

Stvaranje programa

faze programiranja



Rješavanje problemskog zadatka

Čovjek:

- znanje,
- pamćenje,
- iskustvo,
- logičko rasuđivanje,
- intuicija,
- osjećaji, itd.

Računalo:

- pamćenje,
- logičko rasuđivanje.

2

Računalo rješava zadatak

- Da bi računalo riješilo problemski zadatak, treba mu dati **upute** u obliku koji uključuje samo **sposobnosti pamćenja** i **logičkog rasuđivanja**.
- **Postupak** rješavanja problemskog zadatka računalom je **složen** i provodi se kroz **niz koraka** koji se nazivaju **faze programiranja**.

3

Faze programiranja

- planiranje
- analiza zadatka
- stvaranje algoritma
- pisanje pseudokôda
- crtanje dijagrama tijeka
- zapis algoritma programskim jezikom
- provjera i ispravljanje
- održavanje
- stvaranje dokumentacije

4

- Primjer – problemski zadatak: **Pješak**



Pješak u zadanom vremenu učini određeni broj koraka. Potrebno je odrediti:

Brzinu pješaka u km/h i m/s ako je poznata prosječna duljina njegovog koraka.

5

Planiranje

- Prvi korak - **predviđanje** i **raspored** pojedinih **faza izrade** programa.
- Određuje se **tko** će, **kada** i **što** raditi.

6

Analiza zadatka i specifikacija

- Analiza je **raščlamba** i potpuno **razumijevanje zadatka** i traženih **rezultata**.
- Rezultat analize je specifikacija, **dokument** koji sadrži detaljan **opis zadatka** i **željenih rezultata**.
- To nije prijedlog kako riješiti zadatak!

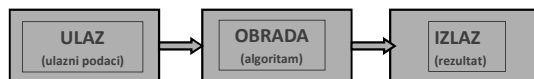
7

Zadano:	oznaka
vrijeme hodanja (jedinica nije zadana; treba provjeriti što je korisnik unio, pa prema potrebi pretvoriti jednu jedinicu u drugu)	t
duljina jednog koraka (jedinica nije zadana; treba provjeriti što je korisnik unio, pa prema potrebi pretvoriti jednu jedinicu u drugu)	sk
broj koraka	N
Treba izračunati:	oznaka
brzina kretanja pješaka u m/s	v1
brzina kretanja pješaka u km/h	v2

8

Algoritam

- To je **naputak** kako riješiti zadatak.
- Sačinjen je od **niza elementarnih radnji** poredanih **u ispravnom redosljedu**.



9

- Unijeti poznate ulazne podatke.
- Provjeriti u kojim su jedinicama uneseni ulazni podaci.
- Ovisno o jedinicama mjere ulaznih podataka odlučiti treba li iznose preračunati u vrijednosti izražene u jedinicama osnovnog SI sustava.
- Izračunati put na osnovu ukupnog broja koraka i prosječne duljine jednog koraka.
- Pomoću formule za računanje brzine jednolikog gibanja i podatka o putu i vremenu izračunati brzinu v1 izraženu u m/s.
- Preračunati vrijednost iz m/s u km/h.
- Ispisati rezultate (izlazne vrijednosti).

10

Algoritam

- Većina zadataka se može riješiti na više različitih načina, no treba nastojati pronaći **što brži, što učinkovitiji i što sigurniji** algoritam.
- Algoritam se može prikazati:
 - dijagramom tijeka,
 - pseudokôdom.

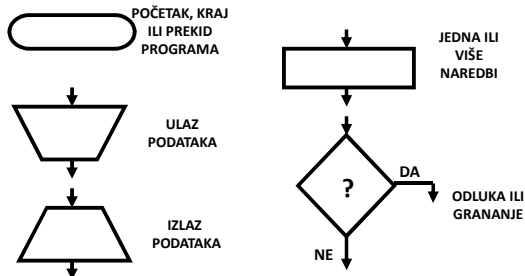
11

Dijagram tijeka

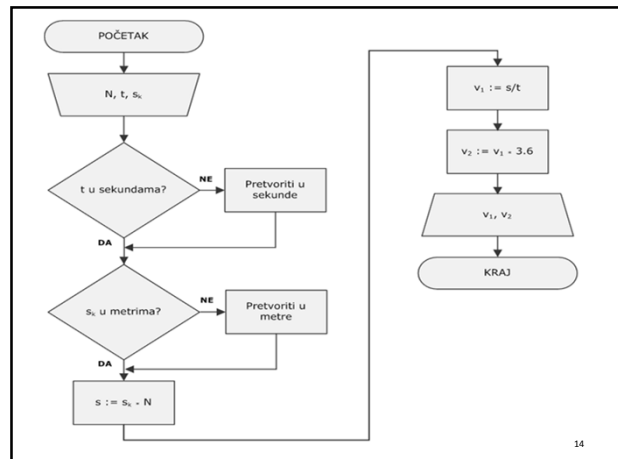
- To je **grafički prikaz** algoritma.
- Sastoji od nekoliko jednostavnih geometrijskih likova spojenih usmjerenim crtama koje pokazuju tijek rješavanja zadatka.
- Omogućava lakšu **analizu i provjeru** predloženog rješenja.

12

Geometrijski likovi dijagrama tijeka



13



14

Pseudokôd

- Pseudokôd je **tobožnji program** (grč. pseudos – laž).
- Sastoji se od kratkih izraza **na govornom jeziku** koji opisuju i objašnjavaju pojedine dijelove algoritma.
- Treba biti napisan tako da programer može na temelju njega napisati program u bilo kojem programskom jeziku.

15

ulaz (N);

ulaz (t);

ako je t izražen u jedinicama koje nisu sekunde onda pretvori t u sekunde ;

ulaz (sk);

ako je sk izražen u jedinicama koje nisu metri onda pretvori sk u metre;

s := sk * N;

v1 := s / t;

v2 := v1 * 3,6; (1m=0,001km; 1s=1/3600h)

izlaz (v1,v2);

16

Programski jezik

- To je **umjetni jezik** posebno osmišljen **za komunikaciju s računalom**.
- Svaki programski jezik određuje:
 - ograničeni skup riječi posebnog značenja (**ključne riječi**),
 - propisana pravila slaganja ključnih riječi u naredbe (**sintaksa**).

17

Programiranje

- Zapis** algoritma **naredbama** odabranog **programskog jezika**.

```

1 #include<iostream>
2 using namespace std;
3 int main ()
4 {
5     float t,N,sk,s,v1,v2;
6     cout<<"Upisi vrijeme kretanja: ";
7     cin>>t;
8     cout<<"Upisi prosječnu duljinu jednog koraka: ";
9     cin>>sk;
10    cout<<"Upisi broj koraka: ";
11    cin>>N;
12    s=sk*N;
13    v1=s/t;
14    v2=v1*3.6;
15    cout<<"Brzina pjesaka bila je "<<v1<<" m/s, to jest "<<v2<<"km/h";
16    return 0;
17 }
18

```

Primjer vrijedi samo za situaciju unosa vrijednosti u osnovnim SI jedinicama!

18

provjera i ispravljanje

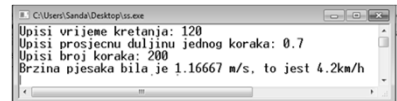
- Da bi se program mogao **izvršiti**, mora biti **pravopisno ispravan**, tj. mora biti zadovoljena sintaksa.
- Da bi program bio uporabno koristan, mora biti **logički ispravan**, tj. potrebno ga je **provjeravati** (testirati).
- Program provjerava **autor** programa, više ljudi kod proizvođača programa ili **neovisni ispitivači**.

19

provjera

Zadano:
t = 2min sk = 70 cm N = 200
Očekivano:
V1 = 1,17 m/s V2 = 4,2 km/h

- Postupak treba ponoviti za **što više ulaznih podataka** za koje je poznat rezultat ili se rezultat zna izračunati.



20

Održavanje programa

- Postupak **mijenjanja** programa tijekom njegovog "životnog" vijeka.
 - **Izravno**: ugovor o održavanju.
 - **Neizravno**: izdavanje novih inačica i ispravaka programa za programe koji se prodaju u velikim količinama (npr. program Windows).

21

Dokumentacija

- To je važan **dodatak** programu. Može sadržavati:
 - **upute za instaliranje** programa,
 - **priručnik** za korisnike,
 - **tehnički opis** programa, itd.

22