

Elementi i struktura Java programa

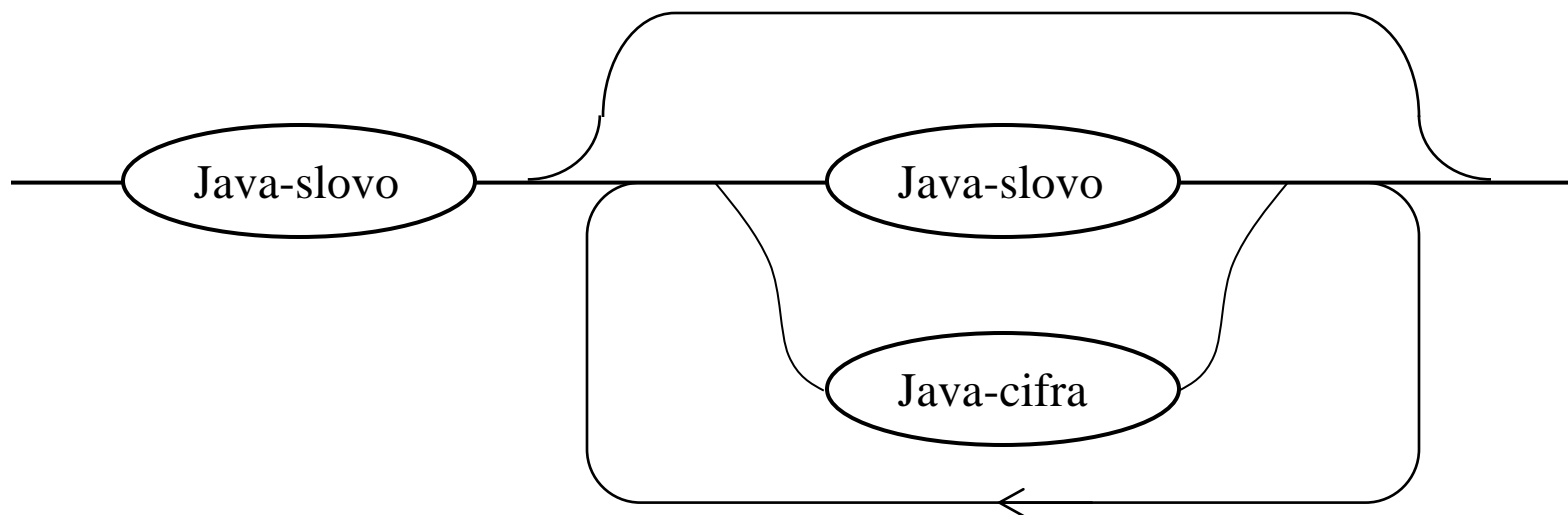
Osnovni elementi Java programa

- Osnovni elementi jezika Java:
 - Reči (lekseme)
 - Identifikatori
 - Literali
 - Specijalni simboli
 - Rezervisane reči
 - Praznine
 - Komentari

Identifikatori

- Identifikatori služe za imenovanje promenljivih, metoda, klasa, i drugih konstrukcija u programu
 - Primeri: `broj`, `main`, `System`, `out`
- Razlikuju se velika i mala slova
 - Primer: identifikatori `broj` i `Broj` nisu isti
- Identifikatori se grade od slova, cifara i nekoliko specijalnih znakova:
 - Počinju slovom, ili znakom `_` ili `$`
 - Sledeći znaci mogu biti i cifre
- Rezervisane reči (npr. `class`, `void`, `int`, itd.) ne mogu se koristiti kao identifikatori
- Identifikatori mogu biti proizvoljne dužine

Identifikatori



- Java-slovo je svaki znak za koji metod `Character.isJavaLetter` vraća vrednost `true`
- Java-cifra je svaki znak za koji metod `Character.isJavaLetter` vraća vrednost `false`, a metod `Character.isJavaLetterOrDigit` vraća vrednost `true`

Primeri identifikatora

Ispravni identifikatori

`ime_i_prezime`

`Suma`

`SUMA`

`element2`

`p218`

Neispravni identifikatori

`element 1` (sadrži prazninu)

`p*218` (sadrži *)

`class` (rezervisana reč)

`2elem` (počinje cifrom)

Konvencije za Java identifikatore

- Pored pravila za formiranje ispravnih Java identifikatora, postoje i konvencije (preporuke) kako imenovati različite konstrukcije jezika
- Imena klasa treba da počinju velikim slovom
Primer: `OvoJeIspravnoImeKlase`
- Imena promenljivih, metoda i paketa treba da počinju malim slovom
Primeri: `broj`, `x`, `deli`
- Ako je identifikator sastavljen od više reči, koristiti velika slova za njihovo odvajanje (sem eventualno kod prve reči)
Primeri: `charArray`, `fileNumber`, `ClassName`
- Izbegavati korišćenje `_` kao prvog znaka identifikatora
Primeri: `_read`, `_write`

Literali

- **Literali** su konstantne vrednosti koje direktno uključujemo u program
- Tipovi literala u Javi:
 - **Brojevi**
 - Celi brojevi (*integer*)
 - Realni brojevi (tačnije brojevi sa pokretnim zarezom, *floating point*)
 - **Logički literali** (*boolean*)
 - **Znakovi** (*character*)
 - **Stringovi** (*string*)

Literali: celi brojevi

- Celi brojevi se mogu navoditi u nekoliko različitih brojnih sistema, sa osnovama 10, 16, 8 i 2
- Decimalni zapis (sa osnovom 10) – standardan zapis
Primer: 42
- Heksadecimalni zapis (osnova 16) – počinje znakovima 0x
Primer: 0x2A
- Oktalni zapis (osnova 8) – počinje sa 0
Primer: 052
- Binarni zapis (osnova 2) – počinje sa 0b
Primer: 0b101010

Literali: realni brojevi

- Strogo uzevši, *racionalni* brojevi
- Dve vrste zapisa: standardni i “naučni” (*scientific*)
- Standardni zapis – decimalni broj, tačka, decimalni broj
Primer: 583.45
- Naučni zapis – mantisa + eksponent
Primer: 5.8345E2 ($5.8345 \cdot 10^2$)

Literali: logički

- Dve moguće vrednosti: `true` i `false`

Literali: znakovi

- Znakovni literali predstavljaju znakove iz Unicode rasporeda
- Unicode znakovi su reprezentovani sa 16 bita (2 bajta, odnosno 65536 mogućih vrednosti), tako da pored slova engleske abecede, cifara i standardnih znakova interpunkcije (stari 8-bitni ASCII raspored) mogu da se predstave znakovi iz drugih jezika
- Znakovni literal navodi se kao znak okružen jednostrukim navodnicima

Primer: znakovni literal za slovo a se piše `'a'`

- Specijalni znakovi koji se ne mogu jednostavno “otkucati” na tastaturi se predstavljaju *escape* znacima

Primeri: `'\n'` znak za novi red

`'\b'` znak za brisanje (*backspace*)

Literali: stringovi

- String literali su sekvence znakova između dvostrukih navodnika
- Primeri: "Ovo je string", "Uvod u programiranje"
- Escape znaci unutar stringova i znakovnih literala:
 - `\n` *line feed* LF
 - `\r` *carriage return* CR
 - `\t` horizontalni tabulator HT
 - `\b` *backspace* BS
 - `\"` dvostruki navodnici "
 - `\'` jednostruki navodnici '
 - `\\` *backslash* \
 - `\broj` redni broj znaka (ASCII raspored, od 0 do 255) u oktalnom zapisu
 - `\uxxxx`, gde je **xxxx** heksadecimalni broj – redni broj Unicode znaka

Primeri literala

Ispravno zapisani brojevi

23

3.14

4.0E-73

Neispravno zapisani brojevi

1,5 (sadrži ',')

1.23.50 (dve decimalne tacke)

1.5Em (sadrži slovo m)

Ispravni stringovi

"Novi Sad"

"skola \"I. Sekulic\""

"NOVI S\u0041D\r\n"

Neispravni stringovi

Novi Sad (fale dupli navodnici)

'NoviSad' (koristiti duple navodnike)

"skola "I. Sekulic"" (mora \")

Ispravni znaci

'a'

'\uffff'

'"' (ne mora \")

Neispravni znaci

' ' (fali znak)

' ' ' (unutrašnji apostrof)

'\ "'

Specijalni simboli i rezervisane reči

- **Specijalni simboli** su operatori i separatori u jeziku, tj. znaci ili nizovi znakova koji imaju neko posebno značenje u programu
 - Neki operatori : < > + += ++ &&
 - Separatori : { } [] () ; , .
- **Rezervisane (ili ključne) reči** su posebne lekseme jezika koje imaju tačno određeno značenje
- Ključne reči se koriste za opis naredbi i drugih konstrukcija programa, i deo su sintakse programskog jezika

Rezervisane reči

- Rezervisane reči se ne mogu koristiti kao identifikatori u programu
- Rezervisane reči u Javi:

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

Praznine

- **Praznine** (engl. *whitespace*) u Javi su posebni znaci u koje spadaju:
 - Oznaka za kraj reda
 - Oznaka za tabulator
 - “Prazan” znak (*space*)
 - Kontrolni znak za štampač (*form feed*)
 - i sl.
- Zanemaruju se prilikom prevođenja
- Strogo uzevši, služe samo za razdvajanje leksema
- Utiču na lepši izgled i bolju preglednost programa

Komentari

- **Komentari** su delovi programa koji ne utiču na smisao i suštinu programa i zanemaruju se prilikom prevođenja (kompajler se prema njima odnosi kao da su praznine)
- U Javi postoje **tri vrste komentara**:
 - **Obični komentari** - tekst koji se nalazi između simbola `/*` i `*/`
 - **Jednolinijski komentari** - tekst koji se nalazi između simbola `//` i kraja reda
 - **Dokumentacioni komentari** - tekst koji se nalazi između simbola `/**` i `*/`. Ovaj komentar služi za automatsko generisanje dokumentacije programa (pomoću alata Javadoc) pa je bitno gde se navodi komentar, dok tekst komentara mora biti strukturiran po određenim pravilima

Primeri komentara

- **Obični komentari:**

```
/* Ovo je komentar  
   koji se prostire na vise linija */  
/* A ovaj komentar je u jednoj liniji */
```

- **Jednolinijski komentari:**

```
// Ovo je jednolinijski komentar  
// A ako zelimo vise linija  
// moramo pre svake linije komentara staviti //
```

- **Dokumentacioni komentar:**

```
/** Slican je obicnom komentaru, s tim da moze  
    sadrzati tagove, npr.  
        @author Milos Radovanovic  
        @version 1.0  
    koji imaju posebno znacenje pri generisanju  
    dokumentacije */
```

Struktura Java programa

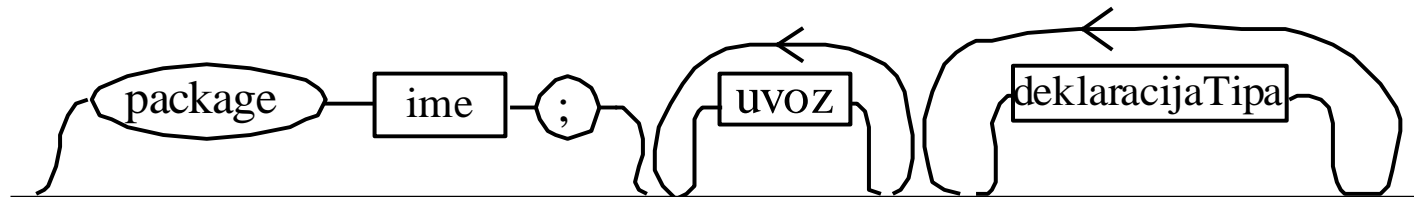
- Program u Javi se sastoji od deklaracija klasa, interfejsa i nabrojivih tipova podataka
 - Klase, interfejsi i nabrojivi tipovi su referencijalni tipovi podataka
 - Na ovom kursu pokrićemo (delimično) klase i nabrojive tipove, dok će interfejsi i ostali aspekti klasa i nabrojivih tipova biti obrađeni na kursevima Objektno-orijentisano programiranje 1 i 2

- Mora biti deklarirana bar jedna klasa sa metodom čije je zaglavlje

```
public static void main(String[] args)
```

- Sve deklaracije se mogu nalaziti u jednom fajlu ili mogu biti raspoređene u više fajlova
- Fajlove koji čine Java program zovemo jedinice prevođenja
- Jedinica prevođenja treba da se imenuje isto kao i klasa koja sadrži `main()` metod (ako postoji, u protivnom može proizvoljno)
- Ustaljena praksa: svaka jedinica prevođenja sadrži tačno jednu deklaraciju referencijalnog tipa, i ima isto ime kao taj referencijalni tip

Jedinica prevođenja



- **Paketi** se uvode radi grupisanja referencijalnih tipova
 - Primer: `package uup;`
 - Na početku fajla (jedinice prevođenja) svi referencijalni tipovi deklarirani u tom fajlu svrstavaju se u isti paket sa zadatim imenom (ako se ovaj deo ne navede ref. tipovi će pripasti tzv. anonimnom paketu)
- U delu **uvoz** navodimo ime paketa i ime koje se iz njega uvozi
 - Primer: `import java.util.HashSet;`
- **Deklaracija tipa** – deklarisanje klasa, interfejsa i nabrojivih tipova