

KAZALO VSEBINE

1	PROGRAM IN PROGRAMIRANJE	6
1.1	Nivoji programiranja.....	9
1.2	Do strojnega zapisa: prevajanje in tolmačenje	14
1.3	Strategije pri gradnji programov	15
1.4	Napake v programih	16
2	ALGORITEM	19
2.1	Kaj je algoritem	19
2.2	Lastnosti algoritmov	19
2.3	Delitve algoritmov	23
2.4	Diagram poteka.....	23
2.5	Sled algoritma	25
3	PROGRAMSKI JEZIK	26
3.1	Programski jezik Java	27
3.2	Moj prvi program v Javi	29
4	ELEMENTI PROGRAMSKEGA JEZIKA JAVA	31
4.1	Komentarji v programskem jeziku Java	31
4.2	Vhod in izhod podatkov	32
4.3	Tipi podatkov v Javi	34
4.4	Spremenljivke in konstante.....	42
4.5	Operacije in operatorji v Javi.....	44
4.6	Naključna števila.....	52
5	KRMILJENJE TOKA PROGRAMA.....	53
5.2	Pogojni stavek	55
5.3	Polni pogojni stavek: <i>if ... else</i>	55
5.4	Stavek (kretnica): <i>switch ... case</i>	59
5.5	Programska zanka <i>while</i> : "Dokler je izpolnjen pogoj, ponavljaj ... "	60
5.6	Zanka <i>do... while</i> : "Ponavljaj ... če/dokler je izpolnjen pogoj."	61
5.7	Zanka <i>for</i>	63
6	POLJA SPREMENLJIVK (TABELE) V JAVI.....	66
6.1	Deklaracija in ustvarjanje polja spremenljivk.....	67
6.2	Delo s polji spremenljivk	68
6.3	Dvodimenzijska polja spremenljivk (2D-tabele).....	69
6.4	Delo z 2D polji spremenljivk.....	70
7	OSNOVE OBJEKTNO USMERJENEGA PROGRAMIRANJA v Javi.....	74
7.1	Prednosti objektno orientiranega programiranja	74
7.2	Objekt in razred.....	76
7.3	Konstruktorji	78
7.4	Več konstruktorjev v istem razredu	81
7.5	Oznaka <i>this</i>	83
7.6	Objektne in statične spremenljivke.....	84
7.7	Metode	87
7.8	Uporaba že pripravljenih <i>objektnih</i> metod: razred <i>String</i>	92
7.9	Uporaba že pripravljenih <i>statičnih</i> metod: razred <i>Math</i>	94
7.10	Prekrivanje (preoblaganje) metod	95
7.11	Dedovanje	99
7.12	Povoženje metod.....	104
7.13	Abstraktni razredi	107

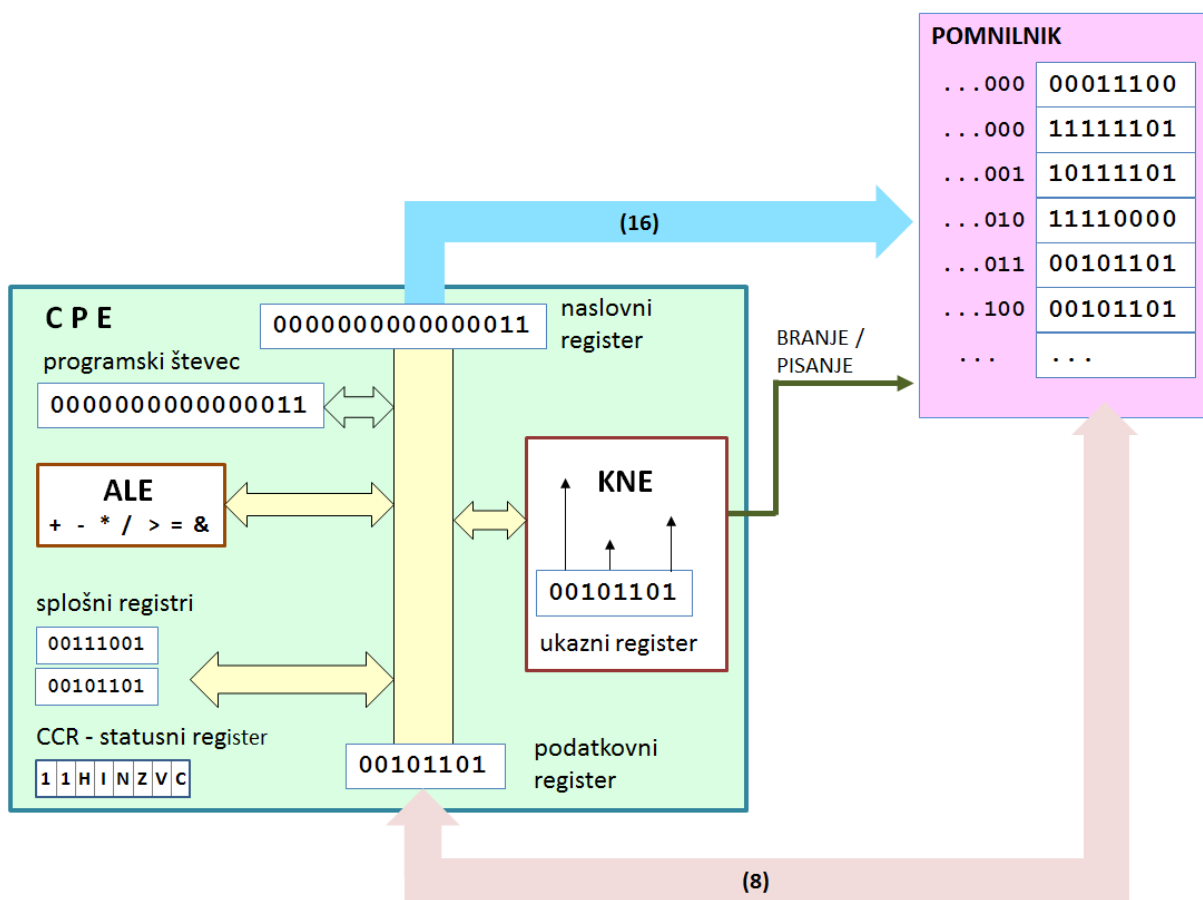
7.14	Abstraktne metode.....	107
7.15	Vmesniki.....	109
7.16	Področja programa in dostopnost spremenljivk.....	110
7.17	Oznake dostopnosti.....	113
7.18	Omogočanje dostopa do zasebnih komponent.....	114
8	DELO Z NIZI ZNAKOV.....	117
8.1	Delo z nizi znakov in razred <i>String</i>	117
8.2	Delo z nizi znakov in razred <i>StringBuffer</i>	121
8.3	Delo z nizi znakov in razred <i>StringTokenizer</i>	123
9	PRESTREZANJE IZJEM.....	124
9.1	Izjeme.....	124
9.2	<i>Try, catch, finally</i>	125
9.3	<i>Throws</i>	128
10	VHOD IN IZHOD PODATKOV.....	129
10.1	Tokovi podatkov.....	130
10.2	Ovijanje tokov.....	131
11	DATOTEKE.....	133
11.1	Osnovno o datotekah.....	133
11.2	Vrste datotek v Javi.....	134
11.3	Delo z binarnimi datotekami.....	135
11.4	Uporaba datotek z naključnim dostopom.....	138
11.5	Delo z besedilnimi datotekami.....	140
11.6	Splošna dejanja z datotekami.....	143
12	PROGRAMI Z GRAFIČNIM UPORABNIŠKIM VMESNIKOM.....	145
12.2	Gradniki grafičnega vmesnika.....	148
12.3	Okvir (<i>JFrame</i>).....	149
12.4	Dokumentacija razredov.....	151
12.5	Gumb (<i>JButton</i>).....	152
12.6	Postavljanje gradnikov na delovno površino.....	153
13	DOGODKI GRADNIKOV IN ODZIVI NANJE.....	156
13.1	Obravnava dogodkov v Javi.....	156
14	ŠE NEKAJ GRADNIKOV GRAFIČNIH VMESNIKOV V JAVI.....	163
14.1	Napis (<i>JLabel</i>).....	163
14.2	Vnosno polje za besedilo (<i>TextField</i>).....	165
14.3	Pogovorna okna (<i>JOptionPane</i>).....	168
14.4	Potrditveno polje (<i>CheckBox</i>).....	171
15	DODATKI.....	174
15.1	Pretvorba diagrama poteka v programsko kodo.....	174
15.2	Pseudokoda algoritma.....	182
15.3	Rekurzija in rekurzivne metode.....	183
15.4	Izdelava dokumentacije razredov z <i>Javadoc</i>	186
15.5	Neposreden zagon javanskih programov iz okolja Windows.....	190
15.6	Grafično razvojno okolje <i>NetBeans</i>	194
15.7	Motiti se je pričakovano: napake pri programiranju.....	196
15.8	Obvestila o napakah: preglednica obvestil o napakah.....	197

1 PROGRAM IN PROGRAMIRANJE

Besedo *program*¹ pozna danes že povprečen otrok, zato si za naše potrebe zapišimo kar definicijo:

- **Program** je urejen seznam ukazov za izvedbo operacij, ki rešujejo določeno nalogo.
- **Računalniški program** je seznam ukazov mikroprocesorju.

V mikroprocesorju se podatki vpisujejo in spreminjajo v **registrih**², nad njimi izvaja računske ali logične operacije **aritmetično-logična enota**, med registri pa se podatki prenašajo po notranjih vodilih. Glede na funkcijo binarne vsebine, ki se prenaša po teh vodilih, razlikujemo **naslovno, podatkovno in krmilno vodilo**. Blokovna shema poenostavljene zgradbe mikroprocesorja je prikazana na sliki.



Slika 1: Zgradba centralne procesne enota (CPE) in povezava z delovnim pomnilnikom.

Ukazi, ki bodo določali delovanje mikroprocesorja, so shranjeni zunaj mikroprocesorja, v **pomnilniku**. Od tam jih procesor prevzema in izvaja. Zaključeno zaporedje takšnih ukazov je **program**. Kako torej "izgleda" posamezen ukaz?

Mikroprocesor je digitalna naprava. To pomeni, da lahko sprejme le binarno kodirane podatke in vrača binarno kodirane rezultate. Za binarne podatke je značilno, da so predstavljeni z določenim

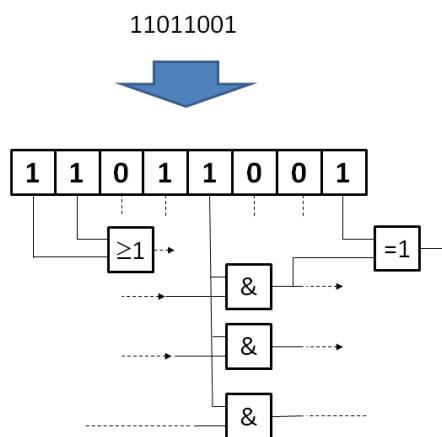
¹ programi pralnega stroja, (telo)vadbeni program, računalniški program ...

² Registri so bralno pisalne pomnilniške enote, običajno velikosti do nekaj zlogov (8-, 16-, 32- ...-bitni registri).

zaporedjem binarnih stanj. Binarni stanji sta lahko predstavljeni bodisi z dvema različnima električnima napetostima (npr. 0 V in 5 V), dvema načinoma namagnetjenja (pri magnetnih nosilcih zapisa), z dvema velikostima luknjice (optični nosilci). Tudi rezultati računskih in drugih operacij so zapisani v binarni obliki – vsak bit rezultata ima eno od samo dveh možnih stanj. Osnovna enota binarnega zapisa je *bit*, za praktično uporabo pa se ja za osnovno ustalil bajt, ki je zaporedje 8 bitov. Podatki so v računalniku zapisani z enim ali več bajti.

Obdelava podatkov v procesorju pa je preklapljanje številnih binarnih stanj po pravilih binarne logike in binarne aritmetike. Tako mikroprocesor z uporabo vgrajenih logičnih in aritmetičnih digitalnih vezij obdeluje podatke in "pripravlja" rezultate logičnih in aritmetičnih operacij. Omenjena vezja so del mikroprocesorja, po funkciji pa so to različna logična vrata, seštevalniki, pomikalni registri in druga temeljna digitalna vezja.

Kateri elektronski podsklopi mikroprocesorja bodo sodelovali pri izvedbi določenega ukaza, npr. pri seštevanju dveh števil? Za vsakega od njih bi morali določiti, ali je pri določenem ukazu uporabljen ali ne. Zaporedje enk in ničel, ki to določa, je že del ukaza. **Zaporedje binarnih vrednosti** ("enk in ničel") v ukaznem registru je v našem primeru že osnutek **ukaza** (mikroukaz) za namišljeno operacijo seštevanja. Za kompletnost ukaza **manjkata** še operanda – števili, ki ju seštevamo.



Slika 2: Posamezni biti mikroukaza (zgoraj) aktivirajo ustrezne elektronske podsklope mikroprocesorja. Za vsako operacije je točno določeno 'njeno' zaporedje bitov - mikroukaz.

Programski ukaz si torej lahko predstavljamo kot določeno binarno kombinacijo, ki določa aktivnosti vgrajenih elektronskih vezij v mikroprocesorju. Na primer 1101011100110101. Kaj nam pove ta binarna kombinacija? Nič, dokler ne vemo, za kateri procesor gre in nimamo v rokah njegove podrobne dokumentacije.

Mikroprocesor izvaja le najosnovnejše operacije. Kompleksnejše obdelave podatkov pa iz osnovnih operacij sestavimo iz zaporedja osnovnih operacij in dobimo ... ja, program! Seznam ukazov - program - pripravimo in shranimo v pomnilniku, od koder se ti drug za drugim prenašajo mikroprocesorju v izvajanje. Vsak prebrani **ukaz** se mora v notranjosti mikroprocesorja – v **mikroukaznem registru** – pretvoriti v zaporedje mikroukazov, ki na najnižjem, fizičnem nivoju, predstavljajo določene električne krmilne signale. Vsak **ukaz**, ki ga mikroprocesor lahko izvede, se torej v mikroukaznem registru predela v eno ali več binarnih kombinacij, mikroukazov. Za zahtevnejšo uporabo mikroprocesorja potrebujemo proizvajalčevo dokumentacijo o arhitekturi in delovanju procesorja. In seveda seznam vseh njegovih ukazov. Takšna dokumentacija lahko obsega več sto gosto popisanih in ilustriranih strani.