

mr. sc. Kristinka Maček Blažeka, mag. ing. el.



UVOD U PROGRAMIRANJE

KAKO MOGU RAZMIŠLJANJEM
RIJEŠITI PROBLEM, A ZATIM
RAČUNALU DATI PRECIZNE UPUTE
DA GA RIJEŠI UMJESTO MENE

POČETI RAZMIŠLJATI O
PROGRAMIRANJU KAO VJEŠTINI
RJEŠAVANJA PROBLEMA.

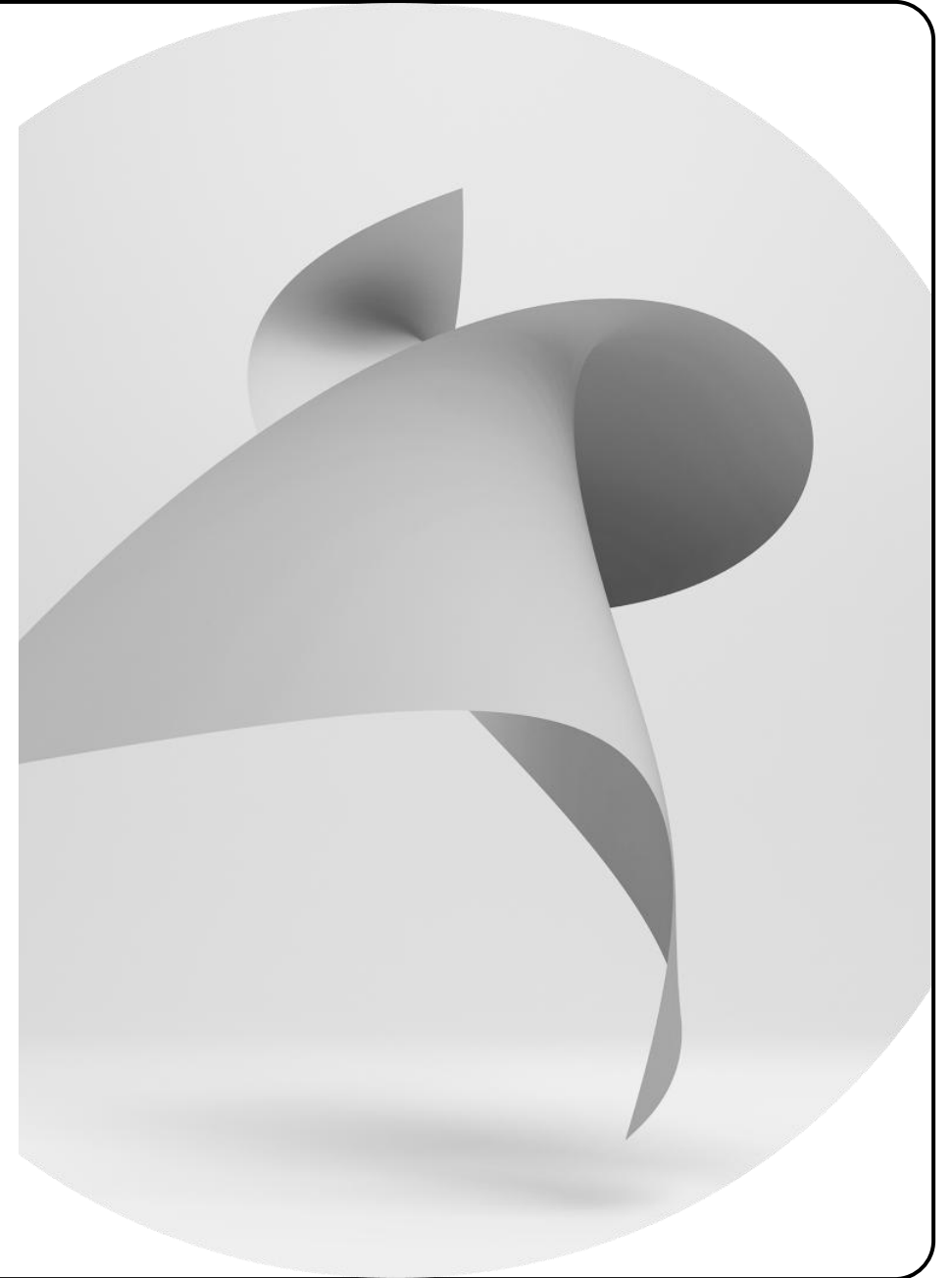
AI može napisati kôd. Čovjek mora razumjeti problem, odrediti što je ispravno, provjeriti rezultat i preuzeti odgovornost za rješenje.

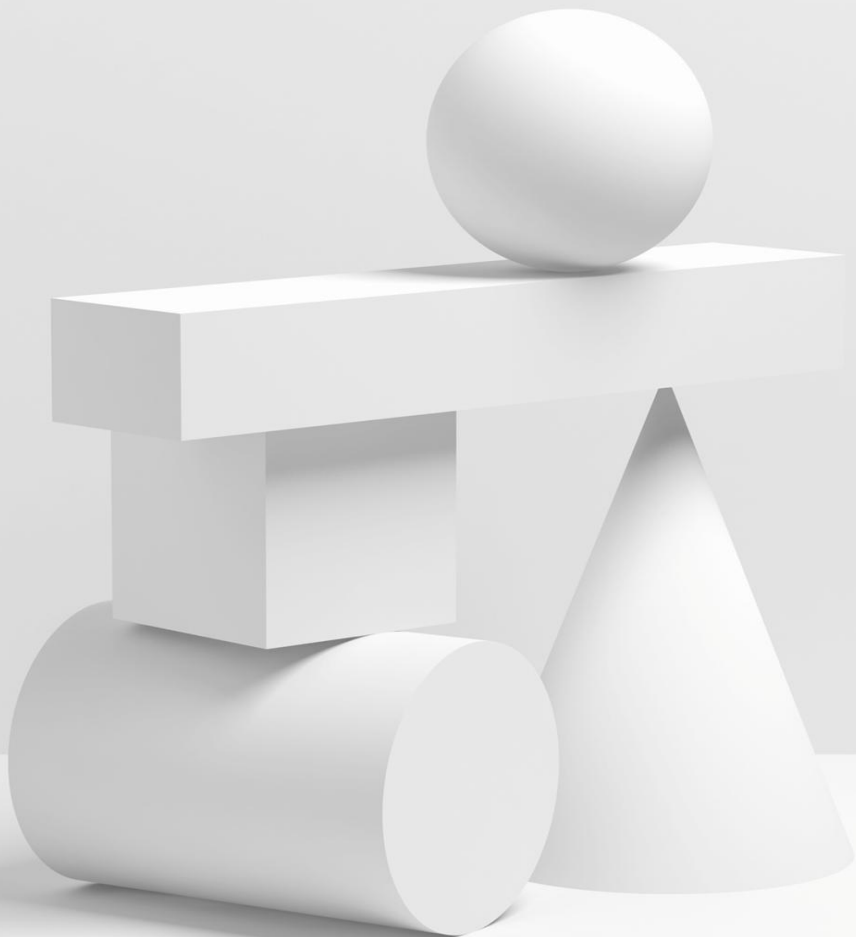
AI može riješiti složene zadatke i upravo zato što može, razumijevanje algoritama i sposobnost provjere postaju još važniji.



Od prvog algoritma do programa koji rješava problem

- razumjeti problem i osmisлити algoritam
- zapisati algoritam i pratiti njegovo izvođenje
- koristiti osnovne podatke, izraze, grananje i petlje
- pisati programe u C++
- obrađivati tekst i skupove podataka
- koristiti spremničke tipove podataka
- testirati program, pronalaziti pogreške i poboljšavati rješenje.





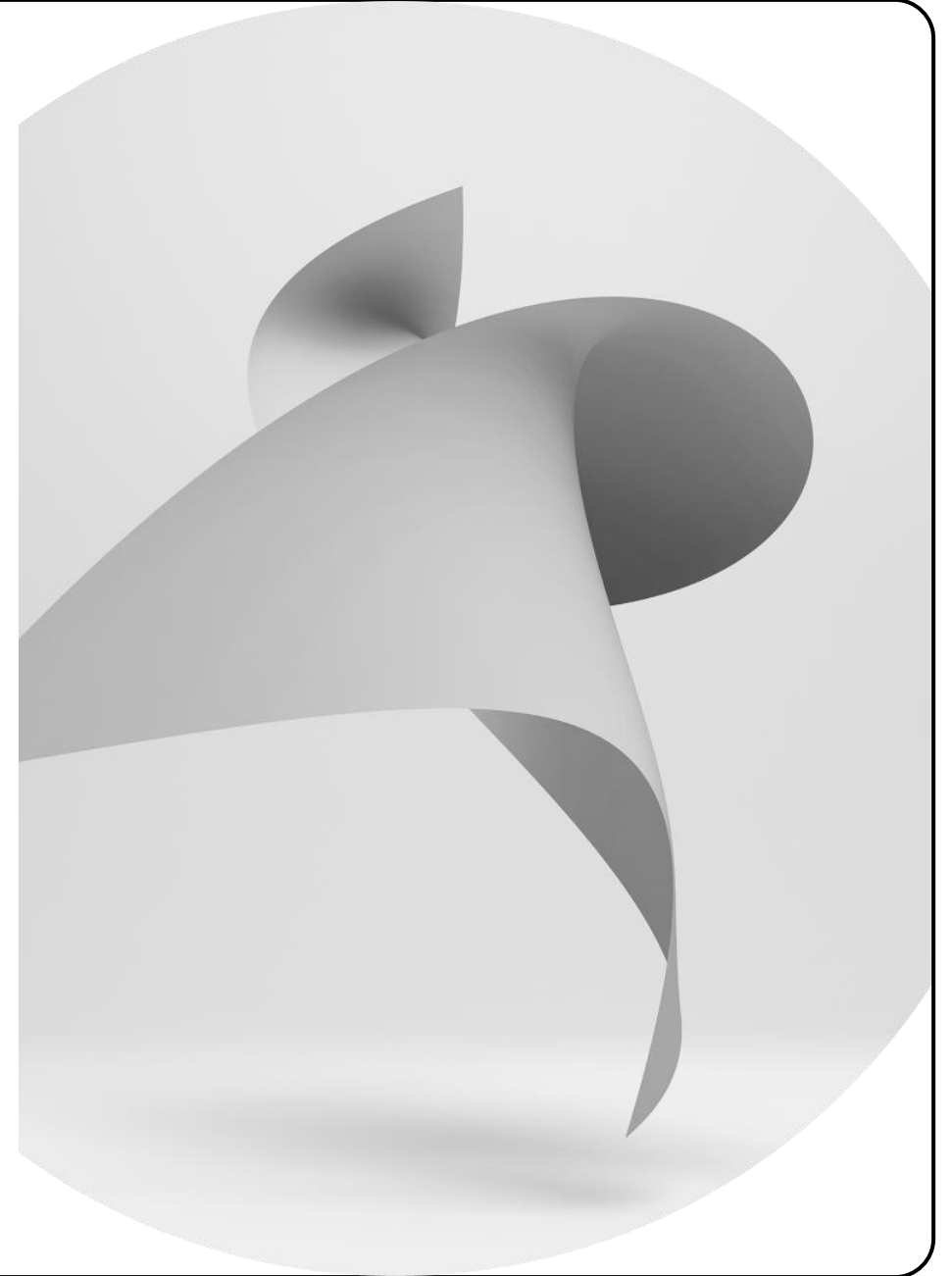
c++ omogućuje da
razvijamo algoritamsko
razmišljanje,
upoznamo temeljne programske
koncepte
koristimo snažnu standardnu
biblioteku za rješavanje složenijih
problema.

1. Učimo temeljne koncepte. Varijable, uvjeti, petlje, funkcije i strukture podataka postoje u mnogim jezicima. Što naučite neće vrijediti samo za C++.

2. C++ se koristi u različitim područjima. Od sustavskog i ugrađenog programiranja do igara, simulacija i računalno zahtjevnih aplikacija.

3. Omogućuje nam rad sa standardnom bibliotekom. Nećemo sve izrađivati od početka; koristit ćemo gotove, provjerene strukture poput vector, stack, queue i list.

4. Dobar je temelj za daljnje učenje. Kasnije ćete lakše razumjeti druge jezike i tehnologije jer ćete već poznavati temeljne koncepte.



KAKO ĆEMO UČITI I GDJE SU MATERIJALI?

- mrežno sjedište, UČklik

Uvod u programiranje → odabir aktivnosti → sadržaji s nastave / praktične vježbe / samostalne aktivnosti učenja.

- tri oblika rada:

Vođeni proces učenja i poučavanja: zajedno otkrivamo i objašnjavamo nove koncepte, analiziramo primjere i oblikujemo algoritme.

Učenje temeljeno na radu: rješavamo praktične zadatke, izrađujemo programe, testiramo ih i popravljamo.

Samostalne aktivnosti učenja: ponavljamo, istražujemo, rješavamo dodatne zadatke i pripremamo se za sljedeći rad.



UDŽBENIK

C++, Školska knjiga

Uvod u programiranje, Element

Udžbenik nam je jedan od izvora za učenje, ali nećemo ga prolaziti stranicu po stranicu. Koristit ćemo ga za objašnjenja, primjere i ponavljanje, a zadatke i dodatne materijale nalazit ćete i na mrežnom sjedištu.

KAKO ĆEMO RADITI I KAKO ĆE SE VREDNOVATI?

Razumijem problem → osmišljavam algoritam → pišem program → testiram → pronalazim pogrešku → popravljam → objašnjavam.

- očekivanja:

redovito sudjelovanje i izrada obveznih vježbi

čuvanje vlastitih programa i bilježaka

pokušaj samostalnog rješavanja prije traženja gotovog rješenja

suradnja i odgovorno korištenje AI-ja

sposobnost objašnjavanja vlastitog rada.

KRITERIJI OCJENJIVANJA

-ne vrednuje se samo točan ispis, nego i razumijevanje, postupak, primjena znanja i objašnjenje rješenja.

Razina	Što učenik pokazuje
2	Rješava osnovne probleme uz određenu podršku.
3	Samostalno rješava poznate probleme.
4	Povezuje znanja i rješava zahtjevnije probleme.
5	Samostalno oblikuje, obrazlaže, testira i poboljšava složenija rješenja.

PROGRAMIRAJ PRIJATELJA

P>			
		C	

Zadatak:

Napišite upute kojima ćete prijatelja dovesti od početka do cilja. Smijete koristiti samo: KORAK NAPRIJED, OKRENI LIJEVO, OKRENI DESNO.

računalo ne razumije našu namjeru → upute moraju biti precizne → redoslijed je važan → algoritam možemo testirati.

„TAJNI BROJ“

- osmisli postupak
- Zašto je pitanje „Je li veći od 50?“ korisno?
Koliko mogućnosti ostaje nakon odgovora?
Što biste pitali sljedeće?
Možemo li uvijek pronaći broj u najviše sedam pitanja?
- Odgovor je da, jer je $2^7 = 128$, pa sedam binarnih odluka može razlikovati 100 mogućnosti.
- Dobar algoritam ne mora samo dati točan rezultat. Može ga dati i mnogo učinkovitije

ALGORITAM: „PRONAĐI NAJVEĆI BROJ“ [M]

7, 12, 4, 19, 8, 15 ili -8, -3, -12, -5

TABLICA PRAĆENJA POSTUPKA

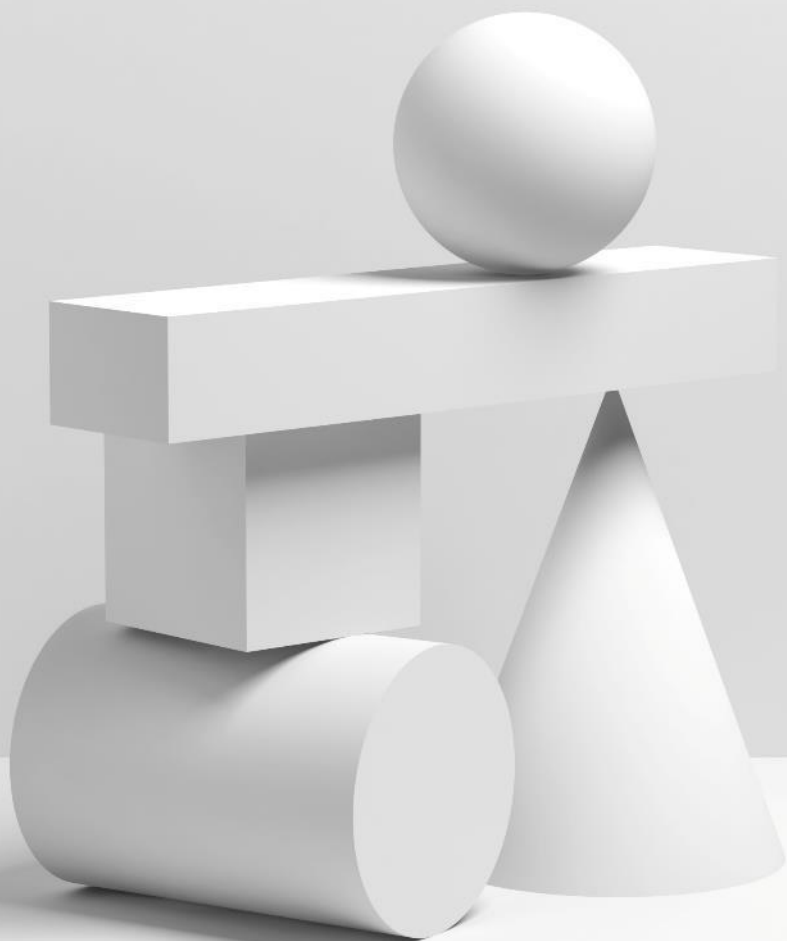
Korak	Trenutni broj	Najveći do sada
Početak	7	7
2	12	12
3	4	12
4	19	19
5		
6		

KAKO ĆEMO KORISTITI AI?

- Prvi učenik: „Napiši mi rješenje zadatka za najveći broj.”
- Drugi učenik: „Osmislio sam postupak, ali ne znam radi li kada su svi brojevi negativni. Možeš li mi postaviti pitanja koja će mi pomoći pronaći pogrešku, bez davanja gotovog rješenja?”
- Pitanje: Koji učenik će nakon deset takvih zadataka vjerojatno bolje znati programirati? Zašto?
- Zaključak:
- AI ćemo koristiti kao pomoćnika za učenje, objašnjavanje, testiranje i istraživanje. Ali sposobnost samostalnog razmišljanja i razumijevanja vlastitog rješenja ostaje naš cilj.
- uporaba AI nije sama po sebi nepoštena – pomaže učiti, a ne oduzima priliku da se nauči.

ZAKLJUČNO

- Ne očekujem da svi danas znate programirati. Očekujem da ćete pokušavati, razmišljati, postavljati pitanja, učiti iz pogrešaka i postupno postajati samostalniji.
- Odaberi jedan svakodnevni postupak koji bi se mogao opisati algoritmom — primjerice priprema sendviča, pronalaženje najbržeg puta do škole, razvrstavanje knjiga ili organizacija reda čekanja, izvršavanje dnevnih zadataka.... Napiši 5–10 preciznih koraka i razmisli gdje se pojavljuju odluka ili ponavljanje (u bilježnici).
- prvo razumij problem, zatim osmisli postupak, a tek onda piši kôd; prije pokretanja predvidi rezultat; nakon pogreške traži uzrok, a ne samo gotov ispravak.



Programiranje je
rješavanje problema,
pogreške su dio učenja,
napredak je važniji od
početnog predznanja,
AI je alat koji nam
pomaže postati
sposobniji – ne
zamjena za naše
razmišljanje