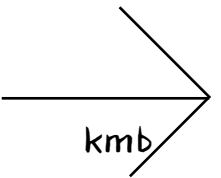




Uvod u programiranje



Od problema do algoritma

ishodi sata

- objasniti svojim riječima što je algoritam i čemu služi;
- prepoznati ulazne podatke, obradu i rezultat jednostavnog problema;
- navesti i na primjeru prepoznati bitna svojstva algoritma: konačnost, jednoznačnost/određenost, izvedivost i rezultat;
- razlikovati problem od postupka njegova rješavanja;
- zapisati jednostavan algoritam prirodnim jezikom ili pseudokodom;
- prepoznati da se isti algoritam može prikazati različitim zapisima;
- provjeriti algoritam na konkretnom primjeru i uočiti neprecizan ili nepotpun korak.

Problemški izazov: „Programiraj robota da pripremi čaj”



Napišite što robot mora učiniti da pripremi šalicu čaja.

Zapisati u 4–6 koraka.

U paru usporediti jesu li upute potpune i mogu li se jednoznačno izvršiti.

doslovno interpretirati nekoliko uputa i pokazati posljedice nepreciznosti:

Npr. „stavi čaj u šalicu” – koji čaj, koliko, prije ili poslije vode?

Zapisati primjedbe: redoslijed, preciznost, završetak, dostupne radnje/podaci.

„Može li računalo **pogoditi** što smo mislili?” Predložite poboljšanja.

Što je algoritam? vlastitim riječima oblikovati značenje pojma algoritam.

Svojstva algoritma

Loša uputa	Što nije u redu?	Svojstvo do kojeg dolazimo
„Ponavljaj prethodni korak zauvijek.”	Postupak nikada ne završava.	konačnost
„Uzmi neki prikladan broj.”	Nije jasno koji broj treba uzeti.	određenost / jednoznačnost
„U jednom koraku izračunaj sve moguće rezultate.”	Korak nije realno izvediv kao elementarna radnja.	izvedivost
„Na kraju izračunaj nešto korisno.”	Nije određen očekivani rezultat.	rezultat / učinkovit ishod postupka

problemski primjer

Za zadanu duljinu a i širinu b pravokutne prostorije izračunati površinu poda.

Što znamo? Što tražimo? Što moramo napraviti?

Moguće upute (?): „ponavljaj zauvijek”; „uzmi neki broj”; „napravi nemoguću operaciju”; „izračunaj nešto”.

Prepoznati potrebu:
da algoritam završi,
da su koraci određeni/jednoznačni i izvedivi
da daje rezultat.

skica jednostavnog dijagrama toka

zapis nije algoritam sam po sebi, nego njegov prikaz.

Usporediti zapise - što je u oba isto: ulaz, redoslijed obrade i rezultat.

ULAZ: a, b
OBRADA: $p \leftarrow a \cdot b$
IZLAZ: p

zapis prirodnim jezikom:

Učitaj duljinu a .
Učitaj širinu b .
Izračunaj $p = a \cdot b$.
Prikaži p .
Završi.

pseudokod:

učitaj a, b
 $p \leftarrow a * b$
ispiši p

Mini izazovi

Uputa: za svaki problem odredite ULAZ → OBRADA → IZLAZ.
Odaberite i napišite algoritam u najviše pet koraka.

A. Cijena ulaznica – Poznata je cijena jedne ulaznice i broj učenika. Odrediti ukupan iznos.

B. Pretvorba vremena – Zadan je broj minuta. Odrediti koliko je to cijelih sati i koliko preostalih minuta.

C. Prosjek – Zadane su tri ocjene. Odrediti njihov aritmetički prosjek.

prvo individualno, zatim u paru. Obrazložiti.

„Što moramo znati prije nego što počnemo pisati program?“

Pitanja za raspravu:

Koji je od ova tri algoritma najjednostavniji?

Gdje je potreban više od jednog koraka obrade?

Možemo li algoritam provjeriti konkretnim vrijednostima prije nego što postoji program?

**najprije razumjeti problem i osmisлити algoritam;
programski jezik dolazi poslije.**

Zaključak

ALGORITAM = konačan i jednoznačno opisan postupak za rješavanje problema.

PROBLEM → ULAZ → OBRADA → IZLAZ → PROVJERA

Bitna svojstva:

- **konačnost,**
- **određenost/jednoznačnost,**
- **izvedivost,**
- **rezultat.**

Algoritam se može **zapisati prirodnim jezikom, pseudokodom ili grafički (npr. dijagramom toka).**

- na konkretnom primjeru prepoznati što postupak čini algoritmom
- uočiti neprecizan ili nepotpun korak.

2.sat

Od algoritma do C++ programa

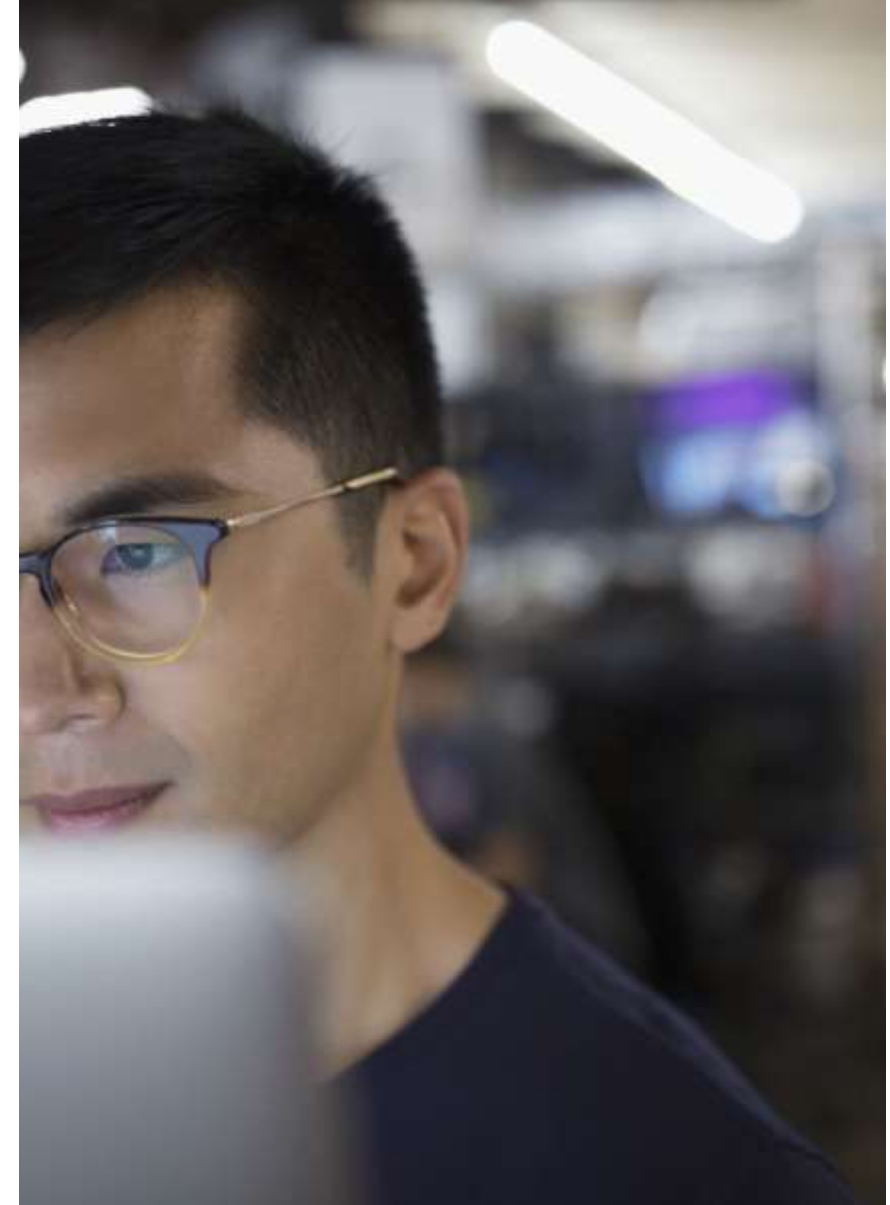


struktura ULAZ → OBRADA → IZLAZ
se ne mijenja prelaskom na programski jezik

C++ samo daje **precizna pravila za zapis podataka i naredbi**

Razumijeti uloge pojedinih dijelova programa

pravilan odabir i uporaba **osnovnih tipova podataka, varijabli, konstanti, komentara, unosa, ispisa i naredbe pridruživanja.**



- povezati dijelove jednostavnog algoritma s odgovarajućim dijelovima C++ programa;
- prepoznati osnovnu strukturu jednostavnog C++ programa i ulogu funkcije main;
- razlikovati ključnu riječ, identifikator i vrijednost/literal u jednostavnom programu;
- odabrati prikladan osnovni tip podatka (int, double, char, bool) za zadanu vrijednost;
- objasniti ulogu varijable i konstante te navesti smislen identifikator;
- razlikovati deklaraciju, inicijalizaciju i kasniju promjenu vrijednosti varijable;
- protumačiti naredbu pridruživanja kao promjenu vrijednosti varijable, a ne matematičku jednakost;
- prepoznati unos, obradu i ispis u kratkom C++ programu;
- uočiti svrhu komentara i nekoliko osnovnih pravila ispravnog zapisa programa.

```
pes.Operator):  
X mirror to the selected  
ect.mirror_mirror_x"  
or X"
```

```
ntext):  
xt.active_object is not
```

PROBLEM → ALGORITAM → PROGRAM → PROVJERA

ULAZ → OBRADA → IZLAZ

ULAZ a,b

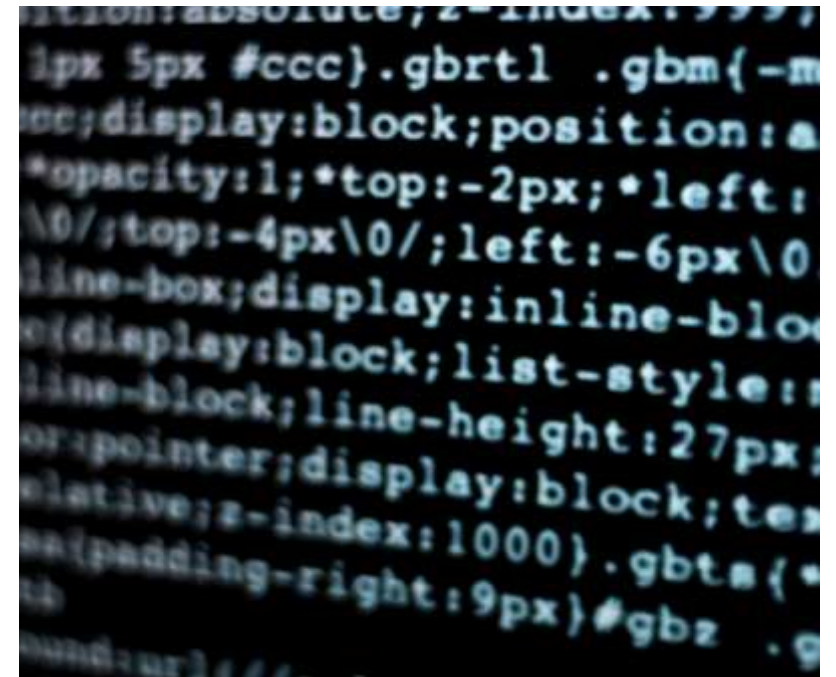
$p \leftarrow a \cdot b$

IZLAZ p

Imamo algoritam.

Trebamo pohraniti podatke, izračunati i prikazati rezultat

Što još računalu nedostaje da bi moglo izvršiti algoritam?



Problem: Za zadanu

duljinu a i širinu b

pravokutne prostorije

izračunati i ispisati

površinu poda.

C++ zapis:

```
#include <iostream>
using namespace std;
```

```
int main()
{
```

```
    double a, b, p;
```

Algoritam:

učitaj a , b

$p \leftarrow a * b$

ispiši p

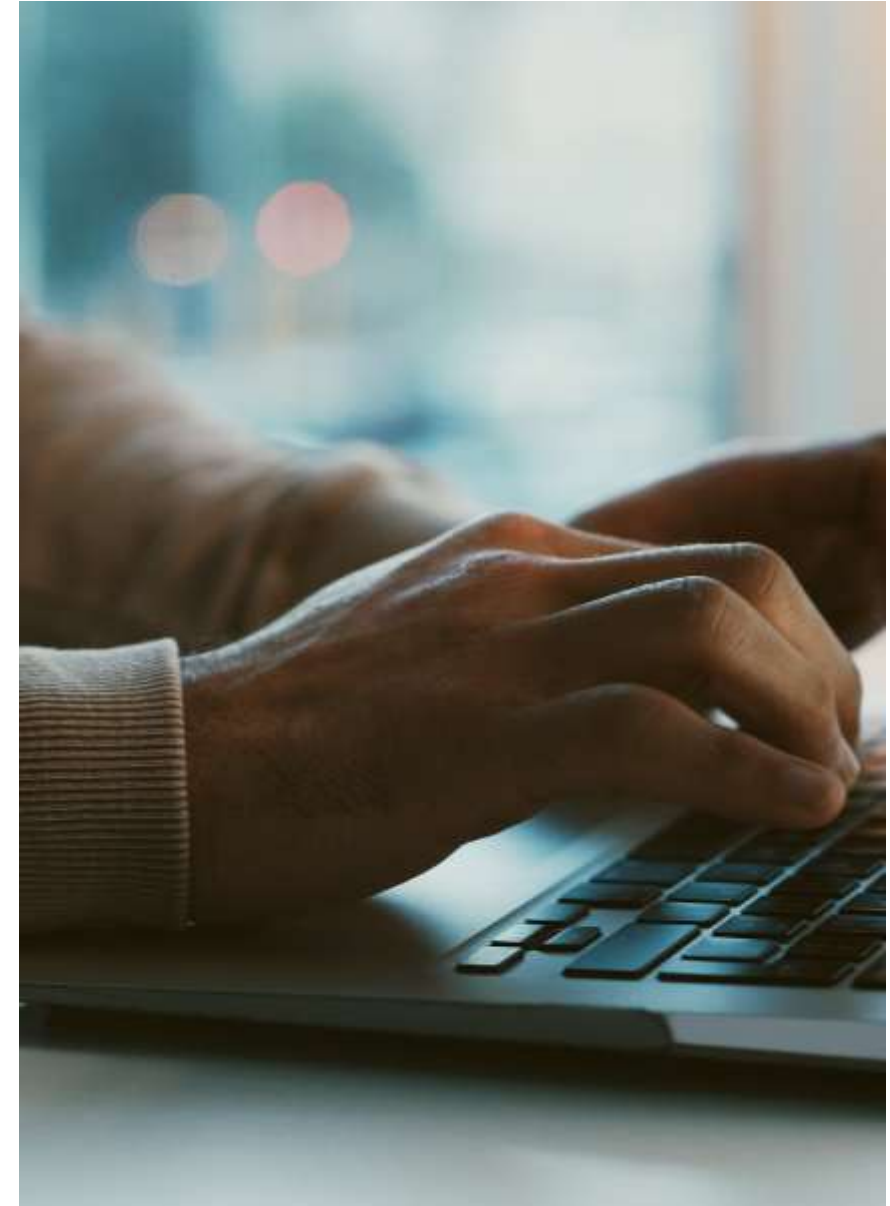
```
    cin >> a >> b;
```

```
    p = a * b;
```

```
    cout << p;
```

```
    return 0;
```

```
}
```



Tipovi (vrste) podataka.^{kmb} Što je podatak i kako ga program pamti?

17

17.5

'A'

true

Koji tip?

int, double, char, bool...

varijabla,

identifikator (ime)

deklaracija



smisleni i loši nazivi

tip ovisi o vrsti vrijednosti

**naziv treba opisivati značenje
podatka**

Koji tip podatka?

kmb

14

tip povezati sa značenjem i mogućim vrijednostima podatka

Podatak	Prikladan tip	Zašto?
broj učenika u razredu	int	cijeli broj
temperatura 21.7 °C	double	može sadržavati decimalni dio
slovo razreda 'A'	char	jedan znak
je li učenik prisutan	bool	dvije logičke vrijednosti: true/false
cijena proizvoda	?	?



Dobar ili loš identifikator?

kmb

15

x
brojUcenika
2broj
cijena_ulaznice
int
prosjecnaOcjena

razlikovati na osnovu tri pitanja:

- (1) je li naziv sintaktički dopušten,**
- (2) je li riječ rezervirana/ključna,**
- (3) je li naziv razumljiv.**

Primjer 'x' može biti dopušten, ali često **nije dovoljno informativan**;



Varijabla, konstanta, deklaracija i inicijalizacija

```
int broj;  
int broj = 5;
```

```
const double PI = 3.14159;
```

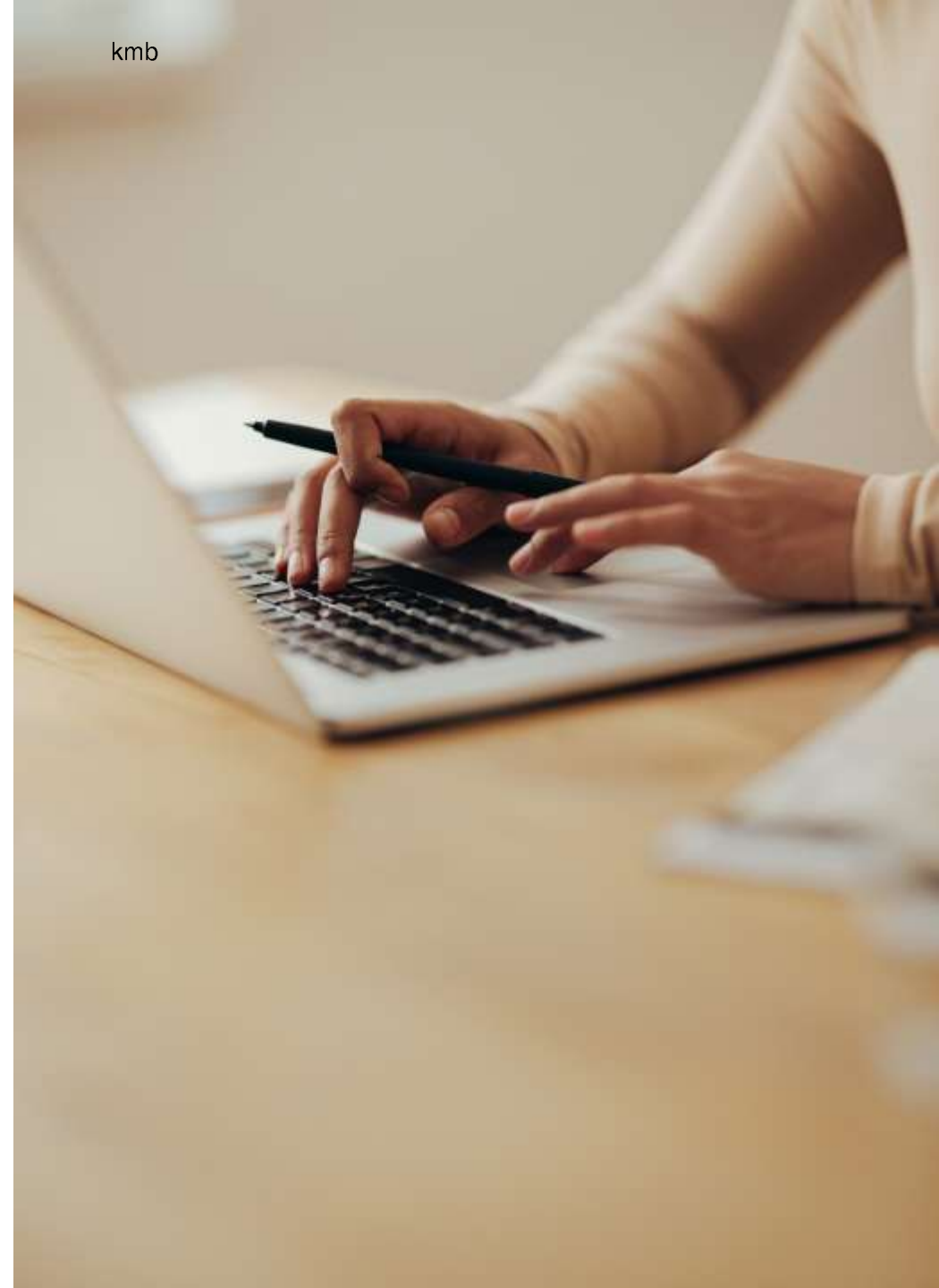
što se može mijenjati, a što ne?

komentar // objašnjenje namijenjeno čovjeku

deklaracija i inicijalizacija?

zašto neke vrijednosti treba zaštititi kao konstante?

gdje komentar može pomoći razumljivosti?





Naredba pridruživanja '=' nije matematička jednakost

```
int x = 5;
```

```
x = 8;
```

```
x = x + 2;
```

kolika je nova vrijednost x?

zašto je $x = x + 2$ smisleno u programu,
iako nije matematička jednakost?

Tablica praćenja: korak po korak prati
stanje varijable

**Najprije se izračuna izraz s desne
strane znaka =, a zatim rezultat
pridruži varijabli lijevo**

Odgovori!

Algoritamska uloga	C++ zapis	Pitanje
pohrana podataka	<code>double a, b, p;</code>	Zašto ovdje bирамо double, a ne int?
ULAZ	<code>cin >> a >> b;</code>	Koje vrijednosti program još ne zna prije ove naredbe?
OBRADA	<code>p = a * b;</code>	Što se prvo izračunava, a gdje se zatim rezultat sprema?
IZLAZ	<code>cout << p;</code>	Što korisnik nakon ove naredbe može vidjeti?

Mini izazov: sastavi model programa^{kmb}

Škola kupuje ulaznice za muzej. Poznati su broj učenika i cijena jedne ulaznice. Potrebno je izračunati i ispisati ukupnu cijenu.

- odredi tipove (obrazloži)
- odredi varijable
- ulaz
- navedi naredbu obrade
- izlaz

ULAZ	
Tipovi	
OBRADA	
Tip rezultata	
IZLAZ	

- -> rješenje,
- bi li 'brojUcenika' mogao biti double i bi li program tehnički radio?

Upute:

- individualno, zatim u paru
- odredi podatke i tipove
- zapiši ključne naredbe

PROBLEM → ALGORITAM → PROGRAM → PROVJERA

ULAZ → OBRADA → IZLAZ

ALGORITAM govori ŠTO i kojim redoslijedom

PROGRAM to zapisuje pravilima programskog jezika.

PROGRAM je **zapis algoritma pravilima programskog jezika.**

Ključne riječi (npr. int, double, const, return) imaju unaprijed određeno značenje i zato ih ne možemo proizvoljno koristiti kao nazive varijabli.

Osnovni tipovi: int – cijeli broj; double – realni broj; char – znak; bool – logička vrijednost.

Varijabla ima naziv, tip i vrijednost koja se tijekom izvođenja **može mijenjati.**

Konstanta (const) je imenovana vrijednost koju program **nakon definiranja ne mijenja.**

Naredba pridruživanja: najprije se izračuna desna strana, zatim se vrijednost sprema lijevo.
cin → unos; cout → ispis; // → komentar.

3.sat



Kako program računa i provjerava izraze?

kako C++ izračunava izraze?

kako se vrijednosti različitih tipova ponašaju u izrazima?

objašnjavanje i pravilan odabir operatora

PREDVIDI → OBRAZLOŽI → PROVJERI ZAJEDNO

VRIJEDNOSTI + OPERATORI → IZRAZ → REZULTAT

ULAZ → OBRADA (izrazi) → IZLAZ



- prepoznati izraz i predvidjeti njegov rezultat za zadane vrijednosti varijabli;
- koristiti aritmetičke operatore +, -, *, / i % u jednostavnim izrazima;
- objasniti razliku između cjelobrojnog i realnog dijeljenja na konkretnom primjeru;
- primijeniti zagrade i osnovna pravila prioriteta operatora;
- prepoznati i koristiti operatore usporedbe ==, !=, <, >, <= i >=;
- prepoznati i koristiti logičke operatore &&, || i ! u jednostavnim logičkim izrazima;
- razlikovati unarni i binarni operator na primjeru, a ternarni operator prepoznati kao sažeti zapis izraza s izborom vrijednosti;
- uočiti kada je potrebna jednostavna konverzija tipa;
- upotrijebiti sqrt(), pow() i abs() u jednostavnom matematičkom izrazu;
- povezati tip podatka, izraz i rezultat s linearnim tijekom ULAZ → OBRADA → IZLAZ.

Što će se ispisati?

```
int a = 7;
int b = 2;
```

```
cout << a + b << endl;
cout << a * b << endl;
cout << a - b << endl;
```

zapisati očekivane rezultate

```
cout << a / b << endl;
cout << a % b << endl;
```

Što bi računalo trebalo napraviti s dva cijela broja 7 i 2?

```
7 / 2    // cjelobrojno dijeljenje
7.0 / 2  // realno dijeljenje
(double)7 / 2
```

Konverzija - praktičan način da računanje izvedemo u željenom tipu



Prioritet operatora i zagrade

kmb

Izraz	Predviđeni rezultat	Zašto?
$2 + 3 * 4$		* prije +
$(2 + 3) * 4$		zagrade mijenjaju redoslijed
$10 - 6 / 2$		/ prije -
$(10 - 6) / 2$		najprije zagrada

zagrade → množenje/dijeljenje/ostatak → zbrajanje/oduzimanje

zagrade služe i jasnoći, ne samo promjeni redoslijeda.



Operatori usporedbe

Za int a=7 i b=2 predvidjeti rezultat:

a > b

a < b

a == 7

a != b

a <= 5

a >= 7

izračunavamo logičku vrijednost izraza

= znači PRIDRUŽI VRIJEDNOST,

== znači USPOREDI JESU LI VRIJEDNOSTI JEDNAKE.

Rezultat usporedbe je bool vrijednost true ili false.



Logički operatori

kmb

Tvrdnja	C++ izraz	Primjer provjere
x je pozitivan i manji od 10	$x > 0 \ \&\& \ x < 10$	$x=6 \rightarrow \text{true}$
x je manji od 0 ili veći od 100	$x < 0 \ \ x > 100$	$x=120 \rightarrow \text{true}$
x nije jednak 5	$!(x == 5)$	$x=3 \rightarrow \text{true}$

Oblik i izračun logičkih izraza

kratka tvrdnja u logički izraz i obrnuto;
Predvidjeti rezultat za konkretne vrijednosti.





Unarni, binarni i ternarni operatori

$-x$ // jedan operand $\rightarrow$ unarni
 $!uvjet$ // jedan operand $\rightarrow$ unarni
 $a + b$ // dva operanda $\rightarrow$ binarni
 $a > b$ // dva operanda $\rightarrow$ binarni

$(a > b) ? a : b$ // tri operanda $\rightarrow$ ternarni, izraz koji vraća jednu od dvije vrijednosti.

Matematičke funkcije

```
sqrt(25)    // 5
```

```
pow(3, 2)   // 9
```

```
abs(-7)     // 7
```

funkciji predajemo vrijednost (argument), ona vraća rezultat koji možemo koristiti u izrazu.

```
zaglavlje <cmath>
```

Sažetak

IZRAZ kombinira vrijednosti/varijable i operatore te daje rezultat.

Aritmetički operatori: + - * / %

Usporedbe: == != < > <= >= → rezultat je true ili false

Logički operatori: && (i), || (ili), ! (ne)

Kod int / int rezultat je cijeli broj; za realno dijeljenje barem jedan operand treba biti realan.

Zagrade imaju prednost kod izračunavanja i mogu učiniti izraz jasnijim.

Primjeri matematičkih funkcija: sqrt(x), pow(x,y), abs(x).