Објектно орјентисано програмирање

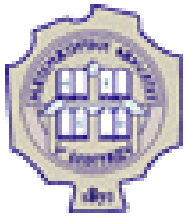
Владимир Филиповић
vladaf@matf.bg.ac.rs
Александар Картељ
kartelj@matf.bg.ac.rs

Елементарне конструкције у Јави

Владимир Филиповић
vladaf@matf.bg.ac.rs
Александар Картељ
kartelj@matf.bg.ac.rs

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs



Бекусова нотација

- Синтакса програмског језика се описује помоћу коначног скупа металингвистичких формула (МЛФ).
- МЛФ се састоји из леве и десне стране раздвојене универзалним мета-симболом ::=
- МЛФ је облика:

  α ::= γ

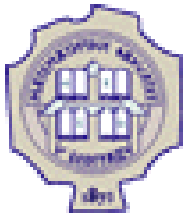
где је:

α металингвистичка променљива;

γ металингвистички израз.

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs

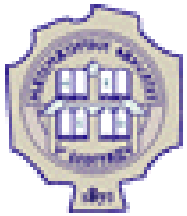


Бекусова нотација (2)

- Металингвистичка променљива:
  - фраза природног језика ограђена угластим (стреличастим) заградама (<, >)
- Металингвистичка константа:
  - знак или кључна реч објект-језика
- Универзални метасимбол ::= чита се 'по дефиницији је'
- Универзални метасимбол | чита се 'или'.
- Металингвистички израз може бити:
  - металингвистичка променљива,
  - металингвистичка константа
  - или коначан низ металингвистичких променљивих или металингвистичких константи раздвојених универзалним метасимболом | или надовезаних једни на друге.

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs



Бекусова нотација (3)

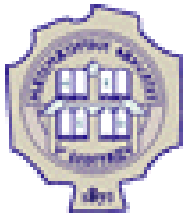
Пример

- Цео декадни број се у програмским језицима записује на исти начин као и у математици.
- Примери коректно записаних целих декадних бројева су:
  3663 -1234 +55252 19 -234 0 833
- Цео број се помоћу Бекусове нотације може дефинисати на следећи начин:

```
<ne nula cifra>                 ::= 1|2|3|4|5|6|7|8|9
<cifra>                         ::= 0|<ne nula cifra>
<znak broja>                    ::= +|-
<dekadni ceo broj bez znaka>    ::= <ne nula cifra>
                                |<dekadni ceo broj bez znaka><cifra>
<dekadni ceo broj>              ::= 0
                                |<dekadni ceo broj bez znaka>
                                |<znak broja><dekadni ceo broj bez znaka>
```

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs

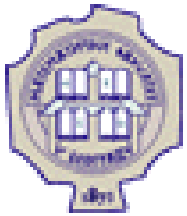


Бекусова нотација (4)

- Постоје разне модификације изворне Бекусове нотације увођењем нових метасимбола:
- лева и десна велика заграда
  - означава понављање обухваћене конструкције нула, једном или више пута;
- лева и десна средња заграда
  - означава опционо понављање обухваћене конструкције и
- лева и десна мала заграда
  - означава груписање конструкција.

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs



Бекусова нотација (5)

Пример

Први начин:

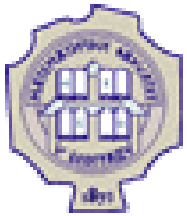
```
<ne nula cifra>                  ::= 1|2|3|4|5|6|7|8|9
<cifra>                          ::= 0|<ne nula cifra>
<znak broja>                     ::= +|-
<dekadni ceo broj bez znaka>     ::= <ne nula cifra>{<cifra>}
<dekadni ceo broj>               ::= 0
                                 | [<znak broja>]<dekadni ceo broj bez znaka>
```

Други начин:

```
<ne nula cifra>          ::= 1|2|3|4|5|6|7|8|9
<cifra>                  ::= 0|<ne nula cifra>
<dekadni ceo broj>       ::= 0| [(+|-)]<ne nula cifra>{<cifra>}
```

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs



Синтаксни дијаграм

- Граф са једним улазом и једним излазом.

- Представља металингвистички израз:

1. Чворови синтаксног дијаграма су металингвистичке константе и променљиве;
2. Елипсе представљају константе;
3. Правоугаоници променљиве које се дефинишу помоћу посебног синтаксног дијаграма.
4. Гране или потези дефинишу везу између чворова, тј. структуру синтаксног дијаграма.

Математички факултет

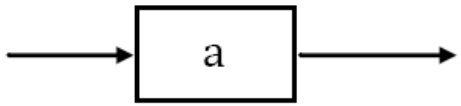
{vladaf,kartelj}@matf.bg.ac.rs

Синтаксни дијаграм (2)

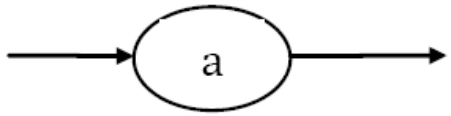
Бекусова нотација

Синтаксни дијаграм

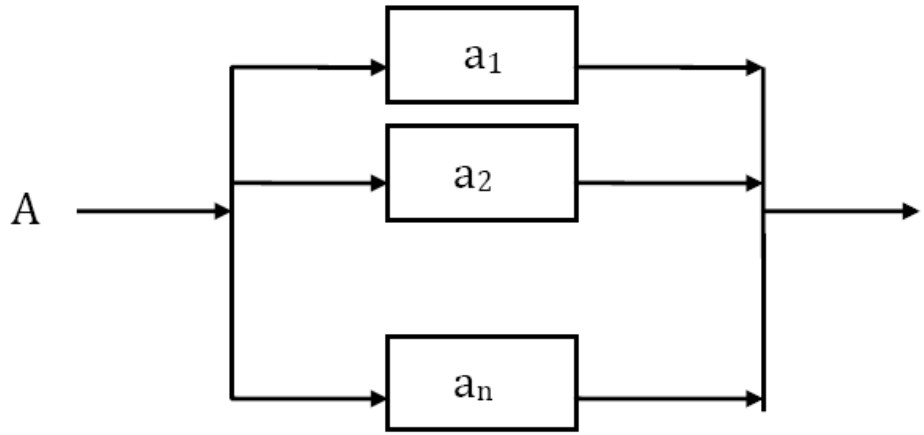
1. Металингвисичка променљива <a>



2. Металингвистичка константа a



3. A ::= a_1 | a_2 | … | a_n

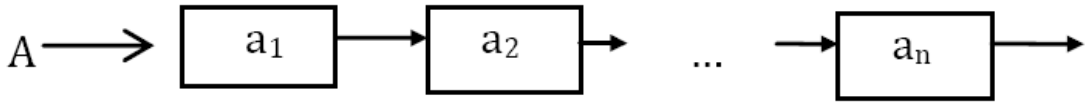


Математички факултет

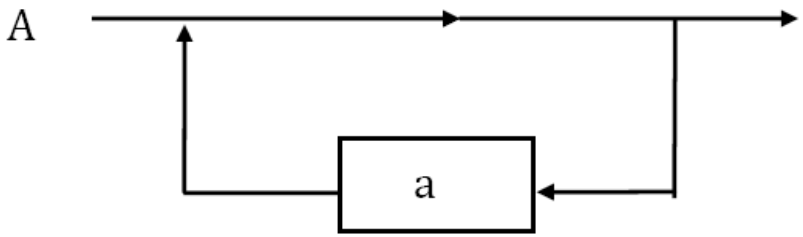
{vladaf,kartelj}@matf.bg.ac.rs

Синтаксни дијаграм (3)

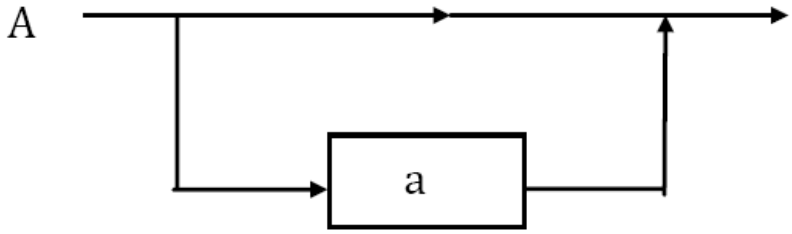
4. A ::= a_1 a_2 … a_n



5. A ::= {a}



6. A ::= [a]

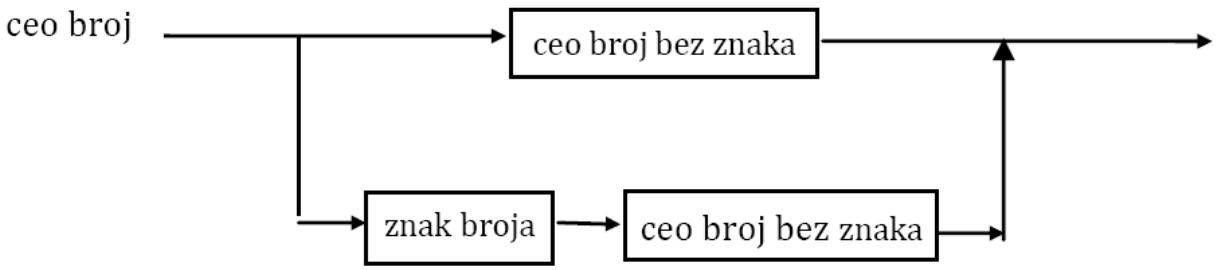


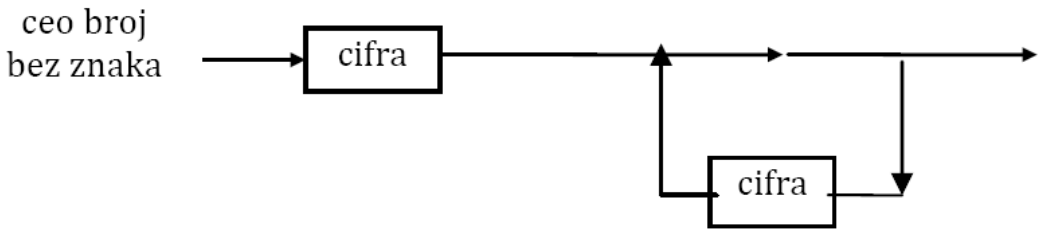
Математички факултет

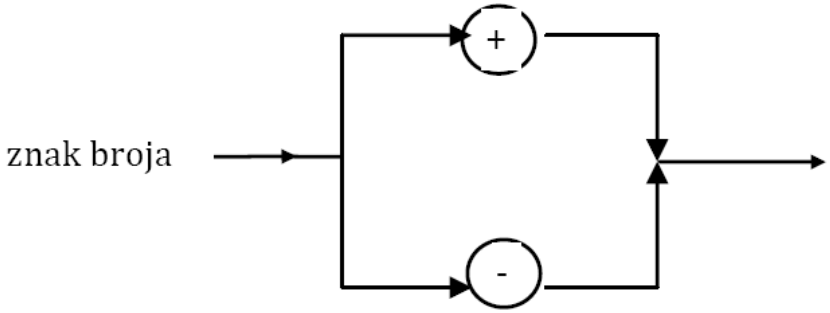
{vladaf,kartelj}@matf.bg.ac.rs

Синтаксни дијаграм (4)

Пример





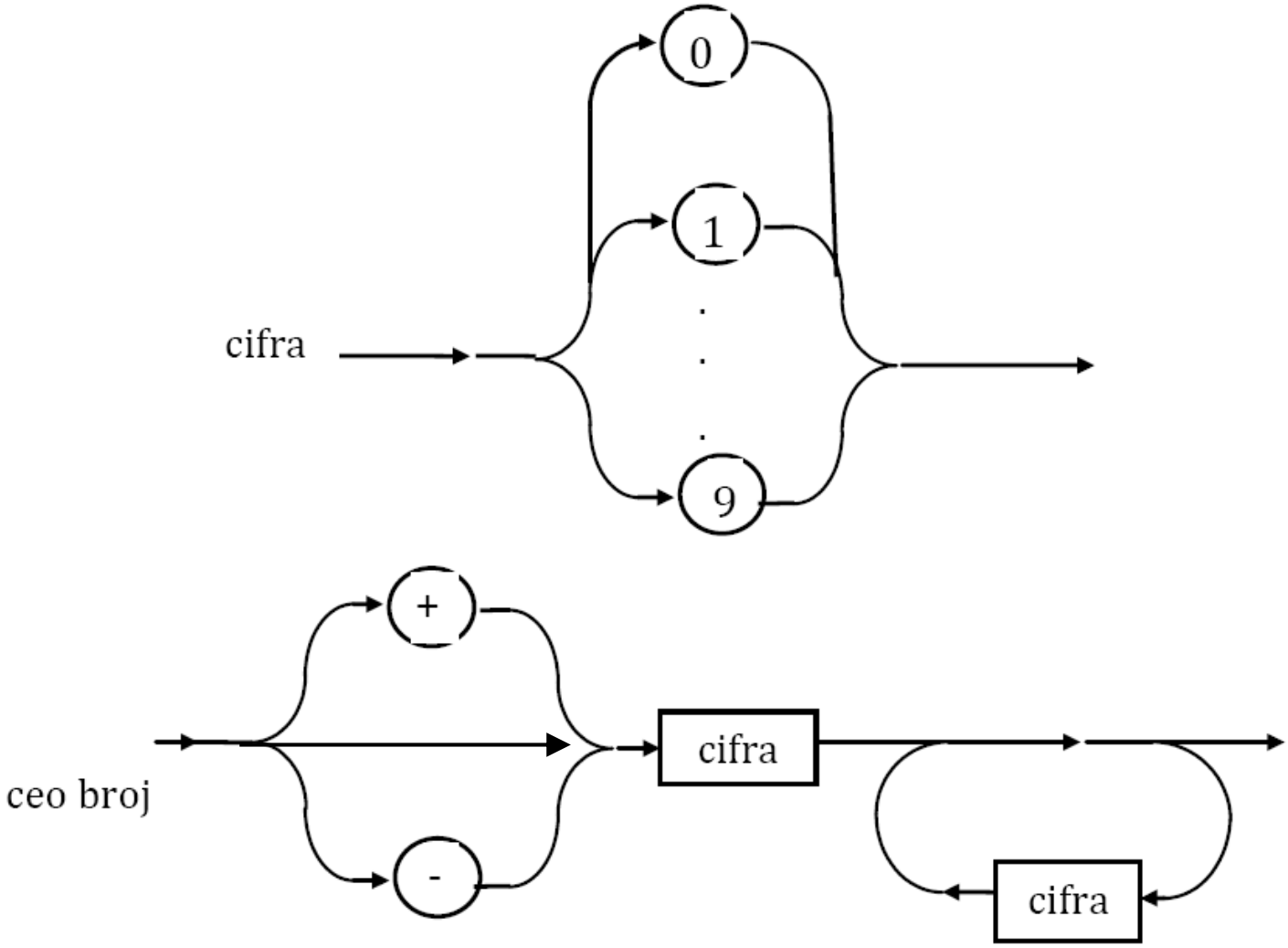


Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs

Синтаксни дијаграм (5)

Пример



Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs

Азбука језика Јава

- Сваки Јава програм је низ Unicode знакова.
- Сваки Unicode знак се представља као 16-битна реч.
- Уколико се у програму појављују не Unicode знаци, Јава преводлиац их преводи у Unicode знаке и цео програм организује по редовима.
- Сваки Unicode знак у језику Јава представља се на јединствен начин преко тзв. ескејп-секвенце:

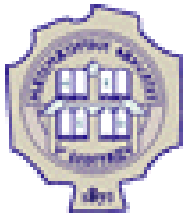
```
<hex cifra>      ::= 0|1|2|3|4|5|6|7|8|9|A|a|B|b|C|c|D|d|E|e|F|f
<izvorni znak>   ::= <bilo koji ASCII ili EBCDIC kôd>
<unicode znak>   ::=
                     \u{u}<hex cifra><hex cifra><hex cifra><hex cifra>
                     |<izvorni znak>
```

Примери:

A = \u0041 Š = \u0161 π = \u03C0

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs



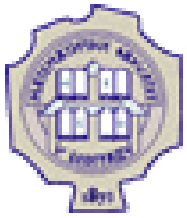
Елементарне конструкције језика Јава

У елементарне конструкције (токене) убрајамо елементе језика које компајлер издваја као недељиве целине приликом превођења програма. У елементарне конструкције спадају:

- Идентификатори
- Литерали
- Сепаратори
- Оператори
- Кључне речи
- Коментари
- Белине

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs



Елементарне конструкције језика Јава (2)

- Користећи Бекусову нотацију, то записујемо на следећи начин:

```
<elementarna konstrukcija> ::= <identifikator>
                              |<ključna reč>
                              |<literal>
                              |<separator>
                              |<operator>
                              |<komentar>
                              |<belina>
```

- Изворни програм језика Јава је низ знакова и он се прослеђује преводиоцу.
- Преводилац анализом програма издваја наредбе, а потом елементарне конструкције (енг. tokens) од којих се креирају сложене конструкције језика Јава.

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs

Идентификатори

- Идентификатор, као што и његово име казује, служи за идентификовање неке конструкције у Јави.
- Све конструкције у Јави, као што су: променљиве, класе, методи итд. на јединствен начин се именују преко идентификатора
- Синтакса идентификатора је следећа:

```
<identifikator>          ::= <unicode slovo>|$|_
                             {<unicode slovo>|$| _|<unicode cifra>}
```

- Идентификатор мора почети словом, знаком за долар или цртом за подвлачење.
  У преосталом делу идентификатора, поред ових знакова, могу да се појаве и цифре.

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs

Идентификатори (2)

Пример

- Следеће речи представљају идентификаторе у Јави:

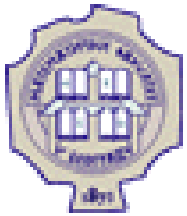
ImeKlase	_read
X	xy123
\u03C0	$b1
I_Ovo_Je_Identifikator	x1y21T23
jo\u0161	MojeSve

- Следеће речи не представљају идентификаторе у Јави:

2brata	- почиње цифром
Novi Sad	- садржи бланко
lose-definisan	- садржи црту (знак минус)
x,y.c	- садржи запету и тачку
a&b	- садржи недозвољени знак &.

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs



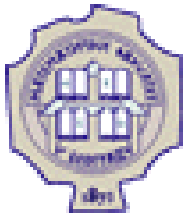
Идентификатори (3)

- Из синтаксне дефиниције не следи да је дужина имена ограничена. Међутим, из практичних разлога, она је увек ограничена.
- Приликом дефинисања сопствених имена препоручује се избор прегледних имена:

  strana, Krug, a1, x11, obimKvadrata, Masa1, godPrihod,...
- Док следећа имена могу лако бити помешана међусобно и проузроковати грешке у програму:

  x1yz, xy1z, x1zy,...
- У Јави постоји разлика између малих и великих слова, тако да су Pera1 и pera1 два различита идентификатора.

Математички факултет
{vladaf,kartelj}@matf.bg.ac.rs



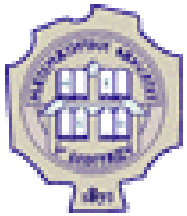
Кључне речи

- Кључне речи су идентификатори који имају специјалну намену у језику Јава и не могу се користити за именовање других ентитета (променљивих, класа и метода).
- Следеће кључне речи постоје у Јави:

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

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs



Литерали

- Литерали у језику су речи које представљају саме себе.
- У Јави то су константе примитивног типа или конкретан примерак класе String.
- Помоћу Бекусове нотације то записујемо на следећи начин:

```
<literal>        ::= <celobrojni>
                   |<realni>
                   |<logički>
                   |<znakovni>
                   |<stringovni>
```

Математички факултет
{vladaf,kartelj}@matf.bg.ac.rs

Целобројни литерали

- Цели бројеви у Јави могу бити записани као: декадни, октални или хексадекадни (а од верзије 7 и као бинарни).

```
<nenula cifra>  ::= 1|2|3|4|5|6|7|8|9
<cifra>         ::= 0| <nenula cifra>
<binarna cifra> ::= 0|1
<oktalna cifra> ::= 0|1|2|3|4|5|6|7
<hex cifra>     ::= 0|1|2|3|4|5|6|7|8|9|A|a|B|b|C|c|D|d|E|e|F|f
```

Пример

- Коректно записани целобројни литерали су:

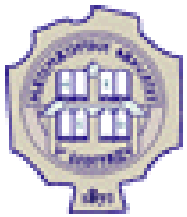
0	125	3567
0564	0XABC	0x23a4
56L	04343343l	0XE653aL

- Некоректно су записани следећи литерали:

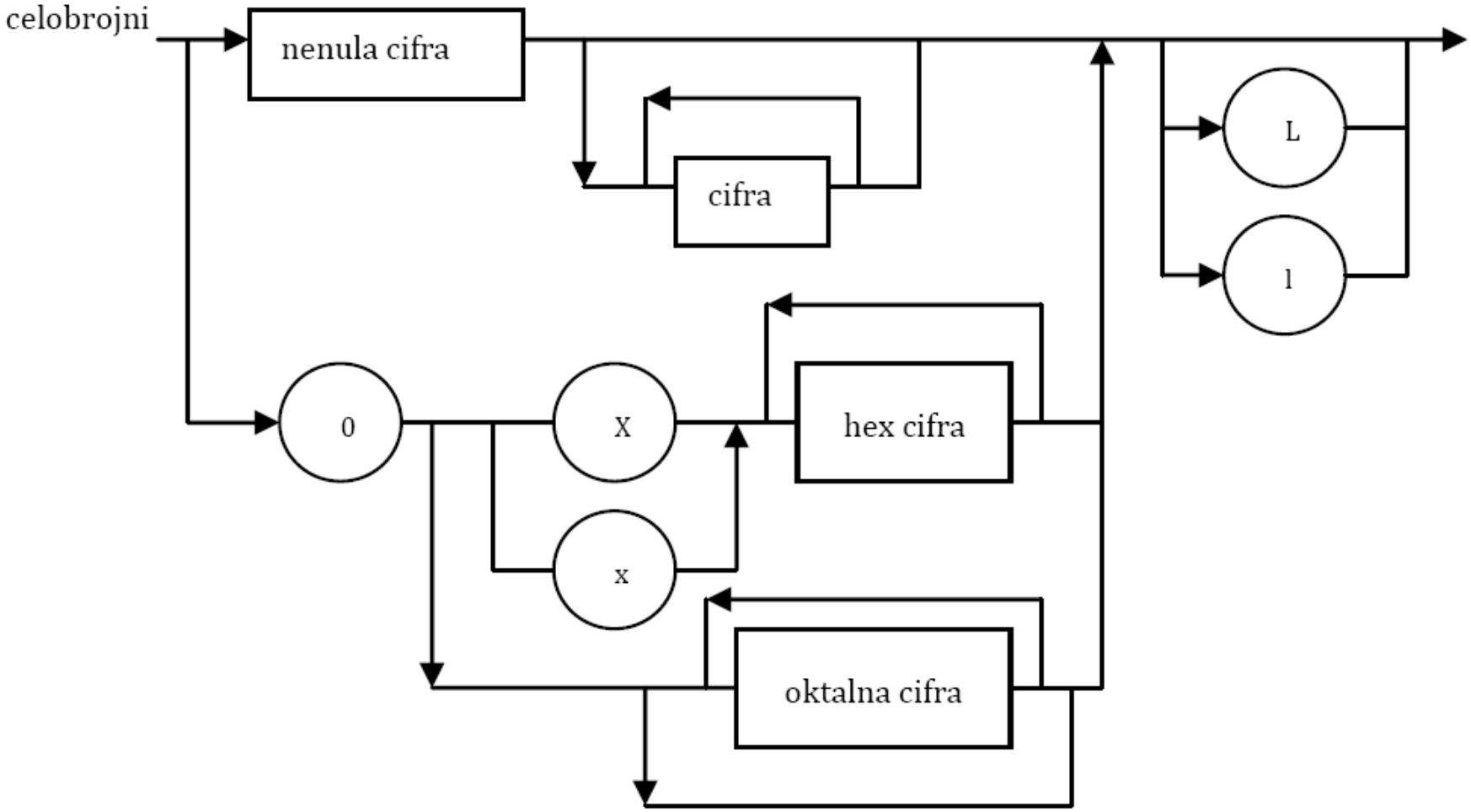
05693 - октални број садржи декадну цифру већу од 7
123A4 - декадни број садржи недекадну цифру
0XaBH2 - појављује се нехексадекадна цифра у запису броја

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs

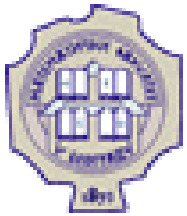


Целобројни литерали (2)



Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs



Реални литерали

- Реални литерали су константе које се записују у облику покретне тачке.
- У Јави разликујемо два типа реалних литерала: float и double.
- Разлика између ових типова појављује се само у прецизности записа литерала. Реални литерал мора садржати бар једну цифру и било децималну тачку, било експонент.
- Реални литерали могу да се изразе у:

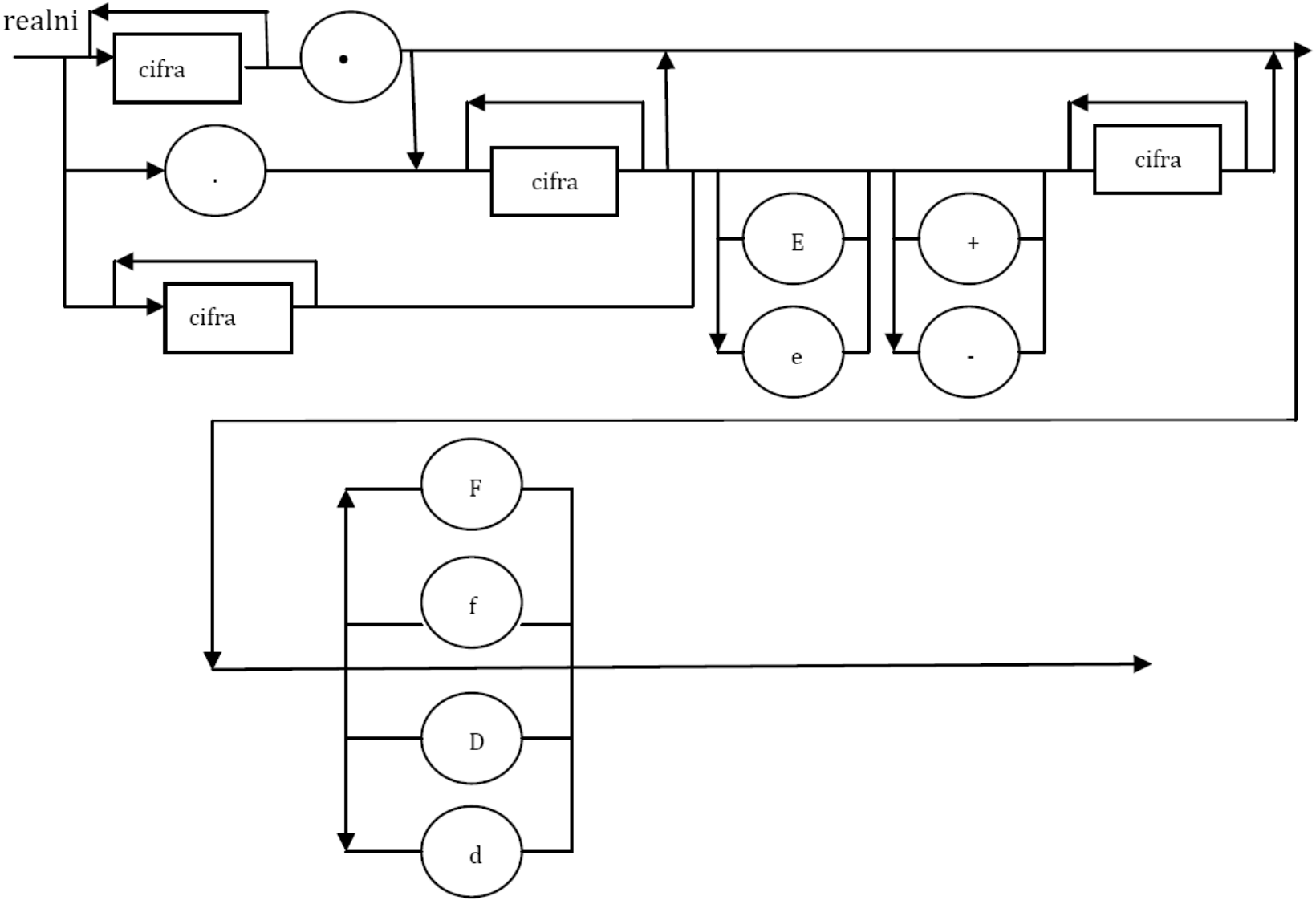
  - позиционом запису или

  - експоненцијалном запису.

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs

Реални литерали (2)



Математички факултет
{vladaf,kartelj}@matf.bg.ac.rs

Реални литерали (3)

Пример

- Следеће речи су синтаксно исправни реални литерали:

23.57	8.879f	.345d
.569	0.569F	4455.D
3.14	2e-5f	0.003e+4d
123E-5	1.456575e+12F	3.5E7

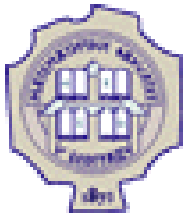
- док следећи записи не представљају реалне литерале:

0x0.233	- не постоје хексадекадни реални литерал
5F	- није реални литерал
53.4-12	- недостаје слово E
999E	- недостаје цео број иза слова E

- Уколико је литерал типа float слово f или F, мора се навести на крају литерала.
- Тип double је подразумевани тип за реални литерал те се слово d или D не мора навести.

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs



Логички литерали

Постоје два логичка литерала представљена резервисаним речима false (нетачно) и true (тачно).

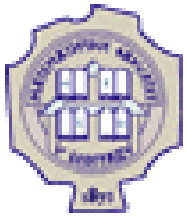
Дакле:

```
<logički literal> ::= false | true
```

Приликом поређења неких величина увек се као вредност добија true или false .

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs



Знаковни литерали

- Синтакса знаковног литерала се описује на следећи начин:

```
<znakovni>                  ::= <bilo koji znak osim apostrofa i obrnute kose crte>
                            | <eskejp sekvenca>
<eskejp sekvenca>           ::= \b | \t | \n | \f | \r | \" | \' | \\
                            | <oktalni eskejp>
                            |<Unicode eskejp>
<oktalni eskejp>            ::= \0|1|2|3[oktalna cifra][oktalna cifra]
<Unicode eskejp>            ::= \u {u}<hex cifra><hex cifra><hex cifra><hex cifra>
```

- У Јави се могу користити следеће ескејп-секвенце:

‘\‘‘ - апостроф
‘\"‘ - наводник
‘\\‘ - обрнута коса црта
‘\r‘ - знак за повратак на почетак реда
‘\n’ - знак за прелазак у нови ред
‘\f‘ - знак за прелазак на нову страну
‘\t‘ - знак табулатора
‘\b‘ - знак за повратак за једно место уназад.

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs

Знаковни литерали (2)

Пример

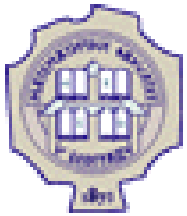
```
class Znaci {
        public static void main (String args []) {
                char zn, ch; zn='M'; ch='\"';
                System.out.println("Izvorni znaci : "+zn+ch);
                zn='\1';
                ch='\114';
                System.out.println("Oktalne sekvence : "+zn+ch);
                zn='\u0065';
                ch ='\u1132';
                System.out.println("Unicode sekvence : "+zn+ch);
        }
}
```

Извршавањем програма добија се:

```
Izvorni znaci : M"
Oktalne sekvence : L
Unicode sekvence : e?
```

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs



Стринговни литерали

- Стринговни литерал је ниска знакова између наводника.
- Као знак стринговног литерала може да се појави било који знак осим апострофа и обрнуте косе црте или ескејп секвенца.
- То преко Бекусове нотације записујемо на следећи начин:

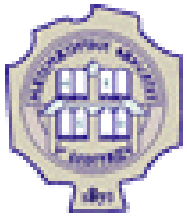
```
<stringovni literal>::= "{[<znak osim apostrofa i obrnute kose crte>]
                        |[<eskejp sekvenca>]}"
```

- Примери стринговних литерала:

```
"" // prazan string
"Programiranje i matematika" // neprazan string
"Ovo je navodnik \", a ovo ne \u3232" //string sa eskejp sekvencama
```

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs



Сепаратори

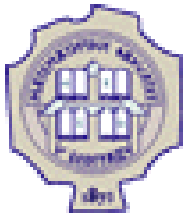
- У Јави постоји неколико знакова који служе за раздвајање једне врсте елементарних конструкција од других. На пример, у сепараторе спада симбол ; који служи за раздвање наредби у Јави.
- У сепараторе спадају следећи знаци:

```
<separator> ::= ( | ) | { | } |[ | ] | ; | : | , | .
```

- Сепаратори служе само за раздвајање и не одређују операције над подацима.

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs



Оператори

- Оператори омогућавају операције над подацима.
  Подаци на које се примењују оператори називају се операнди.
- Операторе можемо да класификујемо на више начине.
  Према позицији у односу на операнде можемо да разликујемо префиксне, инфиксне и постфиксне операторе.
- Међутим, с обзиром на то да ли се помоћу оператора додељује вредност или не, можемо да разликујемо две класе оператора: операторе доделе и остале.

```
<operator>                ::= <operator dodele>
                            |<opertor bez dodele>
<operator bez dodele>     ::= <aritmetički opertor>
                            |<relacioni opertor>
                            |<operator po bitovima>
                            |<logički operator>
                            |<uslovni operator>
                            |<instancni operator>
```

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs

Аритметички оператори

- Аритметички оператори заједно са операндима и сепараторима служе за формирање аритметичких израза. Аритетички изрази служе за израчунавање вредности.

```
<aritmetički operator>::= + | - | * | / | % | ++ |--
```

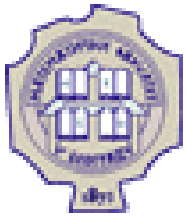
- Оператори + и – могу бити префиксни и инфиксни.
- Поред познатих оператора + – * и /, оператор % се користи за рачунање остатка при дељењу.
- Оператори ++ и –– служе за увећање, односно умањење вредности израза за 1.

Пример

7*3 - 7 / 2 + 4	▶ 21 – 7/2 +4	// Realizuje se množenje
21 – 7/ 2 + 4	▶ 21 - 3 + 4	//Realizuje se celobrojno deljenje
21 - 3 + 4	▶ 18 + 4	// Realizuje se oduzimanje
18 + 4	▶ 22	// Realizuje se sabiranje

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs



Релациони оператори

- Релациони оператори се могу још назвати и операторима поређења. Они служе за поређење вредности операнада.

```
<relacioni operator> ::= == | != | < | > | >= | <=
```

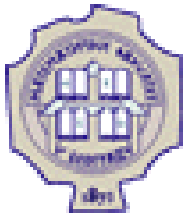
- У Јави се за испитивање да ли су два операнда једнака користи се симбол == (двострука једнакост).
- За испитивање да ли су два операнда различита користи се оператор != . Резултат примене релационих оператора је увек логичког типа (false или true).

Пример

```
(2*3 -10/7)!=(6-7%2)      ► (6 -10/7)!=(6-7%2)      //Mnozenje
(6 -10/7)!=(6-7%2)        ► (6 -1)!=(6-7%2)         //Deljenje
(6-1)!=(6-7%2)            ► 5 != (6-7%2)            //Oduzimanje
5!=(6-7%2)                ► 5 != (6-1)              //Racunanje ostatka
5!=(6-1)                  ► 5 != 5                  //Oduzimanje
5 != 5                    ► false
```

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs



Битовни оператори

- Оператор по битовима може бити логички или оператор померања.

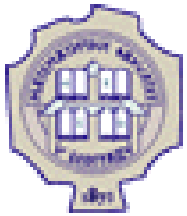
```
<operator po bitovima> ::= & | | | ~ | ^ | << | >> | >>>
```

- (Овде је дисјункција по битовима подвучена, тј. означена са ⊥, да би се разликовала од универзалног метасимбола |).

Оператор	***Операција***
~	Битовна негација (NOT)
&	Битовна конјукција (AND)
|	Битовна дисјункција (OR)
^	Битовна ексклузивна дисјункција (XOR)
<<	Померање (шифтовање) улево
>>	Померање (шифтовање) удесно
>>>	Померање (шифтовање) удесно са нулама

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs



Логички оператори

- Постоје три основна логичка оператора . То су конјукција, дисјункција и негација:

```
<logički operator> ::= && | || | !
```

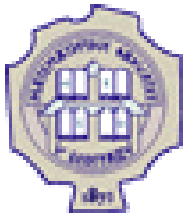
- Оператор ! је унарни и префиксни, док су оператори && и || бинарни и инфиксни.
- Као операнди код логичких оператора могу се појављивати само подаци логичког типа.
- Ефекте примене наведених оператора можемо да опишемо следећим таблицама:

p	! p
false	true
true	false

p	q	p && q	p || q
false	false	false	false
false	true	false	true
true	false	false	true
true	true	true	true

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs



Логички оператори (2)

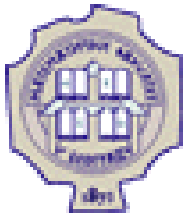
Пример

- Израчунавање сложеног израза (2 < 3) && (3 != 4) || false се реализује на следећи начин:

```
//Realizuje se prvo poredjenje
(2 < 3) && (3 != 4) || false     ► true && (3!=4) || false
//Realizuje se drugo poredjenje
true && (3!=4)|| false           ► true && true||false
//Realizuje se konjunkcija
true && true || false            ► true || false
//Realizuje se disjunkcija
true || false                    ► true
```

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs



Условни и инстанцни оператори

- Условни оператор се описује помоћу знака питања и двотачке:

```
<uslovni operator> ::= (? : )
```

- Условни оператор се најчешће користи у форми:

```
<logicki izraz>?<prvi izraz>:<drugi izraz>
```

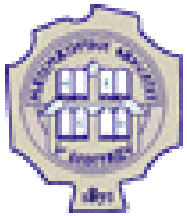
- Помоћу инстанцног оператора проверава се да ли конкретан примерак припада некој класи.

```
<instancni operator>::= instanceof
```

Оператор instanceof генерише вредност true ако је објекат примерак наведене класе (или интерфејса), а у супротном даје вредност false.

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs



Оператори доделе

Оператори доделе као што им име казује, служе да доделе вредност некој промењивој.

```
<operator dodele> ::= <prost operator dodele>| <sastavni operator dodele>
```

Основни облик оператора доделе је:

```
<prost operator dodele> ::= =
```

Оператор доделе се најчешће употребљава у форми:

```
<promenljiva> = <izraz>
```

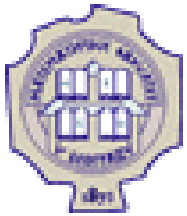
Наравно, тип промењиве мора да буде компатибилан са типом израза.

Пример Оператор доделе се може употребити и у тзв. ланчаном облику за вишеструко додељивање.

```
m = n = k = 5; // k dobija vrednost 5,
               // kako je i vrednost izraza k=5 takodje 5,
               // n dobija vrednost 5,
               // a po tom principu i m dobija vrednost 5.
```

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs



Оператори доделе (2)

- Саставни оператори доделе настају комбиновањем неких претходних оператора и простог оператора доделе.

```
<sastavni operator dodele>      ::= +=
                                  | -=
                                  | *=
                                  | /=
                                  | %=
                                  | &=
                                  | |=
                                  | ^=
                                  | <<=
                                  | >>=
                                  | >>>=
```

- Конструкције типа S = S+xxxx се краће запише у облику S += xxxx.

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs

Оператори доделе (3)

Пример

P *= a; је истоветно са: P = P*a;

d /= x+y*z; је краћи запис за: d = d/(x+y*z);

И саставни оператори доделе могу бити уланчани:

```
int a=2, b=3, s=5, s1=1;
s+=s1+=a*b // s1 dobija vrednost 7, a s vrednost 12.
```

Математички факултет

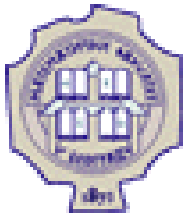
{vladaf,kartelj}@matf.bg.ac.rs

Коментари

- Коментари служе да се објасне поједина места у програму.
- Коментари су, пре свега, намењени човеку, али се у Јави могу искористити и за аутоматско генерисање документације.
- Конструкције Јаве су често довољно јасне па коментари понекад могу бити и сувишни.
- Пожељно је на почетку програма објаснити чему програм служи, ко га је писао, када је написан итд.
- У Јави постоје 3 врсте коментара:
  1. Вишелинијски (коментар у стилу језика C)
  2. Једнолинијски (коментар у стилу језика C++)
  3. Документациони.

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs



Коментари (2)

- Коришћењем Бекусове нотације, коментар дефинишемо на следећи начин:

```
<komentar>::=<jednolinijski> | <višelinijski>|<dokumentacioni>
```

Једнолинијски коментари се могу писати од почетка реда или у реду где се завршава нека наредба

```
<jednolinijski>::= //<unicode znak različit od znaka za kraj reda>
                        {<unicode znak različit od znaka za kraj reda>}
                        <znak za kraj reda>
```

На пример, има смисла писати:

```
// Sledi inicijalizacija promenljivih;
s=0; // pocetna vrednost sume
```

Вишелинијски коментар почиње симболима '/*', а завршава се симболима '*/'.

```
<višelinijski> ::= /* {{<unicode znak različit od *> }
                        * <uncode znak različit od /> } */
```

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs

Коментари (3)

- Пример вишелинијског коментара:

```
/* Ovo je komentar
koji zauzima
tri reda */
```

- Документациони коментар је једна врста вишелинијског коментара.

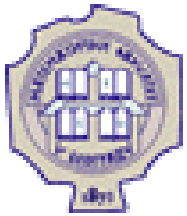
```
<dokumentacioni> ::= /** {{<unicode znak različit od * > } *
                       <unicode znak različit od / > } */
```

- Документациони коментар се може користити за аутоматско генерисање документације.
- Пример документационог коментара:

```
/** U ovom delu vrsi se korekcija nekih podataka Korekcija se vrsi na
osnovu podataka dobijenih iz banke i ucitanih sa Interneta */
```

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs



Белине

- Белина је знак који нема графички приказ на излазном уређају.
- Белине служе за међусобно раздвајање елементарних конструкција и за обликовање програма.

```
<belina>          ::= <razmak>
                  |<horizontalni tab>
                  |<znak za kraj reda>
                  |<znak za novu stranu>
                  |<znak za kraj datoteke>
```

Математички факултет

{vladaf,kartelj}@matf.bg.ac.rs

Захвалница

Велики део материјала који је укључен у ову презентацију је преузет из презентације коју је раније (у време када је он држао курс Објектно орјентисано програмирање) направио проф. др Душан Тошић.

Хвала проф. Тошићу што се сагласио са укључивањем тог материјала у садашњу презентацији, као и на помоћи коју ми је пружио током конципцирања и реализације курса.