

The background of the slide is a solid dark blue. Overlaid on this are several sets of thin, white, curved lines that create a sense of motion and depth, resembling stylized waves or a wireframe mesh. These lines are more concentrated in the upper half of the slide and fade out towards the bottom.

**Kontrola toka i operatori**

**Jednostavni primjeri Java programa**

# Sadržaj predavanja

- Kontrola toka: "if-else"
- Kontrola toka: "while" i "do-while" petlje
- Kontrola toka: "for" petlja
- Kontrola toka: "switch" s poljima i "for" petljom
- Operatori
- Organizacija Java programskog koda
- Primjer: ispisivanje pozdravne poruke
- Primjer: zbrajanje cijelih brojeva
- Primjer: uspoređivanje brojeva

# Kontrola toka: "if-else"

```
if (prviBroj > treciBroj) {  
    System.out.println("Najveći broj je " + prviBroj);  
}
```

```
    if (drugiBroj > treciBroj) {  
        System.out.println("Najveći broj je " + drugiBroj);  
    }  
    else {  
        System.out.println("Najveći broj je " + treciBroj);  
    }
```

```
if (drugiBroj > treciBroj) {  
    System.out.println("Najveći broj je " + drugiBroj);  
}  
else if (drugiBroj > prviBroj) {  
    System.out.println("Najveći broj je " + treciBroj);  
}  
else {  
    System.out.println("Broj " + drugiBroj + " je najmanji!");  
}
```

## Kontrola toka: "while" i "do-while" petlje

```
int i = 0;

while (i < 100) {
    System.out.println(i);
    i++;
}
```

```
int i = 0;

do {
    System.out.println(i);
    i++;
} while (i < 100);
```

## Kontrola toka: "for" petlja

```
for (int i = 0; i < 100; i = i + 2) {  
    System.out.println(i);  
}
```

```
int[] polje = {1, 2, 3, 4};
```

```
for (int broj : polje) {  
    System.out.println(broj);  
}
```

# Kontrola toka: "switch" s poljima i "for" petljom

```
String[] seasons = {"Spring", "Summer", "Autumn", "Winter"};

for (String season : seasons) {
    switch(season) {
        case "Spring":
            System.out.println("Proljeće");
            break;
        case "Summer":
            System.out.println("Ljeto");
            break;
        case "Autumn":
            System.out.println("Jesen");
            break;
        case "Winter":
            System.out.println("Zima");
            break;
        default:
            System.out.println("Nepostojeći naziv godišnjeg doba!");
    }
}
```

# Operatori (1/3)

- Pretpostavka: `int c = 3, d = 5, e = 4, f = 6, g = 12;`

Operator	Izraz	Objašnjenje	Rezultat
+	<code>c = d + e</code>	Zbrajanje	c poprima 9
-	<code>c = d - e</code>	Oduzimanje	c poprima 1
*	<code>c = d * e</code>	Množenje	c poprima 20
/	<code>c = g / e</code>	Dijeljenje	c poprima 3
%	<code>c = d % e</code>	Cjelobrojni ostatak	C poprima 1
+=	<code>c += 7</code>	<code>c = c + 7</code>	c poprima 10
-=	<code>d -= 4</code>	<code>d = d - 4</code>	d poprima 1
*=	<code>e *= 5</code>	<code>e = e * 5</code>	e poprima 20
/=	<code>f /= 3</code>	<code>f = f / 3</code>	f poprima 2
%=	<code>g %= 9</code>	<code>g = g % 9</code>	g poprima 3
++	<code>c++</code>	<code>c = c + 1</code> (post-inkrement)	c poprima 4
++	<code>++c</code>	<code>c = c + 1</code> (pre-inkrement)	c poprima 4



## Operatori (2/3)

- Pretpostavka: `int c = 3, d = 5, e = 4, f = 6, g = 12;`

Operator	Izraz	Objašnjenje	Rezultat
--	<code>c--</code>	<code>c = c - 1</code> (post-dekrement)	C poprima 2
--	<code>--c</code>	<code>c = c - 1</code> (pre-dekrement)	C poprima 2
<	<code>c &lt; d</code>	c manje od d?	true
>	<code>c &gt; d</code>	c veće od d?	false
<=	<code>c &lt;= d</code>	c manje ili jednako d?	true
>=	<code>c &gt;= d</code>	c veće ili jednako d?	false
==	<code>c == d</code>	c jednako d?	false
!=	<code>c != d</code>	c različito od d?	true
&&	<code>c != d &amp;&amp; d != e</code>	Oba izraza istinita?	true
	<code>c == d    d == e</code>	Barem jedan od izraza istinit?	false
!	<code>!(c == d)</code>	Negacija vrijednosti	true



## Operatori (3/3)

- Pretpostavka: `int c = 3, d = 5, e = 4, f = 6, g = 12;`

Operator	Izraz	Objašnjenje	Rezultat
<code>&amp;</code>	<code>c &amp; d</code>	<code>011 &amp; 101</code>	<code>001 (1)</code>
<code> </code>	<code>c   d</code>	<code>011   101</code>	<code>111 (7)</code>
<code>^</code>	<code>c ^ d</code>	<code>011 ^ 101</code>	<code>110 (6)</code>
<code>=</code>	<code>c = d</code>	Vrijednost d se pridružuje c	<code>c = 5</code>
<code>.</code>	<code>Math.pow(c,3);</code>	Pozivanje metode „pow()“ iz klase „Math“	<code>27</code>
<code>new</code>	<code>Vozilo prvo = new Vozilo();</code>	Instanciranje novih objekata	Referenca na novi objekt
<code>()</code>	<code>float h = (float) c;</code>	Pretvaranje cijele vrijednosti u decimalnu vrijednost	<code>h = 3.0</code>
<code>instanceof</code>	<code>vozilo1 instanceof Vozilo</code>	Provjera klase objekta	<code>true</code> ili <code>false</code>

# Organizacija Java programskog koda

- Programski kod Java aplikacija raspoređen je unutar klasa, gdje se nalaze funkcije (metode) i varijable (polja)
- Svaka Java aplikacija može se sastojati od više klasa koje su raspoređene po paketima
- Nazivi paketa određuju se prema zajedničkom kontekstu klasa koje se nalaze unutar njega
- Glavna metoda za pokretanje aplikacije zove se "main" i prima ulazne parametre komandne linije (kao kod "C"-a)
- Metoda "main" predstavlja "ulaznu točku" kod Java aplikacije
- Svaka definicija klase sadrži naziv paketa u kojem se nalazi, naziv same klase (koji mora biti jednak kao i naziv datoteke u kojoj se klasa nalazi, npr. "**HelloWorld.java**"), te metode unutar klase (npr. "main" metoda)

## Ispisivanje pozdravne poruke (bez formatiranja)

```
package hr.tvz.programiranje.java.vjezba;  
  
public class HelloWorld {  
  
    public static void main(String[] args) {  
  
        System.out.println("Programiranje u Javi!");  
  
    }  
  
}
```

## Ispisivanje pozdravne poruke (s formatiranjem)

```
package hr.tvz.programiranje.java.vjezba;  
  
public class PrintfHelloWorld {  
    public static void main(String[] args) {  
        System.out.printf("%s\n%s\n%s\n",  
                           "Programiranje", "u", "Javi");  
    }  
}
```

# Zbrajanje cijelih brojeva

```
public class Zbrajanje {  
  
    public static void main(String[] args) {  
  
        Scanner ulaz = new Scanner(System.in);  
        int prviBroj;  
        int drugiBroj;  
        int rezultat;  
  
        System.out.print("Unesite prvi cijeli broj >> ");  
        prviBroj = ulaz.nextInt();  
  
        System.out.print("Unesite drugi cijeli broj >> ");  
        drugiBroj = ulaz.nextInt();  
  
        rezultat = prviBroj + drugiBroj;  
  
        System.out.printf("Zbroj unesenih brojeva %d i %d je %d ",  
            prviBroj, drugiBroj, rezultat);  
    }  
}
```

# Uspoređivanje brojeva

```
public class NajveciOdTri {
    public static void main(String[] args) {
        Scanner ulaz = new Scanner(System.in);
        int prviBroj, drugiBroj, treciBroj;

        System.out.print("Unesite tri cijela broja >> ");
        prviBroj = ulaz.nextInt();
        drugiBroj = ulaz.nextInt();
        treciBroj = ulaz.nextInt();

        if (prviBroj > drugiBroj) {
            if (prviBroj > treciBroj) {
                System.out.println("Najveći broj je " + prviBroj);
            }
            else {
                System.out.println("Najveći broj je " + treciBroj);
            }
        }
        else {
            if (drugiBroj > treciBroj) {
                System.out.println("Najveći broj je " + drugiBroj);
            }
            else {
                System.out.println("Najveći broj je " + treciBroj);
            }
        }
    }
}
```