

PROGRAMSKI JEZICI

Opis (definicija programskih jezika)

- Komponente u opisu programskog jezika
- Opis leksike pomoću regularnih izraza
- Opis sintakse pomoću beskonteksnih gramatika

Definicija programskog jezika

- **Azbuka**

- Neprazan skup simbola od kojih se prave reči jezika

- **Leksika** – nauka o rečima

- U programskim jezicima definiše vrste reči koje se u programskom jeziku koriste i kako se one grade
- Definiše se formalno pomoću regularnih izraza

- **Sintaksa** – nauka o rečenicama

- U programskim jezicima definiše strukturu programa
- Definiše se formalno pomoću beskonteksnih gramatika

- **Semantika** – nauka o značenju rečenica

- U programskim jezicima definiše veze između strukturnih elemenata koje sintaksa opisuje
- Definiše se neformalno pomoću govornog jezika

```
int a;  
bool b;  
b=a;
```

Azbuka programskih jezika

- Velika i mala slova engleske azbuke
- Dekadne cifre
- Specijalni znaci

Leksika programskih jezika

- Definiše vrste reči
- Za svaku vrstu reči određuje njene moguće oblike i značenje u programu (**TOKEN**)
- Mogući oblici (zapisi) izvesne vrste reči se definišu regularnim izrazima

Vrste reči u višim programskim jezicima

- **Ključne reči**
 - Svaka ima svoj jedinstveni zapis i jedinstveno sintaksno i semantičko značenje (jedinstveni token)
- **Simbolička imena (identifikatori)**
 - Služe za imenovanje različitih entiteta (simboličkih konstanti, promenljivih, funkcija, tipova,...) koje programeri definišu u kodu
- **Konstante (literali)**
 - Vrednosti različitih tipova - definisane za svaki elementarni tip podataka
- **Operatori**
 - Specijalni znaci ili kombinacije specijalnih znaka koje služe za formiranje izraza
- **Separatori**
 - Specijalni znaci koji razdvajaju logičke celine u kodu, tj. različite sintaksne elemente
- **Beli simboli i komentari**
 - Delovi koda koje prevodilac ignoriše, nemaju nikakvo sintaksno ni semantičko značenje – nisu tokeni, koriste se da obezbede čitljivost i razumevanje koda

Regularni izrazi

- Definišu moguće oblike (zapise) određene vrste reči
- Sastavljeni su od:
 - simbola koji čine azbuku programskog jezika i
 - metasimbola – simbola sa specijalnim značenjem u regularnom izrazu

Metasimboli u regularnim izrazima

- **()** – koriste se za kreiranje grupe
- **[]** – alternativa (definišu izbor jenog iz skupa simbola navedenih unutar zagrada)
- **-** - koristi se za definisanje intervala simbola unutar alternative
- **+** - označava da se prethodni simbol ili grupa može pojaviti jednom ili više puta u zapisu
- ***** - označava da se prethodni simbol ili grupa može pojaviti ni jednom ili više puta u zapisu
- **?** – označava da se prethodni simbol ili grupa može, ali ne mora pojaviti u zapisu
- **.** – zamenjuje bilo koji simbol azbuke
- **~** - zamenjuje bilo koji niz karaktera do znaka ili grupe koja sledi
- **|** - izbor jedne od 2 alternative
- **** - poništava specijalno značenje metakaraktera koji sledi (takodje, unutar alternative ili izmedju dvostrukih navodnika metakarakter i nemaju specijalno značenje)

Leksički elementi programskog jezika java

- Ključne reči:**

abstract	default	if	private	throws
boolean	do	implements	protected	transient
break	double	import	public	try
byte	else	instanceof	returnshort	void
case	extends	int	static	volatile
catch	final	interface	super	while
char	finally	long	switch	
class	float	native	synchronized	
const	for	new	this	
continue	goto	package	throw	

Leksički elementi programskog jezika java

- **Simbolička imena** – niz karaktera sastavljen od velikih i malih slova, cifara i simbola '_' i '\$' u kojem prvi znak ne može biti cifra.

`[A-Za-z_$][A-Za-z0-1_$]*`

Leksički elementi programskog jezika java

- **Konstante ili literali:**

- Celobrojne

- Dekadne: Označeni ili neoznačeni niz dekadnih cifara koji počinje cifrom različitom od nule ili nula.

`([+-]?[1-9][0-9]*)|0`

- Oktalne: Označeni ili neoznačeni niz oktalnih cifara koji počinje cifrom 0.

`[+-]?0[0-7]+`

- Heksadekadne: Označeni ili neoznačeni niz heksadekadnih cifara koji počinje prefiksom 0x.

`[+-]?0[xX][0-9A-Fa-f]+`

- Binarne: Označeni ili neoznačeni niz binarnih cifara koji počinje prefiksom 0b.

`[+-]?0[bB][01]+`

Leksički elementi programskog jezika java

- **Konstante ili literali:**

- Realne (tipa *double*): Označeni ili neoznačeni niz dekadnih cifara koji sadrži decimalnu tačku i/ili eksponent.

`([+-]?[0-9]+\.[0-9]*([Ee][+-]?[0-9]+)?)|([+-]?[0-9]+[Ee][+-]?[0-9]+)`

- Znakovne (tipa *char*): Bilo koji karkter napisan između apostrofa.

`'.'`

- Logičke (tipa *boolean*): true ili false.

`(true)|(false)`

- Konstantni znakovni nizovi (tipa *string*): Niz karktera napisan između dvostrukih navoda.

`"~\"`

Leksički elementi programskog jezika java

- **Operatori** po prioritetu:

postfix	<code>expr++ expr--</code>
unary	<code>++expr --expr +expr -expr ~ !</code>
multiplicative	<code>* / %</code>
additive	<code>+ -</code>
shift	<code><< >> >>></code>
relational	<code>< > <= >= instanceof</code>
equality	<code>== !=</code>
bitwise AND	<code>&</code>
bitwise exclusive OR	<code>^</code>
bitwise inclusive OR	<code> </code>
logical AND	<code>&&</code>
logical OR	<code> </code>
ternary	<code>? :</code>
assignment	<code>= += -= *= /= %= &= ^= = <<= >>= >>>=</code>

Leksički elementi programskog jezika java

- **Separatori:**

()	Ograničavaju paremetre u definiciji ili pozivu metoda; regulišu redosled izvršenja operacija u izrazima; ograđuju tipove u izrazima za konverziju tipova; ograničavaju uslovne izraze u naredbama za kontrolu toka
{ }	Omeđuju blokove i vrednosti za inicijalizaciju polja
[]	Deklarišu tip polja i izdvajaju vrednosti iz polja
;	Definiše kraj naredbe
,	Razdvajaju identifikatore u deklaraciji promenljivih, povezuju izraze npr. u zaglavlju petlje...
.	Selektuju atribut ili metodu iz nekog objekta, selektuju podpaket ili klasu iz paketa...
:	Označava kraj labele

Leksički elementi programskog jezika java

- **Komentari:**

- Počinju simbolima // i završavaju se prelaskom na novi red

//~\n

- Počinju simbolima /* i završavaju se simbolima */

/*~(\/)

- Specijalna vrsta komentara koje analizira alat za kreiranje tehničke dokumentadije (JavaDoc), počinju simbolima /** i završavaju se simbolima */

/**~(\/)

Definicija sintakse – Beskonteksne gramatike

- **Beskonteksne gramatike** predstavljaju uređenu četvorku čiji si elementi:
 - **Terminali (terminalni simboli)** – Reči jezika, odnosno njihovi tokeni.
 - Primer: **if, then, else, ID, CONST, + ...**
 - **Neterminali (neterminalni simboli)** – Pomoćne sintaksne promenljive kojima se označavaju skupovi reči. Neterminali se uvode da bi se lakše definisale ispravne jezičke konstrukcije, kao i da bi se lakše definisala hierahijska programskog jezika.
 - Pimer: *Expression, Statement, IfStatement, WhileStatement...*
 - **Startni simbol** – Neterminal iz kojeg se izvode ispravne jezičke konstrukcije.
 - **Produkciono pravilo** – definiše način na koji se kombinuju terminalni i neterminalni simboli da bi se kreirale složenije jezičke konstrukcije. U opštem slučaju pravila su oblika:

$$X \rightarrow Y$$

gde je X neterminalni simbolm a Y niz sastavljen od terminalnih i neterminalnih simbola.

- Primer: *WhileStatement* \rightarrow **while** (*Expression*) *Statement*

Beskonteksna gramatika koja definiše aritmetičke izraze

- Terminali (terminalni simboli):

$\{ \text{CONST, ID, +, -, *, /, (,)} \}$

- Neterminali (neterminalni simboli):

$\{ \textit{Expression, Operator} \}$

- Startni simbol:

Expression

- Produkciona pravila

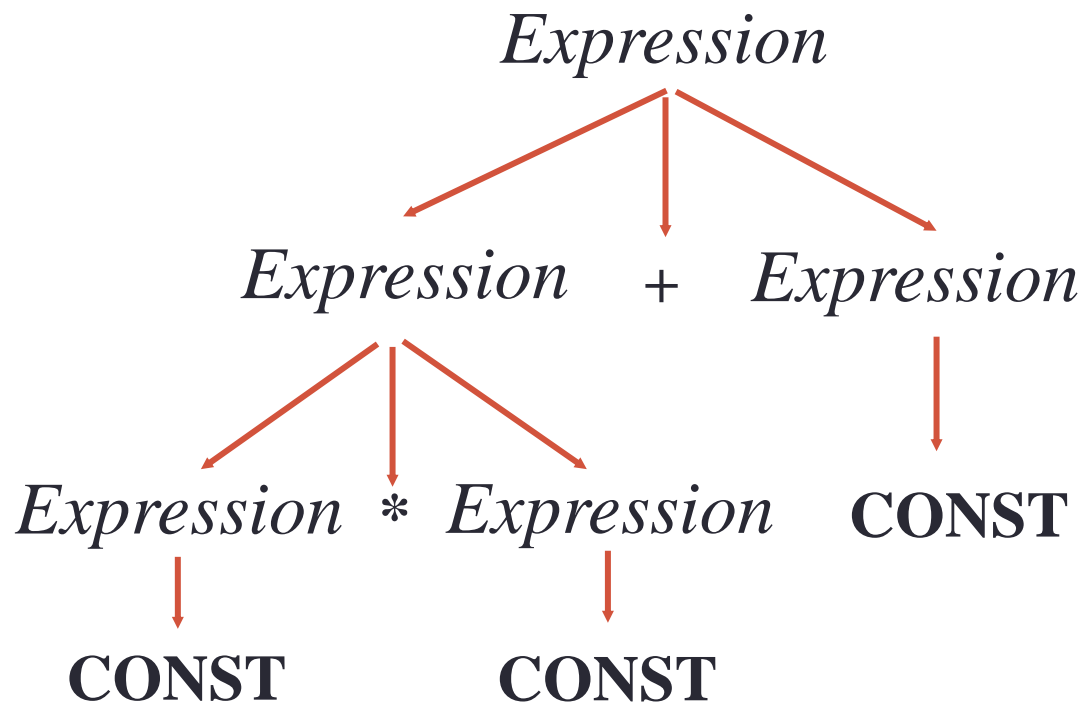
$\textit{Expression} \rightarrow \textit{Expression Operator Expression} \mid - \textit{Expression} \mid \text{CONST}$

$\textit{Operator} \rightarrow + \mid - \mid * \mid /$

Sintaksno stablo

- Prikazuje kako je sagrađena neka jezička konstrukcija.

Primer: **5*3+7**



Načini predstavljanja beskonteksnih gramatika

- Bekusova normalna forma
- BS notacija
- Sintaksni dijagrami

Bekusova normalna forma

- Prvi put upotrebljena za definisanje jezika ALGOL 58
- BNF notacija se sastoji od konačnog broja metalingvističkih formula (produkcioni pravila) koje sadže:
 - Metalingvističke konstante (terminalne simbole),
 - Metalingvističke promenljive (neterminalne simbole) koje imenuju određene sintaksne celine u programu i pišu se između simbola $< i >$.
 - Metasimboli
 - $::=$ - „po definiciji je“
 - $|$ - izbor jedne od više alternativa
- Pravila su oblika:

MetalingvistickaPromenljiva ::= NizMetalingvistickihKonstantiIPromenljivih

Definicija aritmetičkih izraza pomoću BNF

$$\langle \text{expression} \rangle ::= \langle \text{expression} \rangle \langle \text{operator} \rangle \langle \text{expression} \rangle$$
$$| - \langle \text{expression} \rangle | \mathbf{CONST}$$
$$\langle \text{operator} \rangle ::= + \mid - \mid * \mid /$$

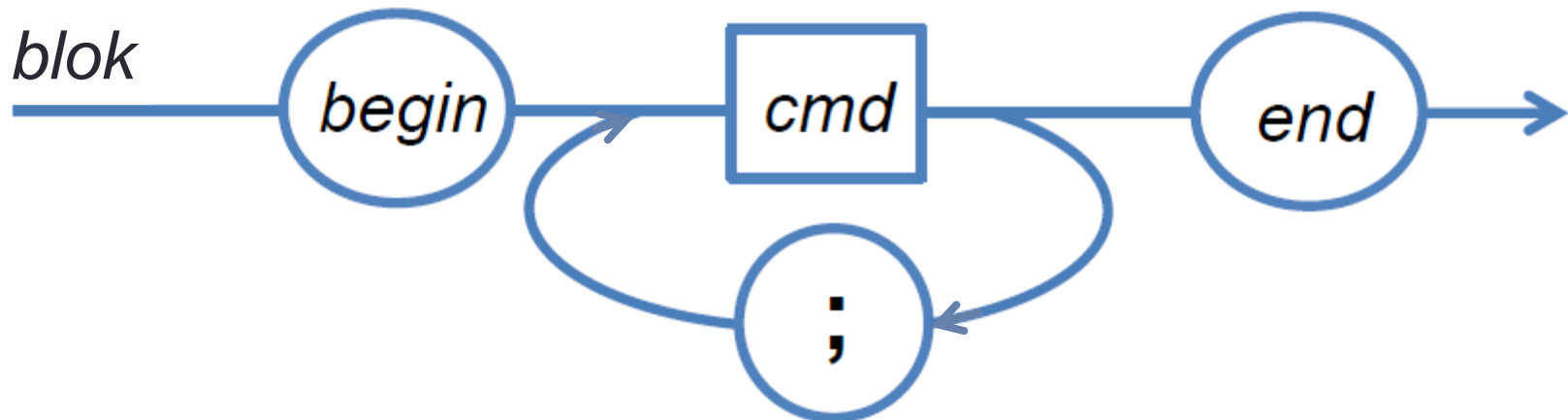
BS notacija

- Zasniva se na Wirtovoj modifikovanoj Bekusovoj normalnoj formi.
- Uvodi veći broj metasimbola:
 - `=` - zamenjuje simbol `::=`
 - `{ }` - označavaju višestruko ponavljanje skupa simbola unutar njih (uključujući i nijednom);
 - `[]` - označavaju opcione elemente;
 - `" "` – označavaju terminalne simbole;
 - `_` - povezuje veći broj reči u jedno ime metalingvističke promenljive;
 - `(*)` – ograničavaju komentar u opisu sintakse programskih jezika;
- Primer pravila u BS notaciji:
variable_declaration = type "id" ["=" "const"] { ",", "id" ["=" "const"] } " ;"

Sintaksni dijagrami

- Orjentisani grafovi koji opisuju sintaksu programskog jezika
- Svakom produkcionom pravilu odgovara po jedan sintaksni dijagram
- Na ulaznom potezu grafa nalazi se neterminalni simbol koji se definiše (simbol sa leve strane produkcionog pravila)
- Graf sadrži dve vrste čvorova:
 - Terminalne – označene krugovima (ili elipsama) koji odgovaraju terminalnim simbolima gramatika
 - Neterminalne – označene pravougaonicima koji odgovaraju neterminalnim simbolima gramatike
- Grane u grafu definišu redosled pojavljivanja simbola u pravilu
- Često korišćeni u opisu programskog jezika Pascal

Primer sintaksnog dijagrama



Semantička pravila programskih jezika

- Promenljive koje učestvuju u izrazu moraju biti prethodno deklarisanе (kod jezika sa statičkom tipizacijom),
- Promenljiva čija se vrednost koristi u izrazu mora biti prethodno inicijalizovana.
- Za svaku operaciju je definisan tip podataka nad kojim se može primenjivati.
- Lista stvarnih parametara u pozivu funkcije mora biti usaglašena sa listom fiktivnih parametara u definiciji funkcije (po broju i tipovima),
- Van klase se mogu koristiti samo članovi sa javnim pravom pristupa.
- ...