

## Povzetek C++

### Vhodno / Izhodne operacije na konzoli

```
cout<<      izpis na ekran
cin>>      branje s tipkovnice
```

```
cout << "Vnesi celo število: ";
cin >> i;
cout << "Število 1 je " << i << endl;
```

### Posebni znaki

```
\ " narekovaj  \\ poševnica nazaj
\n nova vrstica
```

### Podatkovni tipi in spremenljivke

Podatkovni tip	Primer deklaracije spremenljivke	Primer prireditvenega stavka
----------------	----------------------------------	------------------------------

cela števila	int x, y;	x = 0;
--------------	-----------	--------

realna števila	double a;	a = 3.14;
----------------	-----------	-----------

en znak	char z;	z = 'X';
---------	---------	----------

niz znakov	string niz;	niz = "Test";
------------	-------------	---------------

logična vrednost	bool b;	b = true;
------------------	---------	-----------

### Aritmetične operacije

+ - \* / za poljubna števila  
% samo za cela števila

### Matematične funkcije

Vključimo: #include <cmath>

Ime	Pomen
pow(x,y)	$x^y$
sin(x)	
cos(x)	
tan(x)	
exp(x)	$e^x$
log(x)	$\ln(x)$
sqrt(x)	$\sqrt{x}$
floor(x)	največje celo število, ki ni večje od x
ceil(x)	najmanjše celo število, ki ni manjše od x

### Naključna števila

Vključimo: #include <cstdlib>

#### Seme generatorja naključnih števil

```
srand( seme );
```

#### Primer izračuna naključnega celega števila:

```
x = rand();
```

## Odločitveni stavki

### Operatorji primerjav

Operator	Pomen	Primer
==	je enako	x == y
!=	ni enako	x != y
<	manjše	x < y
<=	manjše ali enako	x <= y
>	večje	x > y
>=	večje ali enako	x >= y
&&	in hkrati	(x == y && y > 0)
	ali	(x==y    x<0)

### Vejitveni stavki

```
if (pogoj) {
    stavki
}
```

```
if (pogoj) {
    stavki
} else {
    stavki
}
```

```
switch (spremenljivka) {
    case vrednost1a:
    case vrednost1b:
        stavki;
        break;
    case vrednost2:
        stavki;
        break;
    default:
        stavki;
}
```

### Zanke

```
while (pogoj) {
    stavki
}
```

```
do {
    stavki
} while (pogoj);
```

```
for (inicijalizacija; pogoj; sprememba) {
    stavki
}
```

## Funkcije

### Definicija funkcije

```
tip imeFunkcije(seznam parametrov) {...}
```

### Prenos parametrov po vrednosti

```
void ime(int x) {...}
```

### Prenos parametrov po referenci

```
void ime(int &x) {...}
```

---

## Polja

### dostop do elementov

```
imePolje[indeks]
```

### statična polja

#### deklaracija

```
tip polje[VELIKOST];
```

#### parameter funkcije

```
void izpis(tip polje[], int stElementov);
```

### dinamična polja

#### deklaracija - rezerviranje pomnilnika

```
tip *polje=new tip[VELIKOST];
```

#### brisanje polja iz pomnilnika

```
delete[] polje;
```

#### parameter funkcije

```
void izpis(tip *polje, int stElementov);
```

---

## Nizi - razred string

### deklaracija niza

```
string niz1;           // prazen niz  
string niz2(niz1);    // kopija niza  
string niz3(3, 'X');// "XXX"
```

### velikost niza

```
int n=niz.size();
```

### iskanje podniza

```
int i = niz.find(podniz);
```

i ... indeks, kjer se podniz prvič pojavi, vrne *string::npos*, če ne najde

### brisanje iz niza

```
niz.erase(i, n);
```

i ... indeks prvega znaka, ki ga brišemo

n ... število znakov, ki jih brišemo

### vrivanje v niz

```
niz.insert(i, podniz);
```

i ... indeks, kamor vrivamo

podniz ... tekst, ki ga vrivamo

### kopiranje podniza

```
string podniz = niz.substr(i, n);
```

i ... indeks prvega znaka, ki ga kopiramo

n ... število znakov, ki jih kopiramo

### dostop do posameznega znaka

```
niz[indeksZnaka]
```

### dodajanje na konec - operator +=

```
niz += "dodaj";
```

### lepljenje nizov - operator +

```
niz = niz1 + niz2;
```

### brisanje vseh znakov v nizu

```
niz.clear();
```

### testiranje, ali je niz prazen

```
bool x = niz.empty();
```

vrne true, če je niz prazen, false sicer

---

## Matrike

### dostop do elementov

```
matrika[i][j]
```

i ... indeks vrstice

j ... indeks stolpca

### statične matrike

#### deklaracija

```
tip matrika[stVrstic][stStolpcev];
```

#### kot parametri funkcij

```
void izpis(tip mat[V][S], int stV, int stS);
```

V, S ... dimenziji matrike

stV, stS ... število vrstic, število stolpcev

### dinamične matrike

#### rezervacija pomnilnika

```
tip **matrika;
```

```
matrika = new tip*[stVrstic];
```

```
for(int i=0; i<stVrstic; i++)
```

```
    matrika[i] = new tip[stStolpcev];
```

### brisanje pomnilnika

```
for(int i=0; i<stVrstic; i++)
```

```
    delete[] matrika[i];
```

```
delete[] matrika;
```

### parameter funkcije

```
void izpis(tip **matrika, int stV, int stS);
```

stV, stS ... število vrstic, število stolpcev

