator?

- Diagramy klas – podstawowe składniki programu kompilatora
- Diagramy sekwencji – opis dynamiki działania kompilatora

Jak działa preprocesor?

Jak działa analizator?

- Gramatyki regularne – opis struktury pojedynczych symboli
- Analizator leksykalny – komponent łączący znaki w symbole
- Gramatyki bezkontekstowe – opis struktury języka
- Analizator składniowy – komponent sprawdzający poprawność kodu
- Analizatory LL – analiza zstępująca
- Analizatory LR – analiza wstępująca

Jak generować kod?

Języki formalne i kompilatory

© by Michał Śmiałek