



Univerza na Primorskem
PEDAGOŠKA FAKULTETA KOPER
Università del Litorale
FACOLTÀ DI STUDI EDUCATIVI DI CAPODISTRIA
University of Primorska
FACULTY OF EDUCATION KOPER

Uporabniška programska oprema

1. zvezek

Andrej Brodnik, Iztok Savnik,
Tamara Bertok, Matjaž Kljun

Študijsko gradivo v elektronski obliki

Univerza na Primorskem, Pedagoška fakulteta Koper

2007

Andrej Brodnik, Iztok Savnik, Tamara Bertok, Matjaž Kljun
Uporabniška programska oprema – 1. zvezek

CIP - Kataložni zapis o publikaciji
Narodna in univerzitetna knjižnica, Ljubljana

004.4(075.8)

UPORABNIŠKA programska oprema [Elektronski vir] :
študijsko gradivo v elektronski obliki / Andrej Brodnik ... [et al.]. –
Koper : Pedagoška fakulteta, 2007

Način dostopa (URL): <http://www.pef.upr.si/gradiva>

ISBN 978-961-6528-63-4 (zv. 1)

1. Brodnik, Andrej

231430144

Kazalo**Seminarske vaje**

Vmesniki	5
Predstavitev podakov	13
i18n in l10n	17
Besedila	23
LaTeX	29
LaTeX nadaljevanje	33
Preglednice	39
Preglednice nadaljevanje	45
Vektorska grafika	49
Rastrska grafika	55
XML	61
Matematični problemi	73
CMS	77

Laboratorijske vaje

INTERNACIONALIZACIJA - Postopek prevajanja programov	83
DELO Z BESEDILI - Razumevanje strukture besedila	87
DELO Z BESEDILI - Urejanje dokumenta	91
DELO S PREGLEDNICAMI - Razumevanje dela s preglednicami	93
DELO S PREGLEDNICAMI - Razumevanje dela s preglednicami	95
SLIKE – Vektorske slike	99
SLIKE – Delo s slikami – rastrske slike	101
MATEMATIČNI PROBLEMI – Octave	103

Uporabniška programska oprema

2006/2007
Iztok Savnik

Predavanje: 1

UPO 2006/07
© Andrej Brodnik

1

Kje, kdaj, ...

- Predavanja: torek, 13h – 13.45
- Vaje: četrtek, 13.30 – 14.15
- E-naslov: Iztok.Savnik@PEF.UPR.SI
Matjaz.Kljun@PEF.UPR.SI
- URL: www.pef.upr.si/MARA/2/UPO
- E-spisek: upo@pef.upr.si
www.pef.upr.si/mailman/listinfo

Predavanje: 1

UPO 2006/07
© Andrej Brodnik

2

Smoter

- spoznavanju študentov z osnovnimi elementi različne programske opreme
- možni načini rokovanja in dela z opremo
- principi obdelovanja besedil, števil (matematičnih izrazov) in slik
- posebna orodja, ki združujejo prej pridobljeno znanje

Predavanje: 1

UPO 2006/07
© Andrej Brodnik

3

Snov

1. uporabniški vmesniki
2. predstavitev podatkov in delo z besedili (preprosta besedila, *LaTeX*, *oowriter*, *HTML*, *i18n*, *l10n*)
3. obdelovanje števil in matematičnih izrazov (*sc*, *oocalc*, *octave*)
4. obdelovanje slik (*xv*, *xfig*, ...)

Predavanje: 1

UPO 2006/07
© Andrej Brodnik

4

Snov (nadaljevanje)

1. označevanje besedil (*HTML*, *XML*)
 2. podatkovne baze (*MySQL*)
 3. spletne predstavitve (*http*, *HTML*)
 4. orodja za programsko podporo spletnih predstavitev (*php*)
 5. lastnina programske opreme in pravni vidiki
 6. uporaba računalnika v izobraževanju – sistemi za upravljanje vsebin
- projekt

Predavanje: 1

UPO 2006/07
© Andrej Brodnik

5

Literatura

- M. Jogan in ostali: Uporabniška programska oprema, 2. dopolnjena izdaja, Ljubljana, Fakulteta za računalništvo in informatiko, 2001
- kasneje pride še nekaj naslovov: *LaTeX*, *Open Office*, *Octave*, unix priročniške strani, itd.
- priročniki na spletu

Predavanje: 1

UPO 2006/07
© Andrej Brodnik

6

Ocenjevanje

- Ocena sestavljena iz:
 1. ocene domačih nalog:
 - se delajo po skupinah, ki se menjajo - jih določa asistent
 - približno vsaka dva tedna
 2. ocene skupnega projekta
 3. ocene kolokvijev (dva do trije kolokviji)
 4. ocene pismenega izpita
 5. ocene ustnega izpita
 6. zapiskov
- Vsaka od delnih ocen mora biti pozitivna

Predavanje: 1

UPO 2006/07
© Andrej Brodnik

7

Uporabniška programska oprema

- kaj je uporabniška programska oprema
- najprej moramo odgovoriti na vprašanje kaj je *programska oprema*
- uporabnik je nekdo, ki uporablja računalnik za svoje dnevno delo

Predavanje: 1

UPO 2006/07
© Andrej Brodnik

8

Uporabniški vmesniki

- na začetku **stikala** (dvojiški / binarni zapis):

```
0 0 0 0 0
0 1 1 0 0
1 0 1 1 0
0 1 0 1 0
0 0 0 0 0
...
```

- zapis s papirja:

- na papirnati trak: **tipkovnica**, **teleprinter**
- kartonske kartice: **tipkovnica**, **bralec kartic**
- terminal: **tipkovnica**, **zaslon**; **miška**, **pero**, ...
- posebni primeri: **merilci hitrosti**, **tehtnice**, ...
- navidezna resničnost: **rokavice**, **krmila**, ...

Predavanje: 1

UPO 2006/07
© Andrej Brodnik

9

Uporabniki

- začetniki
 - človek z ulice; splošni potrošnik
 - nekaj osnovne uporabnosti: *reklama*
- občasni uporabnik
 - človek, ki z računalnikom in opravlja osnovna opravila; redno prebiranje pošte, brskanje, pisanje besedil
 - dostop do celovite funkcionalnosti
- zahtevni uporabnik
 - človek, ki z računalnikom opravlja vse svoje delo; uradnik, računovodja
 - **hiter** dostop do celovite funkcionalnosti

Predavanje: 1

UPO 2006/07
© Andrej Brodnik

10

Plasti uporabniškega vmesnika

Kot primer si bomo ogledali brisanje datoteke

- **konceptualna:**
 - kaj ponuja vmesnik
 - brisanje datoteke
- **semantična:**
 - pomen posamezne uporabnikove dejavnosti
 - *rm a.b*: izbriše datoteko z imenom a.b

Predavanje: 1

UPO 2006/07
© Andrej Brodnik

11

Plasti uporabniškega vmesnika

- **sintaktična:**
 - pravilnost posameznega uporabnikovega opravila
 - *rm <ime datoteke>*: ali datoteka obstaja
- **leksikalna:**
 - pravilnost vnosa posamezne uporabnikove zahteve
 - *rm a.b*: ali je prvi niz „*rm*“ in drugi ime datoteke

Predavanje: 1

UPO 2006/07
© Andrej Brodnik

12

Plasti uporabniškega vmesnika

- Programska oprema narejena v obratnem vrstnem redu – primer ukaz: rm a.b
 - **leksikalna**: ali je prvi niz »rm« in drugi ime datoteke
 - **sintaktična**: ali datoteka »a.b« obstaja
 - **semantična**: izbriše datoteko z imenom »a.b«
 - **konceptualna**: brisali bomo datoteko

Predavanje: 1

UPO 2006/07
© Andrej Brodnik

13

Dober uporabniški vmesnik

- **celovitost**: lahko naredimo vse
 - imenik lahko naredimo, si ga ogledamo, ga izbrišemo
- **konsistentnost**: podobna opravila naredimo na podoben način
 - ukaz za brisanje se vedno prične s črkama *rm*; ista ikona za podobna opravila
- **učinkovitost**: izbiro pogostejših opravil naredimo na preprostejši način
 - daljši nizi za redkejše ukaze; krajši ukazi brez miške

Predavanje: 1

UPO 2006/07
© Andrej Brodnik

14

Dober uporabniški vmesnik

- **razširljivost**: uporabnik lahko dodaja svoje ukaze
 - novi ukazi; sestavljeni ukazi; I18N in L10N (slovenjenje GNU orodij - LUGOS)
- **odzivnost**: ko uporabnik izda ukaz, mora to videti
 - ponavljajoči izpis *delam*; spreminjajoča se ikona
- **rokovanje ob napakah**: sistem zazna napačno vnešen ukaz, ki ga uporabnik lahko popravi
 - *datoteka ne obstaja*; *delete* tipka; *Ali je to pravilen ukaz?*

Predavanje: 1

UPO 2006/07
© Andrej Brodnik

15

Dober uporabniški vmesnik

- **pomoč**: sistem nudi uporabniku pomoč pri delu
 - pomoč na različnih plasteh; pomni uporabnikov profil

Predavanje: 1

UPO 2006/07
© Andrej Brodnik

16

Kako meriti kakovost

- čas učenja
- zapletenost (hitrost) doseganja željenega cilja
- odziv ob uporabnikovih napakah in kako verjetne so
- osebno zadovoljstvo uporabnikov
- cena izdelave vmesnika

Predavanje: 1

UPO 2006/07
© Andrej Brodnik

17

Osnovne oblike vmesnikov

Opazujemo:

- kako izgleda delo z nekim vmesnikom in kakšno opremo zahteva
- prednosti in slabosti vmesnika
- primeri

Predavanje: 1

UPO 2006/07
© Andrej Brodnik

18

Osnovne oblike vmesnikov

- ukazna vrstica: školjka
- obrazci: grafična prijava v sistem
- meniji: preko ukazov ali izbire v grafičnem načinu
- neposredno rokovanje: zahtevne simulacije, igre

Predavanje: 1

UPO 2006/07
© Andrej Brodnik

19

Uporabniška programska oprema

2006/2007
Iztok Savnik

Predavanje: 2

UPO 2006/07
© Andrej Brodnik

1

Predstavitev podatkov

- (digitalni) računalniki delajo samo veličinami, ki imajo vrednost 0 ali 1: **dvojiški** ali **binarni** podatek - **bit**
- če obravnavamo več dvojiških podatkov hkrati, dobimo dvojiška števila: 01110, 10111
- osem bitov predstavlja **zlog** (*byte*)
- z dvojiškimi števili lahko računamo
- drugače: *računalnik* računa z dvojiškimi števili (mi pa lahko nadaljujemo z desetiški)

Predavanje: 2

UPO 2006/07
© Andrej Brodnik

2

Predstavitev številskih podatkov

- vsa števila v računalniku so omejena, kar pomeni, da ne moremo delati z zelo velikimi števili
- zato števila zapišemo v decimalni obliki (zapis s **plavajočo vejico**):
 $\pm 0, \langle \text{mantisa} \rangle E \pm \langle \text{eksponent} \rangle$
- tako mantisa kot eksponent sta še vedno omejena po velikosti

Predavanje: 2

UPO 2006/07
© Andrej Brodnik

3

Predstavitev številskih podatkov

- lahko delamo z večjimi količinami, toda problem je natančnost izračuna
- recimo, da sta mantisa in eksponent omejena na dve števki; potem:

$$\begin{aligned}
 0,1 E+1 + 0,1 E-10 &= 0,1 E+1 \\
 \dots + 0,1 E-10 &= 0,1 E+1 \\
 \dots + \dots &= 0,1 E+1 \\
 \dots + 0,1 E-10 &= 0,1 E+1
 \end{aligned}$$

Predavanje: 2

UPO 2006/07
© Andrej Brodnik

4

Predstavitev črkovnih podatkov

- računalnik zna rokovati samo s števili; kaj pa črke in nizi?
- možna rešitev:
 - črke uredimo v abecedo in jim priredimo številčne kode: 0 – A, 1 – B, ..., 25 – Ž, 26 – a, ...
- ASCII:
 - 65 – A, 66 – B, ..., 97 – a, ...
 - 48 – 0, ...
 - 33 – !, ...
- težave z neangleškimi pisavami

Predavanje: 2

UPO 2006/07
© Andrej Brodnik

5

Predstavitev črkovnih podatkov

- težave z neangleškimi pisavami
- ISO 646:
 - 7 bitne kode: 0 ... 127
 - nekateri znaki namenjeni za nacionalne pisave
 - ISO646-YU: ČŠŽ, ISO646-US: ^[@
- ne moremo sestavljati različnih pisav

Predavanje: 2

UPO 2006/07
© Andrej Brodnik

6

Predstavitev črkovnih podatkov

- zato družina kodiranj ISO 8859:
 - 8 bitne kode: 0 ... 255
 - vsak član različne kode 128 ... 255
 - srednja Evropa ISO 8859-2 (Latin2)
 - ISO 8859 vključuje tudi grško, hebrejsko in arabsko abecedo
- težava sedaj kitajščina in podobni jeziki ter pisanje besedila v večih abecedah hkrati

Predavanje: 2

UPO 2006/07
© Andrej Brodnik

7

Predstavitev črkovnih podatkov

- definicija novega standarda **unicode** – ISO10646 (www.unicode.org)
 - 16 ali 32 bitni zapis
 - možnost hkratne uporabe vseh znakov
 - URL: www.unicode.org/versions/bookmarks.html

Predavanje: 2

UPO 2006/07
© Andrej Brodnik

8

UTF-8

- Kode znakov so lahko različno dolge
- URL: www.fileformat.info/info/unicode/utf8.htm

1. zlog	2. zlog	3. zlog	st. bitov	st. znakov
0xxxxxxx			7	127
110xxxxx	10xxxxxx		(5+6)=11	2047
1110xxxx	10xxxxxx	10xxxxxx	(4+6+6)=16	65535

Predavanje: 2

UPO 2006/07
© Andrej Brodnik

9

Predstavitev ostalih podatkov

- vsi ostali podatki (slika, zvok, ...) so *digitalizirani* *analogni* podatki ali kombinacija le-teh z digitalnimi podatki
- na primer slika:
 - sestoji iz točkic (*pixel*), ki so lahko bolj ali manj svetle (stopnja sivine)
 - sivina je določena s številom
 - pri barvnih slikah je točkica sestavljena iz hkratne pojave rdeče, zelene in modre barve (*RGB*)

Predavanje: 2

UPO 2006/07
© Andrej Brodnik

10

Uporabniška programska oprema

2006/2007
Iztok Savnik

Predavanje: 3

UPO 2006/07
© Andrej Brodnik

1

i18n in l10n

- programsko opremo želimo pisati tako, da bo uporabna v več naravnih jezikih
 - *internationalization* – **i18n**
- tako napisano programje potem postavimo v določeno jezikovno okolje
 - *localization* – **l10n**

Predavanje: 3

UPO 2006/07
© Andrej Brodnik

2

i18n

- Sporočila v programju ne zapišemo neposredno, ampak njihove **identifikatorje**
- Izgradimo **katalog (imenik)** vseh identifikatorjev in njim prirejenih sporočil
- Programje uporabi na mestu identifikatorja sporočilo iz kataloga

Predavanje: 3

UPO 2006/07
© Andrej Brodnik

3

110n

- Katalog lahko sedaj prevedemo v poljuben jezik
- Uporabnik programu pove, v katerem jeziku želi, da se programje z njim pogovarja

Predavanje: 3

UPO 2006/07
© Andrej Brodnik

4

Pametnejše prevajanje

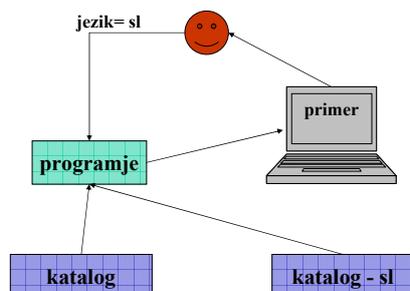
- Besedila ne moremo preprosto prevajati po besedah
 - spol, sklon, število
 - vrstni red besed v sporočilu
- Lokalizirati moramo še vsaj:
 - kako se pariyo velike in male črke ter kakšen je abecedni red črk
 - kako izpisujemo čas in datum
 - kako izpisujemo denarne vrednost
 - pisanje števil

Predavanje: 3

UPO 2006/07
© Andrej Brodnik

5

Celotna slika



Predavanje: 3

UPO 2006/07
© Andrej Brodnik

6

GNU gettext

- gettext je primer orodja in sistema s podporo za i18n in l10n
- Priročniki:
 - www.gnu.org/software/gettext/gettext.html
 - Primož Peterlin *Krajevno prilagajanje programov s paketom GNU Gettext*
www.lugos.si/arhiv/prispevki/gettext/
 - info gettext
 - man gettext

Predavanje: 3

UPO 2006/07
© Andrej Brodnik

7

Slovenjenje ali l10n v sl

- slovenjenje različne programske opreme:
 - URL: www.lugos.si/delo/slo/
 - Primož Peterlin: *Linux in slovenščina*,
primoz.peterlin@biofiz.mf.uni-lj.si

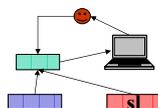
Predavanje: 3

UPO 2006/07
© Andrej Brodnik

8

Primer uporabe gettext – i18n

- V imeniku primer
 - *primer-zacetni.c*: nepripravljen program
 - *primer.c*: program pripravljen za i18n
 - črkovni nizi sami predstavljajo identifikatorje
 - zato je predoločen niz kar sam identifikator



Predavanje: 3

UPO 2006/07
© Andrej Brodnik

9

Primer uporabe gettext – 110n

- priprava kataloga
`primer[1] %> xgettext --extract-all primer.c`
- dobimo proto-katalog *messages.po*
- pomnožimo ga v *messages-sl.po*, ki ga slovenimo
- pri slovenjenju uporabljamo *emacs* v *po-mode* načinu dela (glej C-Hm) ali *kbabel*
- pripravimo jedrnatejši zapis
`primer[2] %> msgfmt --output-file=primer.mo messages-sl.po`
- kam s katalogom?

Predavanje: 3

UPO 2006/07
© Andrej Brodnik

10

110n - nadaljevanje

- Katalog se nahaja v datoteki:
IMENIK/LOCALE/LC_CATEGORY/DOMENA.mo
 - IMENIK/LOCALE = locale
 - sicer */usr/share/locale* ali */usr/local/share/locale*
 - LC_CATEGORY = LC_MESSAGES
 - DOMENA = primer
- Katalog prestavimo na pravo mesto in poskusimo
`primer[3] %> ./primer`
example of gettext usage
`primer[4] %> setenv LC_ALL sl_SI`
`primer[5] %> ./primer`
primer uporabe gettext sistema

Predavanje: 3

UPO 2006/07
© Andrej Brodnik

11

110n - nadaljevanje

- Ali pa v nemščini:
`primer[5] %> setenv LC_ALL de_DE`
`primer[6] %> ./primer`
example of gettext usage
`primer[7] %> msgfmt --output-file=primer.mo messages-de.po`
`primer[8] %> mv primer.mo locale/de/LC_MESSAGES/.`
`primer[9] %> ./primer`
Beispiel der gettext Anwendung

Predavanje: 3

UPO 2006/07
© Andrej Brodnik

12

GNU gettext - orodja

- **xgettext** - izvlečemo vse i18n nize
- **msgfmt** - katalog v strojno optimizirano obliko (Machine Object, MO)
- **msgunfmt** - obratno od programa **msgfmt**
- **msgcmp** - ali dva kataloga sporočil vsebujeta isti nabor identifikatorjev
- **msgmerge** - združimo obstoječ katalog sporočil z novim, ki smo ga pripravili z **xgettext**

Predavanje: 3

UPO 2006/07
© Andrej Brodnik

13

Uporabniška programska oprema

2006/2007
Iztok Savnik

Predavanje: 4

UPO 2006/07
© Andrej Brodnik

1

Izgled besedila

- sestoji iz črk, ki so zapisane v neki pisavi
- uporabljen črke v pisavi imajo več lastnosti:
 - sorazmerna ali enakomerna
 - s serifi ali brez njih
 - običajne, **mastne ali krepke**
 - pokončne ali *poševne*
 - **večje** in manjše

Predavanje: 4

UPO 2006/07
© Andrej Brodnik

2

Izgled besedila

- besedilo ima lahko svoje lastnosti:
 - levo poravnano
 - desno poravnano
 - središčeno
 - pisano z leve
 - ensed z ila

Predavanje: 4

UPO 2006/07
© Andrej Brodnik

3

Ustroj besedila

- ustroj ali struktura besedila vsebuje:
 - naslov, avtorja, datum
 - povzetek
 - poglavja oziroma razdelke
 - uvod, pregled znanih dejstev, predstavitev novih dognanj, zaključek
 - kazalo, indeks slovar
 - spisek uporabljene literature

Predavanje: 4

UPO 2006/07
© Andrej Brodnik

4

Ustroj besedila

- ustroj ali struktura besedila vsebuje:
 - preglednice ali tabele
 - slike
 - pomembnejše (poudarjeno) besedilo
 - sklicevanja
 - ...

Predavanje: 4

UPO 2006/07
© Andrej Brodnik

5

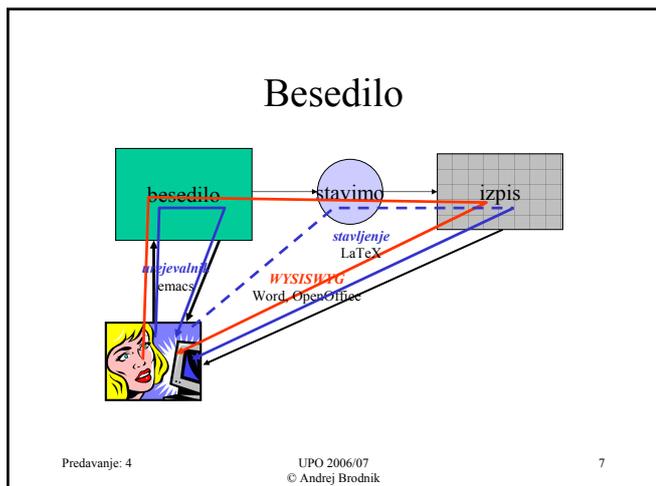
Besedilo

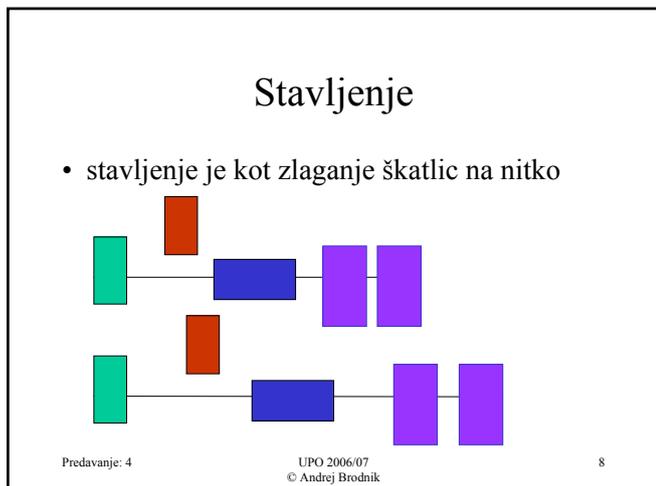
- pri besedilu govorimo o **ustroju** (vsebini), katero predstavimo s pomočjo **izgleda**
- izgled je lahko različen na različnih napravah
- pravimo, da imamo najprej besedilo, ki ga nato **stavimo** ter si ga ogledamo

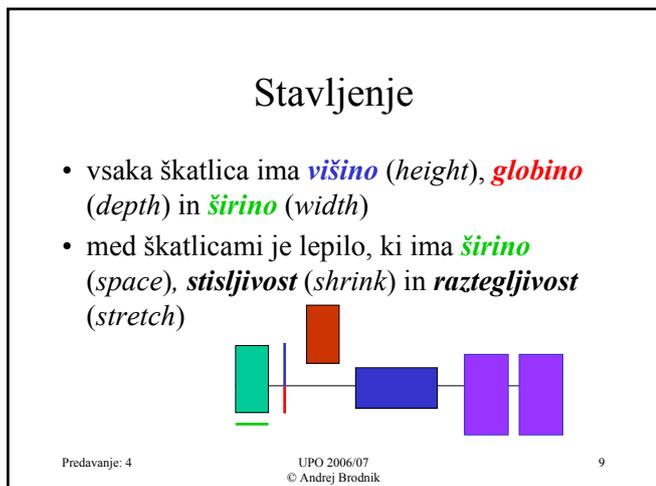
Predavanje: 4

UPO 2006/07
© Andrej Brodnik

6

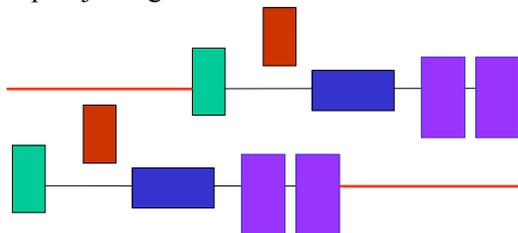






Stavljenje - poravnava

- lepilo je magična sestavina



Predavanje: 4

UPO 2006/07
© Andrej Brodnik

10

Stavljenje - običajna vrstica

Stavi vrstico širine w :

1. seštejemo vse širine škatlic in lepil v s
2. naj bo $r = |w - s|$
3. seštejemo vse stisljivosti lepil d_i v d in raztegljivosti lepil x_i v x
4. če $w > s$ **potem** vsako lepilo razegnemo za $(x_i / x) r$
5. **sicer** vsako lepilo stisnemo za $(d_i / d) r$

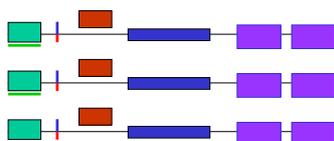
Predavanje: 4

UPO 2006/07
© Andrej Brodnik

11

Stavljenje večih vrstic

- tudi med vrsticami je lepilo
- postopek stavljenja je enak kot pri eni vrstici



Predavanje: 4

UPO 2006/07
© Andrej Brodnik

12

Stavljenje večih vrstic

- za izgled celotne strani ni dovolj, da je vsaka vrstica po sebi zelo lepa, lepa mora biti celotna stran
- vsaka vrstica ima mero kakovosti stavljenja (*demerit*)

Predavanje: 4

UPO 2006/07
© Andrej Brodnik

13

Stavljenje strani

- program za stavljenje stavi stran tako:
 - da je *vsota kakovosti vseh vrstic* najboljša
 - da je *cena prelomov vrstic* čim manjša
 - da je *cena zaključkov vrstic* čim manjša
- prelomi vrstic so lahko na različnih mestih, a se njihov vrstni red ne spreminja
- izračunati moramo kakovost strani za vse možne prelome
- zatorej je metod, ki jo uporabimo,?...

Predavanje: 4

UPO 2006/07
© Andrej Brodnik

14

Kaj pa ostali okraski

- slike so v resnici škatlice, le malce večje
- isto velja za razpredelnice
- škatlice lahko *združujemo* v večje škatle
- vsako škatlico lahko označimo ter se nanjo sklicujemo
- če izpišemo spisek vseh škatlic določenega tipa, dobimo kazalo, indeks, slovar, spisek slik ipd.

Predavanje: 4

UPO 2006/07
© Andrej Brodnik

15

Uporabniška programska oprema

2006/2007
Iztok Savnik

Predavanje: 5

UPO 2006/07
© Andrej Brodnik

1

Struktura dokumenta v LaTeX

- pisanje dokumenta v LaTeX je na nek način podobno programiranju
- določenemu besedilu priredimo lastnosti:
 - besedilo v slovenščini, v Latin-2 zapisu
 - poudarjeno besedilo, naštevanje, navedek, e-naslov, ...
 - označevanje besedila za sklicevanje
 - definiramo lahko nove lastnosti za besedilo

Predavanje: 5

UPO 2006/07
© Andrej Brodnik

2

Primer: opis.tex

```
\documentclass[12pt,a4paper]{article}

\usepackage{times}
\usepackage{slovene}{babel}
\usepackage[latin2]{inputenc}
\usepackage{html}

\newcommand{\HTTP}[1]{
  \htmladdnormallink{\texttt{#1}}{http://#1} }
\newcommand{\EMAIL}[1]{
  \htmladdnormallink{\texttt{#1}}{mailto:#1} }
```

Predavanje: 5

UPO 2006/07
© Andrej Brodnik

3

Opis dokumenta in okolja

- definicija dokumenta

```
\documentclass[<stikalo>,<stikalo>]{<oblika dokumenta>}
\documentclass[12pt,a4paper]{article}
```

- dodatni paketi

```
\usepackage[<stikalo>,<stikalo>]{<ime paketa>}
\usepackage{times}
\usepackage[slovene]{babel}
\usepackage[latin2]{inputenc}
\usepackage{html}
```

Predavanje: 5

UPO 2006/07
© Andrej Brodnik

4

Definicija novih ukazov

- kjerkoli v dokumentu lahko definiramo nov ukaz

```
\newcommand{<ime ukaza>}[<stevilo arg>]{<ukaz>}
```

```
\newcommand{\HTTP}[1]{%
  \htmladdnormallink{\texttt{#1}}{http://#1}%
}% \HTTP
\newcommand{\EMAIL}[1]{%
  \htmladdnormallink{\texttt{#1}}{mailto:#1}%
}% \EMAIL
```

Predavanje: 5

UPO 2006/07
© Andrej Brodnik

5

Kaj je to

- *Alma mater?*
 - *alma* – blagorodna, mila
 - *mater* – mati
- mila mati (, ki nam da znanje kot popotnico v življenje)
- univerza, na kateri zaključimo šolanje

```
\newcommand[1]{\tujka}{\textit{#1}}
\tujka{alma mater}
```

Predavanje: 5

UPO 2006/07
© Andrej Brodnik

6

Primer: opis.tex

```

% -----
\title{Uporabniška programska oprema}
\author{2002 -- 2003}
\date{ \relax } % \date{ \today }
% -----
\begin{document}

\maketitle

...

\end{document}

```

Predavanje: 5

UPO 2006/07
© Andrej Brodnik

7

Primer: opis.tex

```

\section*{Smoter}

Predmet \emph{Uporabniška programska oprema} je namenjen ...

\section*{Snov}

\begin{itemize}
\item uporabniški vmesniki
\item obdelovanje besedil (preprosta besedila, \LaTeX, Word)
\item ... (\texttt{sc}, \texttt{excel}) ...
\end{itemize}

\section*{URL}

```

Predavanje: 5

UPO 2006/07
© Andrej Brodnik

8

Razdeljevanje besedila

- poglavje (*chapter*), razdelek (*section*), podrazdelek (*subsection*), podpodrazdelek (*subsubsection*), ...
- če imenu sledi zvezdica, se razdelek ne številči

```

\section*{Smoter}
\section*{Snov}

```

Predavanje: 5

UPO 2006/07
© Andrej Brodnik

9

Spiski

- navadni spisek

```
\begin{itemize}
\item uporabniški vmesniki
\end{itemize}
```

- naštevalni spisek

```
\begin{enumerate}
\item uporabniški vmesniki
\end{enumerate}
```

- opisni spisek

```
\begin{description}
\item[uporabniški] vmesniki
\end{description}
```

Predavanje: 5

UPO 2006/07
© Andrej Brodnik

10

Uporabniška programska oprema

2006/2007
Iztok Savnik

Predavanje: 6

UPO 2006/07
© Andrej Brodnik

1

Primer: opis.tex

```
\section*{Urniki}

\begin{tabular}{ll}
predavanja & pon 13h-14h \\
\\ vaje & čet 9h-10h \\
\end{tabular}

\section*{Predavatelj}

\begin{tabular}{ll}
Andrej Brodnik \\
\\ E-naslov & \EMAIL{Andrej.Brodnik@IMFM.Uni-Lj.SI} \\
\\ & \EMAIL{Andrej.Brodnik@PEF.UPR.SI} \\
\\ Govorilne ure & čet 13h-14h \\
\end{tabular}
```

Predavanje: 6

UPO 2006/07
© Andrej Brodnik

2

Tabela

- tabela sestoji iz vrstic, vsaka od katerih sestoji iz stolpcev
- obliko in razvrstitev stolpcev popišemo s kalupom

```
\begin{tabular}{ll}
predavanja & pon 13h-14h \\
\\ vaje & čet 9h-10h \\
\end{tabular}
```

- v kalupu lahko uporabimo še različne oznake za razmejevanje stolpcev (npr. |)
- med vrsticami lahko naredimo črto z \hline

Predavanje: 6

UPO 2006/07
© Andrej Brodnik

3

Matematika

- stavljenje matematičnih izrazov zahteva nekatere posebnosti:
 - matematični operatorji
 - indeksi in potence
 - različni matematični znaki

Predavanje: 6

UPO 2006/07
© Andrej Brodnik

4

Matematika

- v (La)TeX-u stavimo matematične izraze med dvema znakoma \$:
$$\sum_{i=1}^n i = \frac{n(n+1)}{2}$$
- lahko stavimo kot posebno škatlo preko cele strani:

```
\[  
\sum_{i=1}^n i = \frac{n(n+1)}{2}  
\]
```

Predavanje: 6

UPO 2006/07
© Andrej Brodnik

5

Matematika

- v Office okolju uporabimo poseben urejevalnik matematičnih enačb – matematično enačbo vstavimo kot ločeno škatlo
- pri stavljenju za splet ni dobre podpore in si pomagamo z vstavljanjem slik

Predavanje: 6

UPO 2006/07
© Andrej Brodnik

6

Definicija novih okolij

- želimo imeti preprosto okolje, v katerem bi lahko pisali programsko kodo:

```
\begin{code}
public class mojRazred \{
    public static void main(String args[]) \{ ... \}
\} // mojRazred
\end{code}
```

- podobno dosežemo v Office okolju z uporabo stilov

Predavanje: 6

UPO 2006/07
© Andrej Brodnik

7

Definicija novih okolij

- definiramo kaj naj se zgodi ob nastopu okolja (**begin**) in zaključku (**end**)

```
\newenvironment{code}{%
\par \medbreak \begin{group}%
\parindent 0pt \parskip 0pt%
\renewcommand{\baselinestretch}{1}%
\setbox\strutbox=%
\hbox{\vrule height9pt depth4pt width 0pt}%
\normalbaselines\everypar{\strut}
\obeylines \obeyspaces%
\texttt%
}{%
\endgroup\ignorespaces \medbreak \noindent%
} % code
```

Predavanje: 6

UPO 2006/07
© Andrej Brodnik

8

Plavajoče škatle

- za plavajoče škatle velja, da poznamo njihovo velikost, vendar prepuščamo sistemu, da jih postavi na tisto mesto, za katero misli, da je najprimernejše
- primeri plavajočih škatel so slike, tabele ipd.

Predavanje: 6

UPO 2006/07
© Andrej Brodnik

9

Plavajoče škatle

- Kam naj sistem postavi plavajočo škatlo naročimo z eno od črk:
 - h: tukaj (čim bližje) (*here*)
 - b: na dnu te strani (*bottom*)
 - t: na vrhu naslednje strani (*top*)
 - p: na ločeni strani (*page of floats*)
- v LaTeX okolju sta definirani dve obliki plavajočih škatel: slike (*figure*) in razpredelnice (*table*)

Predavanje: 6

UPO 2006/07
© Andrej Brodnik

10

Sklicevanje

- V besedilu se želimo sklicevati na neko mesto nekje drugje v besedilu. Na primer:
 - na enačbo,
 - na razpredelnico,
 - na sliko,
 - na definicijo, ...
- sklicujemo se lahko na samo stvar (npr. na sliko to in to) ali na stran, na kateri se nahaja ta stvar
- da se lahko sklicujemo moramo najprej stvar označiti

Predavanje: 6

UPO 2006/07
© Andrej Brodnik

11

Sklicevanje

- LaTeX:
 - označujemo stvari z ukazom `\label{oznaka}`
 - sklicujemo se z `\ref{oznaka}` oziroma `\pageref{oznaka}`
- Office okolje:
 - je označevanje samodejno
 - sklicujemo se tako, da vstavimo navzkrižni sklic (*cross-reference*)
- HTML:
 - označimo z `...`
 - se sklicujemo z `...`

Predavanje: 6

UPO 2006/07
© Andrej Brodnik

12

Sklicevanje v LaTeX-u

- Dodamo oznake enačbi:

```
definicija $n$-tega Fibonaccijevega števila
$F(n)$ je:
%
\begin{equation} \label{enacba:fib}
F(n) = F(n-1) + F(n-2)
\end{equation}
```

Predavanje: 6

UPO 2006/07
© Andrej Brodnik

13

Sklicevanje v LaTeX-u

- Kodo predstavimo kot sliko (plavajoča škatla):

```
\begin{figure}[hbt]
\begin{code}
~public class fibonacci \{
~ public int fib(int n) \{
~ if (n <= 1) return 1;
~ return fib(n-1) + fib(n-2)
~ \} // fib
~\} // fibonacci
\end{code}
\caption{Izračun $n$-tega Fibonaccijevega števila.}
\label{slika:fib}
\end{figure}
```

Predavanje: 6

UPO 2006/07
© Andrej Brodnik

14

Sklicevanje v LaTeX-u

- na škatlo se sedaj sklicujemo takole:

```
Program na sliki \ref{slika:fib}, ki je na strani
\pageref{slika:fib} rekurzivno izračuna
Fibonaccijevo zaporedje ...
```

Predavanje: 6

UPO 2006/07
© Andrej Brodnik

15

Sklicevanje v HTML

- V HTML je sklicevanje splošnejše, saj brskalnik omogoča takojšnji ogled stvari, na katero se sklicujemo
- Sklicujemo se lahko tudi izven besedila in zato so vsi dokumenti med seboj prepleteni / spleteni – zato *splet*
- za primer

<http://www.pef.upr.si/TeX>

Uporabniška programska oprema

2006/2007
Iztok Savnik

Predavanje: 7

UPO 2006/07
© Andrej Brodnik

1

Glavna knjiga

- Računovodstvu se ukvarja z vodenjem stanja računov - s prihodki in odhodki
- Glavna knjiga je glavni dokument, v katerem so zavedeni vsi prihodki in odhodki v obliki razpredelnice
- Poleg tega mora računovodja v vsakem trenutku vedeti stanje na računu

Predavanje: 7

UPO 2006/07
© Andrej Brodnik

2

Primer

<i>postavka</i>	<i>v dobro</i>	<i>v breme</i>
prenos	10.123,90	
bencin		7.245,60
cestnina		840,00
SKUPAJ:	10.123,90	8.085,60
STANJE:	2.038,30	

Predavanje: 7

UPO 2006/07
© Andrej Brodnik

3

Primer

<i>postavka</i>	<i>v dobro</i>	<i>v breme</i>
prenos	10.123,90	
bencin		7.245,60
cestnina		840,00
bencin		6.250,30
SKUPAJ:	10.123,90	14.335,90
STANJE:		4.212,00

Predavanje: 7

UPO 2006/07
© Andrej Brodnik

4

Težave

- Ko dodamo nov vnos v glavno knjigo, moramo preračunati zadnji vrstici
- Možna rešitev: dodamo stolpec/ce in vanj vnašamo delne vsote
- Dodamo programje, ki samodejno popravlja dodatne stolpce
- Sedaj želimo dodatno ločene seštevke za bencin in za cestnino
- Več dodatnih stolpcev

Predavanje: 7

UPO 2006/07
© Andrej Brodnik

5

Rešitev

Združimo dve ideji:

1. problem posplošimo, da dobimo dvo-razsežnostno tabelo, v katero lahko vnašamo vrednosti
2. vrednost posameznega polja je lahko konstanta ali rezultat poljubnega izraza, ki vsebuje kot parametre (trenutne) vrednosti drugih polj

Predavanje: 7

UPO 2006/07
© Andrej Brodnik

6

Rešitev

- Ostajata dve odprti vprašanji:
 1. kakšne so lahko vnešene vrednosti
 2. kakšni so lahko izrazi (funkcije)
- Ker želimo imeti splošno rešitev, so lahko vnešene vrednosti poljubnega tipa, ki ga v splošnem pozna sistem
- podobno velja za izraze, ki jih razširimo s funkcijami

Predavanje: 7

UPO 2006/07
© Andrej Brodnik

7

Preglednica

- Preglednica je dvorazsežna tabela, ki sestoji iz polj
- Vsako polje ima svoje koordinate: tipično črko za stolpec in številko za vrstico
- V polja vnašamo vrednosti in funkcije
- Ker so parametri funkcije vrednosti drugih polj, ki so ponovno lahko funkcije, moramo po vsakem vnosu ponovno preračunati vrednosti v tabeli

Predavanje: 7

UPO 2006/07
© Andrej Brodnik

8

Programi

- med prvimi je bil *VisiCalc*
- med najbolj znanimi *Lotus-1-2-3*
- MicroSoft *Excel*
- mi bomo najprej uporabljali program *sc* (glej priročnik pod *man sc*)
 - prednost preprostost a dovolj sposoben, da si pogledamo koncepte
 - na voljo izvorna koda
- kasneje bomo uporabljali razpredelnico iz *OpenOffice*

Predavanje: 7

UPO 2006/07
© Andrej Brodnik

9

Prvi primer

- preglednica je shranjena v datoteki *primer1.sc* in jo pogledamo z ukazom
- dobimo

n	B	C	D	E	F	G
postavka v dolarih v dnevi						
prenos	10129,50					
benzina		7245,60				
cestarina		840,00				
bencin		6250,30				
skupaj	10129,50	14335,50				
stanej						-1117,000

Predavanje: 7

UPO 2006/07
© Andrej Brodnik

10

Prvi primer

- vsebina datoteke *primer1.sc* je v ASCII obliki
- izziv: ročno popravite datoteko

Predavanje: 7

UPO 2006/07
© Andrej Brodnik

11

Preračunavanje

- Kako sploh vse skupaj deluje?
- Zasnova je dogodkovno gnano programiranje
- Dogodek je sprememba vrednosti, ki jo vnesemo
- Naprej vsako polje v \hat{e} , vrednost katerih polj je odvisna od njegove spremembe in jih ustrezno popravi

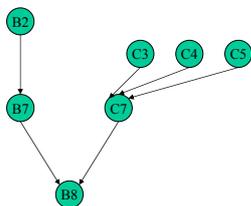
Predavanje: 7

UPO 2006/07
© Andrej Brodnik

12

Primer

```
let B2 = 10123.9
let C3 = 7245.6
let C4 = 840
let C5 = 6250.3
let B7 = @sum(B1:B6)
let C7 = @sum(C1:C6)
let B8 = B7-C7
```



Predavanje: 7

UPO 2006/07
© Andrej Brodnik

13

Preračunavanje

- imamo usmerjen graf polj
- ko pride do spremembe v polju X_i , pokličemo funkcijo $popravi(X_i)$:
- function $popravi(X_i)$:
 - popravi vrednost v X_i
 - za vsa vozlišča Y_j , do katerih obstaja usmerjena povezava (X_i, Y_j) kliči funkcijo $popravi(Y_j)$
- vse v redu?
- ne, manjka **zaustavitveni pogoj**

Predavanje: 7

UPO 2006/07
© Andrej Brodnik

14

Preračunavanje

- problem preračunavanja je poseben primer sprehoda po grafu
- sprehod **v širino** ali **v globino**
- v obeh primerih moramo zaznati, ali smo obiskali vozlišče, ki smo ga obiskali že prej
- obraten problem je **problem iskanja cikla** v grafu
- podoben problemu sprehoda po labirintu

Predavanje: 7

UPO 2006/07
© Andrej Brodnik

15

Preračunavanje

- kaj naredimo?
- na vozliščih puščamo kamenčke
- function popravi(X_i):
 - če ima polje že kamenček
 - napaka
 - pusti kamenček
 - popravi vrednost v X_i
 - za vsa vozlišča Y_j , do katerih obstaja usmerjena povezava (X_i, Y_j) kliči funkcijo *popravi*(Y_j)
 - poberi kamenček

Predavanje: 7

UPO 2006/07
© Andrej Brodnik

16

Preračunavanje

- ali je rešitev zadovoljiva?
- vrednost polja je lahko poljubna funkcija
- funkcije so poseben primer Turingovega stroja
- za Turingove stroje ne moremo vedeti ali se na nekih vhodnih podatkih ustavijo ali ne
- zato je običajno kamenček števec

Predavanje: 7

UPO 2006/07
© Andrej Brodnik

17

Uporabniška programska oprema

2006/2007
Iztok Savnik

Preda vanje: 8

UPO 2006/07
© Andrej Brodnik

1

Oblikovanje izpisa

- Nadzor nad obliko izpisa obstaja na splošno pri vseh programskih jezikih (glej *man 3 printf*)
- Pri oblikovanju polja lahko zahtevamo (ukaz *F*):
 - oblika številke: `###` ali `000`
 - mesto decimalne pike (vejice) in razmejitev tisočic: `#,###.00`
 - odstotkovna vrednost: `#,###,###.00%`
 - poljuben niz: `#,###,###.00 SIT`
 - znastveni izpis: `0.### E+00`

Preda vanje: 8

UPO 2006/07
© Andrej Brodnik

2

Oblikovanje izpisa

- Datoteka *primerF.sc*



Preda vanje: 8

3

Oblikovanje izpisa

- Poleg posameznega polja lahko oblikujemo tudi izgled celotnega stolpca
- Oblikujemo lahko (ukaz *f*):
 - dolžino izpisa
 - dodatno formatiranje
 - tip (cela števila, znanstveni zapis, inženirski zapis, datumi)

Preda vanje: 8

UPO 2006/07
© Andrej Brodnik

4

Oblikovanje izpisa

- Datoteke



Preda vanje: 8

UPO 2006/07
© Andrej Brodnik

5

Sklicevanje na polja

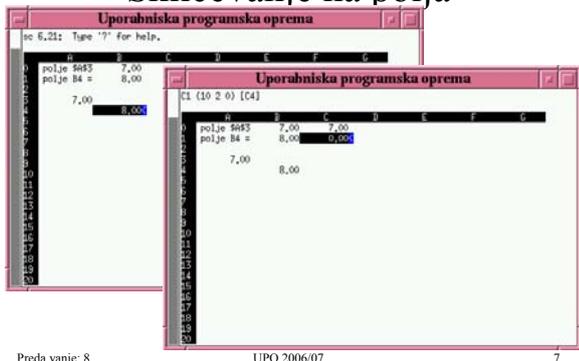
- Lahko tudi prepisujemo vsebino posameznih polj na druga polja
- Pri tem se moramo nadzorovati ohranjanje sklicevanja:
 - točno določeno polje (stolpec, oziroma vrstico): $\$A\3
 - sklicevanje se ustrezno popravi: B4
- Osnova bo datoteka *primer1* in bomo prepisali polji B0 in B1 ter dobili *primer2*

Preda vanje: 8

UPO 2006/07
© Andrej Brodnik

6

Sklicevanje na polja



Preda vanje: 8

UPO 2006/07
© Andrej Brodnik

7

Sklicevanje na polja

- Sklicevanje na polja je pomembno, ko prepisujemo celo območje polj
- Če prepisovano območje tvori zaključeno celoto, želimo, da se odnosi med polji območja ohranijo

Preda vanje: 8

UPO 2006/07
© Andrej Brodnik

8

Sklicevanje na polja

- Lahko definiramo spremenljivke:
`#define "vsota" A10`
- in se potem sklicujemo na spremenljivko *vsota*
- Problem, če se sedaj polje premakne, saj *vsota*, pomni staro mesto polja

Preda vanje: 8

UPO 2006/07
© Andrej Brodnik

9

Funkcije

- splošne funkcije: vsota, povprečje, preštej, maksimum, minimum, ...
- matematične funkcije: logaritem, eksponentna funkcija, kotne funkcije, ...
- funkcije nad nizi: stik, podniz, ...
- statistične funkcije: (vsota), standardni odklon, ...
- finančne funkcije: obresti, pologi, obroki, ...
- datumske funkcije

Preda vanje: 8

UPO 2006/07
© Andrej Brodnik

10

Uporabniška programska oprema

2006/2007
Iztok Savnik

Predavanje: 9

UPO 2006/07
© Andrej Brodnik

1

Slikovno gradivo

- Doslej smo si ogledali kako oblikovati:
 - besedila (tekstovno gradivo): OpenOffice, MS Word, LaTeX, HTML
 - številčne podatke (preglednice): sc, OpenOffice, MS Excel
- Pogledali si še bomo oblikovanje slikovnega gradiva
 - strukturirano (predmeti): xfig, OpenOffice, gnuplot
 - slike: zapisi bmp, jpeg, gif ipd.

Predavanje: 9

UPO 2006/07
© Andrej Brodnik

2

Strukture

- Strukturirano slikovno gradivo je razdeljeno na gradnike
- Obravnavajmo gradnike kot predmete v predmetno naravnem programiranju
- Vsak gradnik ima lastnosti:
 - da se zna narisati
 - da vê, kje je njegova *baza*
 - da vê, kateri plasti pripada

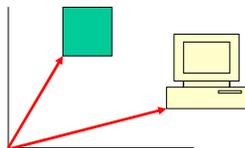
Predavanje: 9

UPO 2006/07
© Andrej Brodnik

3

Baza gradnika

- Vsak gradnik ima svojo bazo
- Baza je vektor odmika od izhodišča



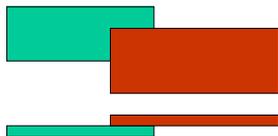
Predavanje: 9

UPO 2006/07
© Andrej Brodnik

4

Plasti

- Posamezne gradnike nalagamo v plasti ali sloje
- Gradniki v višje ležečih slojih so vidni in zakrivajo tiste v spodnjih slojih



Predavanje: 9

UPO 2006/07
© Andrej Brodnik

5

Združevanje

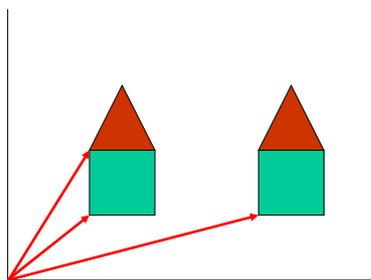
- Gradnike lahko združujemo v nove gradnike
- Podobno kot če tvorimo nove razrede, ki vsebujejo prejšnje razrede
- Vsi gradniki pa so nasledniki pra-gradnika
- Zato imajo vsi lastnosti: možnost izrisovanja, pripadnost plasti in bazo

Predavanje: 9

UPO 2006/07
© Andrej Brodnik

6

Združevanje



Predavanje: 9

UPO 2006/07
© Andrej Brodnik

7

Osnovni gradniki

- črta
 - sklenjena in neskenjena
 - prostorska in parametrizirana (zlepki)
 - lomljena
 - krožni lok
- večkotnik
 - kvadrat / pravokotnik
- krog / elipsa
- besedilo
- slika
- knjižnični element



Predavanje: 9

UPO 2006/07
© Andrej Brodnik

8

Lastnosti

- Gradniki imajo svoje lastnosti
- Nekateri gradniki imajo enake vrste lastnosti (debelina crte, barva ploskve ipd.)
- Ker so nekateri gradniki izpeljani iz drugih gradnikov podejujejo lastnosti
- Orodje za strukturirano risanje omogoča speminjanje lastnosti

Predavanje: 9

UPO 2006/07
© Andrej Brodnik

9

Lastnosti črt

- Črte so lahko gradniki same po sebi ali predstavljajo robove likov, nosilce vektorjev (puščic) ipd.
- Običajne lastnosti so:
 - debelina, oblika (polna, črtkana, pikčasta, ...)
 - barva črte
 - način stikanja večih črt
 - oblika puščic → → ↔



Predavanje: 9

UPO 2006/07
© Andrej Brodnik

10

Lastnosti ploskev

- Barva ploskve
- Vzorec barvanja



Predavanje: 9

UPO 2006/07
© Andrej Brodnik

11

Lastnosti besedil

- Velikost črk
- Pisava
- Poravnanos (levo, središčeno, desno)
- Naklon pisanja
- Posebne lastnosti



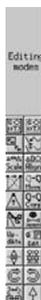
Predavanje: 9

UPO 2006/07
© Andrej Brodnik

12

Urejanje slik

- Orodja za urejanje slik lahko združimo v skupine:
 - celotni gradniki: zrcaljenje (preko točke ali preko premice) in premikanje; poravnavanje; povečevanje; kopiranje
 - sestavljeni gradniki: združevanje in razdruževanje; odpiranje združenih gradnikov in urejanje v njih



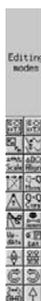
Predavanje: 9

UPO 2006/07
© Andrej Brodnik

16

Urejanje slik

- delo s črtami: lomljenje in lepljenje črt; spreminjanje v zlepke in obratno; zaključevanje črt
- splošno urejanje in spreminjanje lastnosti



Predavanje: 9

UPO 2006/07
© Andrej Brodnik

17

Uporabniška programska oprema

2006/2007
Iztok Savnik

Predavanje: 10

UPO 2006/07
© Andrej Brodnik

1

Slikovno gradivo

- Zadnjič smo si ogledali kako pripravljamo strukturirano slikovno gradivo
- Pri strukturiranem slikovnem gradivu imamo predmete z lastnostmi, ki jih urejamo – osnovni gradnik je predmet
- Primer programa:
 - *xfig*
 - *AutoShapes* v *Office* paketu
 - do neke mere *gimp*

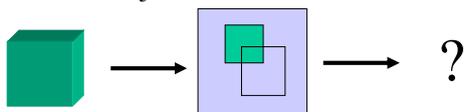
Predavanje: 10

UPO 2006/07
© Andrej Brodnik

2

Slikovno gradivo

- Program *gimp* pozna tudi operacijo *izreži*
- Kako deluje *izreži*?



Predavanje: 10

UPO 2006/07
© Andrej Brodnik

3

Slikovno gradivo

- Srednja slika je *pogled* na strukturo (predmet) in operacija nad pogledmo moramo preslikati na operacijo nad predmetom
- Enak problem nastopa pri predmetno usmerjenem programiranju in uporabi večobličnosti
- Podoben problem tudi pri podatkovnih bazah

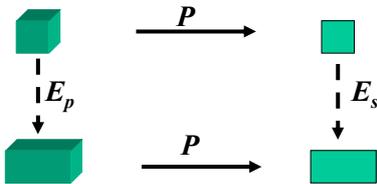
Predavanje: 10

UPO 2006/07
© Andrej Brodnik

4

Preslikave in pogledi

- V osnovi imamo naslednji problem



Predavanje: 10

UPO 2006/07
© Andrej Brodnik

5

Urejanje slikovnega gradiva

- Težavam se izognemo tako, da ne urejamo strukture, če je prezapletena
- Pri slikovnem gradivu, to pomeni delo s samo sliko
- Slika je predstavljena kot polje točkic
- Orodja: *gimp*, *xv*, *slikar* (*paint brush*), ...

Predavanje: 10

UPO 2006/07
© Andrej Brodnik

6

Urejanje slike

- Slika sestoji iz polja (rastra) točkic (*pixel*)
- Če je slika črno-bela ima vsaka točkica prilastek osvetljenosti
- Če je slika barvna ima vsaka točkica prilastek osvetljenosti z rdečo, z zeleno in z modro barvo – *RGB*
- Prilastek osvetljenosti je celo število:
 - 24 bitno
 - 8 bitno

Predavanje: 10

UPO 2006/07
© Andrej Brodnik

7

Urejanje slike

- Primer uporabe 8 bitnega prilastka osvetljenosti je stavljenje HTML strani:
 - `<BODY bgcolor="#84480a">`
 - zapis je šestnajstiški *#RRGGBB*
- Vse ostale barve dobimo kot linerano kombinacijo teh treh barv
 - imamo tri-razsežnostni prostor z normalnimi vektorji *R*, *G*, in *B*
 - ostale vektorje zapišemo kot linearno kombinacijo le-teh

Predavanje: 10

UPO 2006/07
© Andrej Brodnik

8

Barve in *RGB*

- Osnovne barve:
 - bela (255, 255, 255) in črna (0, 0, 0)
 - (255, 0, 0) **rdeča**
 - (255, 255, 0) **rumena**
 - (0, 255, 0) **zelená**
 - (0, 255, 255) **cian**
 - (0, 0, 255) **modra**
 - (255, 0, 255) **magenta**

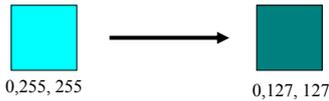
Predavanje: 10

UPO 2006/07
© Andrej Brodnik

9

Barve in *RGB*

- Druge barve dobimo s preostalimi kombinacijami RGB komponent
- Barve lahko obledimo ali potemnimo – vsako komponento zmanjšamo za enak delež



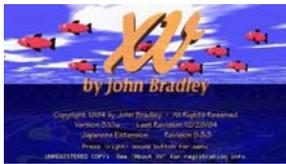
Predavanje: 10

UPO 2006/07
© Andrej Brodnik

10

Primer RGB barv

- Program xv:



Predavanje: 10

UPO 2006/07
© Andrej Brodnik

11

Barve

- Kako smo se pri likovnem pouku učili o barvah?
- Nekako tako, da sta rumena in modra komplementarni barvi ter podobno rdeča in modra (modrikasta)
- Tudi točkicam lahko barve dodeljujemo po podobnem postopku

Predavanje: 10

UPO 2006/07
© Andrej Brodnik

12

Barve in HSV

- Barve razvrstimo na **barvni krog** od 0° do 360° in potem izbiramo barve po njem
 - 0° [255, 0, 0] – rdeča
 - 60° [255, 255, 0] – rumena
 - 120° [0, 255, 0] – zelena
 - 180° [0, 255, 255] – cian
 - 240° [0, 0, 255] – modra
 - 300° [255, 0, 255] – magenta
- Govorimo o barvi (*hue*)

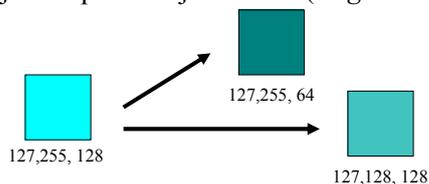
Predavanje: 10

UPO 2006/07
© Andrej Brodnik

13

Barve in HSV

- Vsaka barva je lahko bolj ali manj polna (bleda) – *saturation*
- Tretja komponenta je svetlost (*brightness*)



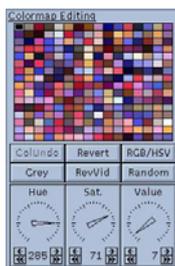
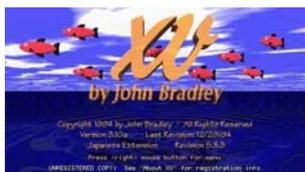
Predavanje: 10

UPO 2006/07
© Andrej Brodnik

14

Primer HSV barv

- Ponovno program xv:



Predavanje: 10

UPO 2006/07
© Andrej Brodnik

15

Naloga

1. Poiščite na spletu program *xv*
2. Preberite si njegovo dokumentacijo (razdelek 5 o *Urejevalniku barv*)
3. Dodatna naloga: namestite program *xv* na študentske računalnike
 - Za pomoč sprašujte na e-spisek
 - Preko e-spiska tudi izmenjujte informacije o iskanju, o nameščanju, o izpisovanju

Predavanje: 10

UPO 2006/07
© Andrej Brodnik

16

Naloga - namigi

- Poiščite najprej program *xv*
- Dokumentacija bo v PostScript obliki, zato potrebujete program, ki iz PS dokumenta izloči samo nekaj strani (*pstops* in podobni programi)
- Za nameščanje potrebujete paket (glej *Red Hat Package Manager*)

Predavanje: 10

UPO 2006/07
© Andrej Brodnik

17

Uporabniška programska oprema

2006/2007
Iztok Savnik

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

1

Viri

- Peter Peer, Andrej Brodnik
– prosojnice za 2003/04
- Raghu Ramakrishnan
– Knjiga: Database Management Systems
– Prosojnice: Dan Suci
- <http://www.w3.org/XML>
<http://www.w3schools.com/>
- <http://www.informatik.uni-freiburg.de/~dbis/lehre/seminar-ws9900/>

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

2

Ozadje

- 1978 - ANSI (*American National Standards Institute*) - ustanovljena skupina za pripravo standarda jezika za opis besedil.
- 1984 - ANSI in ISO (*International Standards Organization*) - priprava mednarodnega standarda. Osnutek je bil objavljen leta 1985, naslednje leto pa tudi sam standard SGML.

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

3

Ozadje

- SGML je bil že med nastajanjem in ob samem začetku podprt z dvema večjima projektoma:
 - 1983-1987 je delovna skupina pri *Association of American Publishers* pripravila v SGML opise zvrsti *knjiga, časopis in članek* – **EMP** (Electronic Manuscript Project)
 - 1987 *US DoD* zahteva enotno tehnično dokumentacijo vseh naročenih izdelkov – **CALS** (Continuous Acquisition and Life-cycle Support)
- 1990 – nov projekt – **HTML** (Tim Berners-Lee uporabi SGML in zgradi DTD (specifična množica označb) za hipertekst – *podobno lahko obravnavamo LaTeX, pdflatex, html2latex,...* kot ~SGML/XML s specifičnim DTD)

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

4

SGML - HTML

- HTML – specifična fiksna množica značk
- HTML – opis prezentacije
- SGML - ločuje med **obliko in vsebino**
- SGML - objavljanje (*publishing*)
- SGML – elektronski dokumenti, arhivi, knjižnice

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

5

Zakaj XML ?

- Kompleksnost SGML
- Definicija strukture dokumenta/podatkov
 - HTML ne omogoča
- Jezik za opis podatkov
- Prenos podatkov med aplikacijami (v organizacijah, med organizacijami)
- Shranjevanje/prenos podatkov na spletu

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

6

XML – zahteve

1. XML mora biti neposredno uporaben na spletu
2. XML mora podpirati veliko množico uporab
3. XML mora biti skladen s SGML
4. Pisanje programov, ki procesirajo XML dokumente, mora biti preprosto
5. Število dodatnih (opsijskih) elementov XML naj bo minimalno, idealno nič
6. XML dokumenti morajo biti berljivi in jasni
7. Priprava modela XML jezika mora biti hitra
8. XML mora biti formalen in jedrnat Priprava XML dokumentov mora biti preprosta
9. Razumljivost XML dokumentov je bistvenega pomena (in zgoščenost minimalnega pomena)

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

7

<Tag – značka>

- Očitno: označevalni jeziki uporabljajo značke
- Ideja označevanja je zelo stara: presledki (*white spaces*), označevanje začetka in konca stavka, odstavki, ..., uredniški popravki,

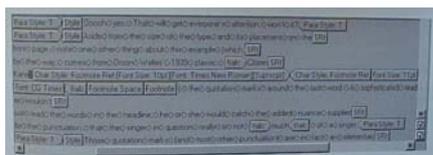
Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

8

<Tag – značka>

- dandanašnje značke,...



WordPerfect 7 *Reveal Codes* okno

WordStar, LaTeX, Word, HTML,...

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

9

<Tag – značka>

- Značke:
 - Postopkovno označevanje: značka določa **kaj se zgodi** z označenim delom besedila (WordStar, WordPerfect, Word, PS, RTF,...)
 - Vsebinsko označevanje:
 - opisno – oblikovanje glede na izbran stil ((La)TeX, Word, ...)
 - posplošeno – določitev značilnih sestavin in njihove medsebojne povezanosti – tvorimo *znakovno podatkovno bazo* (SGML, XML)

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

10

<Tag – značka>

- SGML navodilo: značka se naj začne z znakom **<** in konča z znakom **>**
- Zakaj tekstovne značke? Večja prenosljivost med različnimi okolji! Boljša berljivost!

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

11

XML

- XML je namenjen
 - Strukturiranju,
 - shranjevanju in
 - prenosu podatkov
- XML opisuje podatke in zveze med podatki
- XML nima definiranih značk
 - značke definiramo sami - slovnica
 - XML je množica podobnih jezikov (prim. kontekstno neodvisni jeziki)
- XML uporablja DTD ali XML Schema za opis strukture podatkov - **slovnice**

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

12

XML – zgradba jezika

- **Struktura + Vsebina**
- DTD (*Document Type Definition* – zvrst spisa) definira **strukturo** zvrsti dokumenta
 - slovnica G in organizacija značk
- XML dokument, ki uporablja te značke za označevanje **vsebine**
 - beseda w , ki je v jeziku $L(DTD)$

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

13

XML – struktura dokumenta

1. Uvod (*Prolog*) – ukazi za definiranje razpoznavalnika (*parser*)
 - slovnica, G
2. *Telo* – vsebina, pomembna za ljudi
 - beseda w
3. Zaključek (*Epilog*) – zaključni komentarji

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

14

XML - jezik

- XML dokument je beseda w v jeziku, ki ga definira slovnica DTD $G=(\Sigma, \Gamma, S, P)$:
 - vsak dokument ima korenski element (S)
 - med elementi veljajo odnosi kot jih definirajo produkcije P (drevo izpeljave)
 - simbole ($X \in \Gamma$) imenujemo tudi imenujejo **veje**, črke ($a \in \Sigma$) pa **listi**
 - elementi lahko vsebujejo attribute ali prilastke, ki jim dodajo funkcionalnost (npr. podajajo opise, omogočajo iskanje ipd.)
- Napake v besedilu bodo programsko opremo ustavile
 - prevajanje
 - do napake pride, če $w \notin L(DTD)$

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

15

XML - osnovni gradniki

- Elementi
 - Značke – opisujejo podatkovni objekt
- Atributi (prilastki)
 - Opisujejo lastnosti objektov - elementov
- Entitete
 - Deli skupnega teksta
- Komentarji

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

16

XML - primer

```
<?xml version="1.0"?>
<!-- File Name: Inventory.xml -->

<inventory>
  <book genre="comp" id="b724">
    <title>Database Management Systems</title>
    <author><firstname>Raghu</firstname>
      <lastname>Ramakrishnan</lastname>
    </author>
    <publisher>McGraw Hill</publisher>
    <year>2000</year>
  </book>
  ...
</inventory>
```

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

17

XML - osnovna pravila

1. Vsebuje deklaracijo dokumenta
2. Vsaka začetna značka mora imeti tudi pripadajočo končno značko
3. Strukture morajo biti pravilno gnezdene
4. Vrednosti atributov so navedene v narekovajih
5. Na najvišjem nivoju je lahko le en element – korenski element (S)

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

18

XML - veljaven dokument

- XML dokument je **dobro oblikovan**, če veljajo v njem osnovna pravila
 - Ima pravo obliko in strukturo
 - Preverjanje vsebine (telo XML dokumenta)
- XML dokument je **veljaven**, če:
 - a) je dobro oblikovan in
 - b) njegova struktura ustreza slovnici (DTD definiciji)
 - Preverjanje DTD in XML
 - Beseda jezika L (DTD)

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

19

XML - atribut

- Alternativni način opisa informacij
 - Opis lastnosti elementa
- Vrednost atributa je v narekovajih ""

```
<book genre="comp">
```

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

20

XML - identifikatorji

- Zgolj oblikovni gradnik
- Veljajo znotraj celotnega dokumenta

```
<person id="o555"> <name>Janez</name> </person>  
<person id="o456"> <name>Marija</name>  
  <children idref="o123 o555"/>  
</person>  
<person id="o123" mother="o456"><name>Tone</name>  
</person>
```

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

21

DTD

- Document Type Descriptor
- Podedovan od SGML
- Podoben DB shemi, čeprav ni zares ...
- BNF slovnica (prim. kontekstno neodvisni jezik)
- Definicija elementov specifičnega XML jezika

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

22

DTD - primer

```
<!element book (title,author*,publisher,year) >
<!element title #PCDATA >
<!element publisher #PCDATA >
<!element year #PCDATA >
<!element author (firstname,lastname,address?,age?) >
<!attlist book id ID #required >
<!attlist book genre CDATA #required >
...

```

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

23

DTD - zgradba

- Elementi
 - <!ELEMENT element-name category>
 - <!ELEMENT element-name (element-content)>
 - E1,E2,... ; E* ; E+ ; E? ; E1 | E2
 - Tipi: #PCDATA, EMPTY, ANY
- Atributi
 - <!ATTLIST el-name attr-name attr-type default-value>
 - Tipi: CDATA, (en1|en2|...), ID, IDREF, ENTITY,...
 - Določila: #required, #implied, #fixed
- Entitete
 - <!ENTITY entity-name "entity-value">
 - <!ENTITY writer "Donald Duck."> -- uporaba: &writer;

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

24

DTD+XML vs. Relacije

- XML
 - Primeren bolj za dokumente kot podatke
 - Ime in tip elementov so povezani globalno
 - Ni podpore podatkovnim tipom
 - Ni validacije podatkov
 - Imamo lahko en sam ključ
 - Ni ključev z več atributi
 - Ni tujih ključev (reference na ključe)
 - Ni omejitev za IDREF
 - Ni podpore za ponovno uporabo strukture
 - OO strukture niso podprte
- Ni mogoče ohraniti informacije pri prevajanju iz relacijskega podatkovnega modela v XML

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

25

XML Schema

- XML oblika DB sheme !
- Imena elementov in tipi so povezani lokalno
- Vsebuje osnovne podatkovne tipe (integer, string, date, itn.)
- Podpira omejitve vrednosti (integer > 100)
- Uporabniški podatkovni tipi
- Dedovanje (extension, restriction)
- Tuji ključi

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

26

XML Schema - primer

```
<schema version="1.0"
  xmlns="http://www.w3.org/1999/XMLSchema">
  <element name="author" type="string" />
  <element name="date" type="date" />
  <element name="abstract">
    <type>
      ...
    </type>
  </element>
  <element name="paper">
    <type>
      <attribute name="keywords" type="string"/>
      <element ref="author" minOccurs="0" maxOccurs="*" />
      <element ref="date" />
      <element ref="abstract" minOccurs="0" maxOccurs="1" />
      <element ref="body" />
    </type>
  </element>
</schema>
```

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

27

XML – pomembni standardi

- CSS,XSL/XSLT:
 - Presentacija in transformacija dokumentov in podatkov
- RDF: resource description framework
 - Meta-info, kategorije, semantične mreže, ...
- Xpath/Xpointer/Xlink:
 - standard za povezovanje dokumentov in elementov
- Namespaces:
 - Delo z imeni
- DOM: "Document Object Model"
 - Delo z XML dokumenti v prog. jezikih
- SAX: Enostaven API za XML
- XQL,Xquery, XML-QL: poizvedovalni jezik

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

28

XQL

- Osnova
 - W3C standard
 - Rezultati izrazov so množice objektov
 - `./author, author, //author, /bookstore`
- Dostop do otrok
 - Elementi: `author/firstname, bookstore//title, bookstore/*title`
 - Atributi: `price/@exchange, price/@exchange/total, book[@style]`
- Filtri
 - `book[excerpt], book[author/degree]`
 - `author[degree and award]`
 - `author[not$ (degree or award) and publication]`
 - `author[any lastname = 'Bob']`
- Funkcije
 - `author[first-name!text() = 'Bob'], author[*!text() = 'Bob']`

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

29

XML-QL – primer 1

- Izpiši avtorje in naslove vseh knjig založbe Addison-Wesley.

```
WHERE <book>
  <publisher>Addison-Wesley</publisher>
  <title>$t</title>
  <author>$a</title>
</book> IN www.bookstore.com/books.xml
CONSTRUCT <result>
  <author> $a</author>
  <title> $t</title>
</result>
```

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

30

XML-QL – primer 2

- Izpiši leta in priimke avtorjev, ki so izdali knjigo na to leto.

```
WHERE <book> $e </book> IN www.bookstore.com/books.xml,
  <author> $n </author>,
  <year> $p </year> IN $e
CONSTRUCT <result> <year> $p </year>
  WHERE <lastname> $l </lastname> IN $n
CONSTRUCT <lastname> $l </lastname>
</result>
```

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

31

Uporaba Cascading Style Sheet

```
<?xml version="1.0"?>
<!-- File Name: Inventory02.xml -->
<?xml-stylesheet type="text/css" href="Inventory02.css"?>
<INVENTORY>
  <BOOK>
    <TITLE>The Adventures of Huckleberry Finn</TITLE>
    <AUTHOR>Mark Twain</AUTHOR>
    <BINDING>mass market paperback</BINDING>
    <PAGES>298</PAGES>
    <PRICE>$5.49</PRICE>
  </BOOK>
  <BOOK>
    <TITLE>Leaves of Grass</TITLE>
    <AUTHOR>Walt Whitman</AUTHOR>
    <BINDING>hardcover</BINDING>
    <PAGES>462</PAGES>
    <PRICE>$7.75</PRICE>
  </BOOK>
  ...
</INVENTORY>
```

procesni ukaz

```
/* File Name: Inventory02.css */
BOOK
{display: block;
 margin-top: 12pt;
 font-size: 10pt;}
TITLE
{display: block;
 font-size: 12pt;
 font-weight: bold;
 font-style: italic;}
AUTHOR
{display: block;
 margin-left: 15pt;
 font-weight: bold;}
BINDING
{display: block;
 margin-left: 15pt;}
PAGES
{display: none;}
PRICE
{display: block;
 margin-left: 15pt;}
```

definicija imenskega prostora

vzorec se nanaša na cel XML dokument

upoštevanje XML pravil

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

32

eXtensible Stylesheet Language

```
<?xml version="1.0"?>
<!-- File Name: XslDemo02.xml -->
<xml-stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl"
  xsl:template match="/">
  <xsl:for-each select="INVENTORY/BOOK">
    <SPAN STYLE="font-style:italic">Title: </SPAN>
    <xsl.value-of select="TITLE"/><BR />
    <SPAN STYLE="font-style:italic">Author: </SPAN>
    <xsl.value-of select="AUTHOR"/><BR />
    <SPAN STYLE="font-style:italic">Binding type: </SPAN>
    <xsl.value-of select="BINDING"/><BR />
    <SPAN STYLE="font-style:italic">Number of pages: </SPAN>
    <xsl.value-of select="PAGES"/><BR />
    <SPAN STYLE="font-style:italic">Price: </SPAN>
    <xsl.value-of select="PRICE"/><P />
  </xsl:for-each>
</xsl:template>
</xsl:stylesheet>
```

XSL ID

definicija imenskega prostora

vzorec se nanaša na cel XML dokument

upoštevanje XML pravil

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

33

- Omogoča izločanje in sortiranje podatkov
- Dosegljiv je vsak podatek, četudi je podan s prilastkom

Zaključek

- "Proof by Google"
 - database+XML – 13,500,000 pages ;-)
- Podjetja uvajajo XML kot medij za prenos podatkov med oddelki
- Sistemi (distribuirani) uporabljajo XML kot medij za prenos podatkov
- Konfiguracijske datoteke, logi, ... sistemov so pogosto v XML
- Večina DBMS ima XML vmesnik

Predavanje: 11

UPO 2006/07
© Andrej Brodnik, Iztok Savnik

34

Uporabniška programska oprema

2006/2007
Iztok Savnik

Predavanje: 12

UPO 2006/07
© Andrej Brodnik

1

Matematični problemi

- Za reševanje matematičnih problemov smo doslej spoznali:
 - preglednice
 - analitično reševanje problemov
- Reševanje numeričnih problemov
 - do neke mere lahko s preglednicami
 - ali z orodji za analitično reševanje problemov
 - toda, tudi lahko v Javi!-)

Predavanje: 12

UPO 2006/07
© Andrej Brodnik

2

Matlab

- Matlab je orodje, ki **ne** omogoča reševanja analitičnih matematičnih problemov
- je prikladen za reševanje številskih problemov
- uporabljali bomo brezplačno različico *Octave*, ki je skoraj povsem primerljiva s pravim Matlab-om
- vse primere iz knjige lahko rešimo z octave

Predavanje: 12

UPO 2006/07
© Andrej Brodnik

3

Osnovne strukture

- Matlab pozna nekaj osnovnih struktur (v programskem jeziku bi rekli *tipov*):
 - realna in kompleksna števila
 - vektorji in matrike
 - polinomi
- značilnost struktur je, da so zanje na voljo tudi operacije

Predavanje: 12

UPO 2006/07
© Andrej Brodnik

4

Števila

```
octave:1> sqrt(1)
ans = 1
octave:2> sqrt(-1)
ans = 0 + 1i
octave:3> sqrt(i)
ans = 0.70711 + 0.70711i
octave:4> i*i
ans = -1
octave:5> 1/i
ans = 0 - 1i
octave:6> e^i
ans = 0.54030 + 0.84147i
octave:7> (e^i)^2
ans = -0.41615 + 0.90930i
octave:8> e^(2i)
ans = -0.41615 + 0.90930i
```

Predavanje: 12

UPO 2006/07
© Andrej Brodnik

5

Polinomi

```
octave:9> p = [1 0 -1]
p = 1 0 -1
octave:10> polyval(p, 0)
ans = -1
octave:11> polyval(p, 1)
ans = 0
octave:12> roots(p)
ans = -1 1
octave:13> q = [1 -1]
q = 1 -1
octave:14> r = deconv(p, q)
r = 1 1
octave:15> conv(q, r)
ans = 1 0 -1
octave:16> polyder(p)
ans = 2 0
octave:17> xi = [0 2 4]
xi = 0 2 4
octave:18> yi = [0 4 16]
yi = 0 4 16
octave:19> polyfit(xi, yi, 2)
ans = 1 0 0
octave:20> polyfit(xi, yi, 3)
ans = 5.9406e-02 6.4356e-01
4.7525e-01 -2.3870e-15
octave:21> polyfit(xi, yi, 1)
ans = 4.0000 -1.3333
```

Predavanje: 12

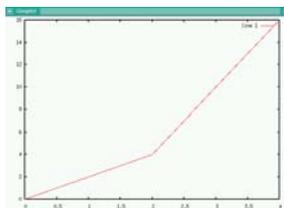
UPO 2006/07
© Andrej Brodnik

6

Risanje

- rišemo lahko samo točke in ne funkcij

```
octave:25> plot(xi, yi)
```



Predavanje: 12

UPO 2006/07
© Andrej Brodnik

7

Risanje

- če želimo narisati funkcijo, moramo napisati program, da izračuna vrednosti funkcije v določenih točkah ter nato narišemo točke
- glej primer v knjigi
- *domača naloga*: napišite program, ki bo narisal polinom

Predavanje: 12

UPO 2006/07
© Andrej Brodnik

8

Vektorji in matrike

- polinom je v bistvu vektor koeficientov pri posameznih potencah
- oba pa sta – seznama
- silno podobno:
 - *Mathematic-i*
 - *Emacs-u*
 - ...
- ponovno smo pri seznamih
- matrika je
 - seznam seznamov

Predavanje: 12

UPO 2006/07
© Andrej Brodnik

9

Vektorji in matrike

- kot pri polinomih je vrsta že pripravljenih operacij
 - seštevanje, odštevanje, množenje
 - obračanje, transpozicija
 - izračun determinante, lastne vrednosti
 - reševanje sistema enačb
 - ...
- primeri so v knjigi, tu si oglejmo samo reševanje sistema enačb

Predavanje: 12

UPO 2006/07
© Andrej Brodnik

10

Reševanje sistema enačb

$$4x - 3y + 2z = 15$$

$$3x + y - z = -7$$

$$21x - 6y + 3z = 24$$

```
octave:15> A = [4 3 2; 3 1 -1; 21 6 3];
```

```
octave:16> y = [15 -7 24];
```

```
octave:17> A\y
```

```
error: operator \: nonconformant arguments (op1 is 3x3, op2  
is 1x3)
```

```
error: evaluating binary operator '\\' near line 17, column 2
```

```
octave:17> y = [15; -7; 24];
```

```
octave:18> A\y
```

```
ans = 0.10000 0.00000 7.30000
```

Predavanje: 12

UPO 2006/07
© Andrej Brodnik

11

Uporabniška programska oprema

2006/2007

Matjaž Kljun, Andrej Brodnik

Predavanje: 13

UPO 2006/07
© Matjaž Kljun, Andrej Brodnik

1

CMS

- CMS – Content Management System
Sistem za ravnanje z vsebinami
- WCM – Web Content Management

Predavanje: 13

UPO 2006/07
© Matjaž Kljun, Andrej Brodnik

2

Programska oprema za izdelavo,
spreminjanje, odstranjevanje virov
informacij iz organiziranega repozitorija.

V grobem vsebuje orodja za objavo,
upravljanje oblike, indeksiranje, iskanje in
dosego informacij.

Predavanje: 13

UPO 2006/07
© Matjaž Kljun, Andrej Brodnik

3

Zgodovina

- Interni razvoj v organizacijah - več ljudi upravlja z ogromno količino informacij
- 1995 Cnet ponudi svoje interni sistem v ločenem podjetju Vignette.
- 1998 Pencom objavi Metaphoria Data Transformation system – Java servlet (med podatki in strežnikom omogoča):
 - ista vsebina na različnih straneh (reklamiranje, grafična podoba) glede na brskalnik, naslov dostopa, ...
 - dobava vsebine glede na pravice in geografsko lego (različni jeziki), ...
 - shranjevanje profilov uporabnikov (interesi želje, način plačila, ...)
- V začetku stoletja podjetja ponudijo trgu weblog sisteme in CMS orodja približajo množicam.

Predavanje: 13

UPO 2006/07
© Matjaž Kljun, Andrej Brodnik

4

CMS vloge

- Na uporabniški strani je uporabnik – lahko dodaja vsebine ali ne.
- Na vzdrževalčevi strani imamo 3 skupine:
 - Uredniki vsebin (kaj in kje se objavi)
 - Objavitelji vsebin
 - Pisci vsebin
- Meja med vzdrževalčevim in uporabniškim delom spletnega sestavka?

Predavanje: 13

UPO 2006/07
© Matjaž Kljun, Andrej Brodnik

5

CMS Delitev

- Po tehnologiji:
 - Pull CMS
 - Push CMS
- Po namembnosti: glede na naravo vsebine, ki jo predstavljajo.

Predavanje: 13

UPO 2006/07
© Matjaž Kljun, Andrej Brodnik

6

Pull CMS in Push CMS

- Grajenje strani v času povpraševanja, CMS razdeljen na dva dela.
- Strani grajene v času izdelave, CMS je samo na vzdrževalčevi strani.



Predavanje: 13

UPO 2006/07
© Matjaž Kljun, Andrej Brodnik

7

Prednosti in slabosti

	Push CMS	Pull CMS	Rešitev
Prijazni URL naslovi (uporabnik, iskalniki)	http://www.pef.upr.si/urnik/mara.html	http://www.pef.upr.si/vsebina.php?id_urnik=287	URL rewriters. Kateri iskalnik ne sledi več dinamičnim naslovom?
Objava vsebine na več strežnikih	Preprosto (ftp, vpn, sftp, ...) – podatki o strežnikih na enem mestu	Težje – informacije odvisne od strežnika in programske opreme (baza, skriptni jezik, ...)	Zakaj bi imeli isto vsebino na dveh mestih?
Neodvisnost vsebine od CMS	Lahko arhiviranje, prenašanje.		Zakaj bi prenašal spletno stran iz strežnika na strežnik?
Hitra nadgradnja (več strežnikov)	Samo na enem mestu	Vsak strežnik posebej	Skripte za samodejno posodabljanje.
Optimizacija spletnega strežnika	Hiter prikaz datotek, manjša obremenitev strežnika	Večja obremenitev strežnika	Predpomnjenje (Caching)
Dinamičnost vsebin	Popolni Push CMS je ne omogoča.	Vsebinska popolnoma dinamična.	

Predavanje: 13

UPO 2006/07
© Matjaž Kljun, Andrej Brodnik

8

Namembnost

- Web content management systems (WCMS) – avtomatizacija vsebine na spletnih straneh, spletni portali.
- Transactional content management systems (T-CMS) - podpora in upravljanje elektronskih transakcij.
- Integrated content management systems (I-CMS) – podpora in upravljanje dokumentov in vsebin podjetja.
- Publications management systems (P-CMS) – podpora in upravljanje življenjskega kroga publikacij (pripočnike, knjige, napotki, viri, ...).
- Learning management systems (L-CMS) – upravljanje z učnimi vsebinami na spletu.
- Document imaging systems – spletne galerije, upravljanje s slikovnimi dokumenti, ...
- Enterprise (E-CMS) se delijo po namembnosti: ponavadi so skupek dveh ali več omenjenih CMSjev (WCMS, TCMS, ICMS). Največkrat so lahko tudi za požarnimi zidovi podjetij in nudijo podporo prenosu informacij v podjetju (transakcije, kupci, nakupi, informacije, ...)

Predavanje: 13

UPO 2006/07
© Matjaž Kljun, Andrej Brodnik

9

LMS

- Oddaljeno izobraževanje ali Dodatek k standardnim predavanjem.
 - Pomoč učiteljem pri izdelavi in vzdrževanju vsebin določenega predmeta.
 - Pomoč učencem pri poglobljanju vsebine predmeta (učbeniki, testi, ...).

Predavanje: 13

UPO 2006/07
© Matjaž Kljun, Andrej Brodnik

10

LMS – prenos vsebin

Prenos vsebine enega predmeta med različnimi sistemi žal ni preprosto!

Dve možnosti:

- Izdelava standardov za učne objekte:
 - SCORM Shareable content object reference model (US department of defense)
 - IMS Global Content Packaging
- Uporaba iste platforme za e-izobraževanje.

Predavanje: 13

UPO 2006/07
© Matjaž Kljun, Andrej Brodnik

11

LMS mora vsebovati:

- Spisek predmetov
- Administrativne informacije predmetov (urnik, lokacija, literatura, kako do pomoči, ...).
- Sistem novic po posameznega predmeta.
- Registracija študentov, sledenje in plačilni sistem.
- Materiali za predavanja (celotna vsebina, izseki, prosojnice, ...) – odvisno od namena
- Dodatni viri informacij (povezave na psletne strani, knjižnice, ...)
- Kvizi za samoocenjevanje študentov
- Testi za ocenjevanje študentov
- Komunikacija na različnih ravneh – sinhrona (klepetalnica) in asinhrona (e-pošta, forum)
- Sistem pravic za študente in predavatelje
- Izdelava dokumentacije in statistike za sledenje kvalitete
- Preprosta orodja za izdelavo vsebin.

Predavanje: 13

UPO 2006/07
© Matjaž Kljun, Andrej Brodnik

12

LMS in podobni izrazi

- E-izobraževanje
- CMS: course management system
- MLE: managed learning environment
- CMC: computer mediated communication
- Online education
- VLE: virtual learning environment
- LSS: learning support system
- LP: learning platform

Predavanje: 13

UPO 2006/07
© Matjaž Kljun, Andrej Brodnik

13

Prednosti

- Omogoča ne tehnično podkovanim avtorjem in urednikom preprosto in hitro objavljane vsebin na spletu
- Omogoča jasn in natančen sistem pisanja in objavljanja s pomočjo pravic, ki jih ima posameznik v sistemu.
- Prihranek časa in denarja pri izobraževanju tehnično podkovanih kadrov.
- Zmanjšanje časa od pisanja vsebine do njene objave
- Hitra menjava predlog spletnih strani
- ...

Predavanje: 13

UPO 2006/07
© Matjaž Kljun, Andrej Brodnik

14

INTERNACIONALIZACIJA - Postopek prevajanja programov

Ukazna vrstica

1. Začetni primer (primer-zacetni.c)

```
#include <stdio.h>

void main(int argc, char ** argv){

    printf (gettext("Demo message for gettext\n"));
    printf (gettext("Another demo message for gettext\n"));
    printf (gettext("This is the %1$s command with process id:
                    %2$d\n"), argv[0],getpid());
    exit(0);
}
```

2. Obdelava začetnega primera: vključitev podpore i18n (primer.c)

```
#include <stdio.h>
#include <locale.h>
#include <libintl.h>

#define PACKAGE "primer"
#define LOCALEDIR "locale"

void main(int argc, char ** argv){

    setlocale (LC_ALL,"");
    bindtextdomain (PACKAGE, LOCALEDIR);
    textdomain (PACKAGE);

    printf (gettext("Demo message for gettext\n"));
    printf (gettext("Another demo message for gettext\n"));
    printf (gettext("This is the %1$s command with process id:
                    %2$d\n"), argv[0],getpid());

}
```

3. Izdelava kataloga sporočil: **xgettext primer.c**

Rezultat je datoteka: **messages.po**

4. Dejanski prevod kataloga (rezultat shranimo v datoteko **messages-sl.po**)

```
# translation of messages.po to
# This file is distributed under the same license as the PACKAGE
package.
# Copyright (C) YEAR THE PACKAGE'S COPYRIGHT HOLDER
# <tamara@kp.pef.uni-lj.si>, 2004
#
msgid ""
msgstr ""
"Project-Id-Version: messages\n"
"POT-Creation-Date: 2004-11-09 12:07+0100\n"
"PO-Revision-Date: 2004-11-09 12:19+0100\n"
"Last-Translator: <tamara@kp.pef.uni-lj.si>\n"
"Language-Team: <sl@li.org>\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=UTF-8\n"
"Content-Transfer-Encoding: 8bit\n"
"X-Generator: KBabel 1.0\n"

#: primer.c:14
msgid "Demo message for gettext\n"
msgstr "Moj prvi prevod (Tamara).\n"

#: primer.c:15
msgid "Another demo message for gettext\n"
msgstr "Drugi prevod (Tamara)\n"

#: primer.c:16
#, c-format
msgid "This is the %1$s command with process id: %2$d\n"
msgstr "Moje tretje sporočilo: %1 ukaz za proces z identifikacijo
%2\n"
```

5. Priprava jedrnatejšega kataloga: **msgfmt -o primer messages-sl.po**

6. Namestitev oz. umestitev kataloga na pravo lokacijo v datotečnem sistemu:
cp primer.mo locale/sl_SI.UTF-8/LC_MESSAGES

7. Prevajanje začetnega programa s podporo i18n: **cc -o primer primer.c**

8. Preizkus delovanja:

```
>export LC_ALL=sl_SI.UTF-8
>./primer
```

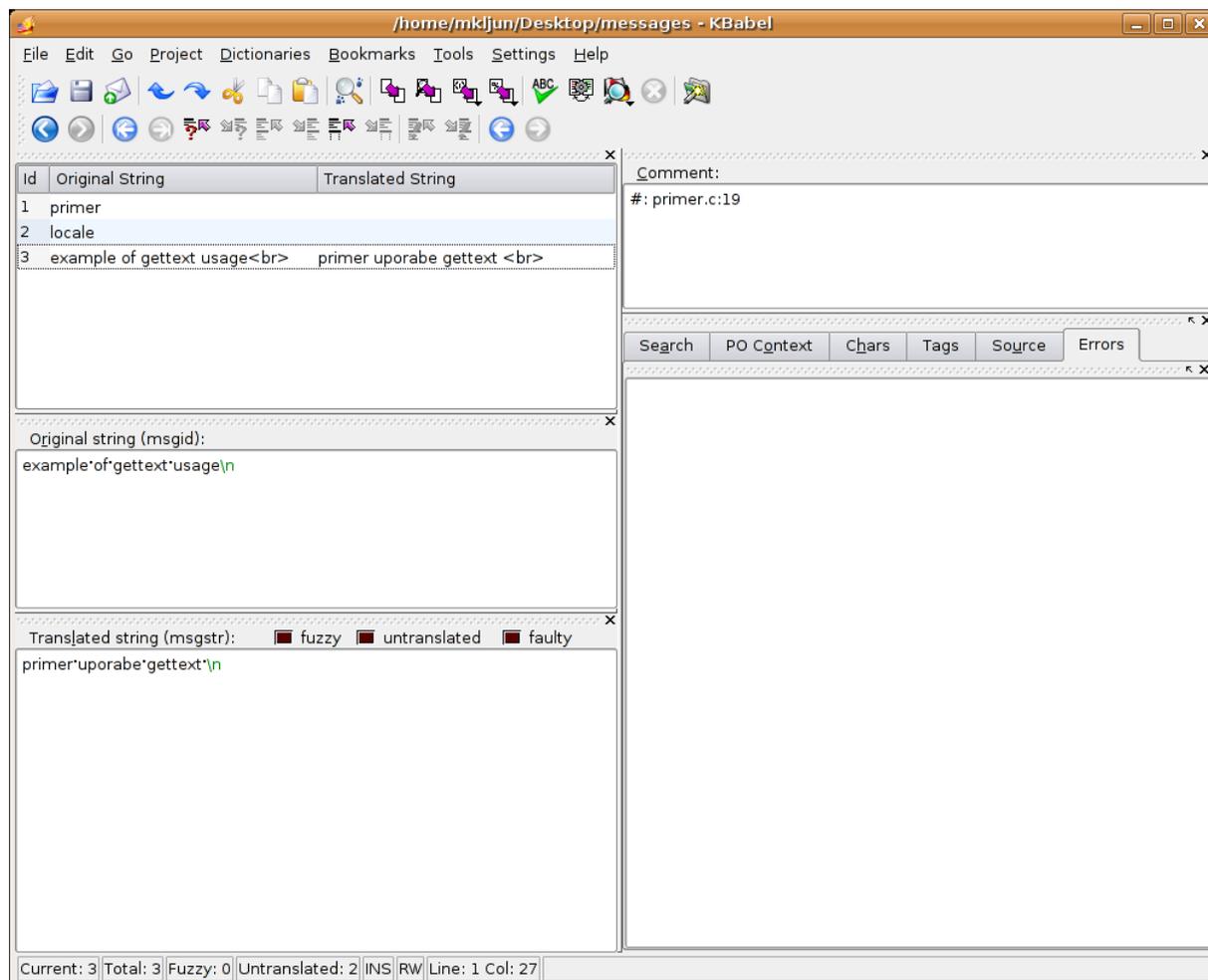
```
>export LC_ALL=en_EN
>./primer
```

Prevajanje z grafičnimi orodji

Vsa orodja uporabljajo gettext kot osnovo (glej prvi del vaj).

Kbabel - <http://kbabel.kde.org/>

Gtranslator - <http://gtranslator.sourceforge.net/>
 poEdit - <http://www.poedit.org/>



Še zanimiv članek:

<http://programming.newsforge.com/article.pl?sid=05/03/16/206232&tid=140>

DELO Z BESEDILI - Razumevanje strukture besedila

Za obdelavo besedil imamo na voljo več uporabniških orodij oz. programov, zato lahko besedila oblikujemo in pripravljamo na različne načine. Na predavanjih in vajah smo se v glavnem srečali s tremi različnimi orodji oz. okolji: OpenOffice Writer, LaTeX, HTML.

1. OpenOffice Writer

OOo Writer je del pisarniškega paketa OpenOffice, ki omogoča vizuelno urejanje in obdelavo besedil. Podoben je drugim komercialnim izdelkom na področju pisarniškega poslovanja (npr. Microsoft Office), ki ga lahko namestimo na različne operacijske sisteme (npr. Microsoft Windows, Linux, Solaris).

Uporabljamo ga preko zmogljivega uporabniškega vmsnika, ki nam omogoča dostop do posameznih funkcij za oblikovanje, spreminjanje skratka delo z besedilom. Do posameznih ukazov lahko dostopamo preko menija ali gumbov – ikon v orodni vrstici.

Najpomembnejši ukazi za delo z dokumentom:

- odpiranje datoteke z besedilom: File → Open
- ustvarjanje nove datoteke z besedilom: File → New
- shranjevanje datoteke z besedilom: File → Save ali File → Save as
- zapiranje datoteke z besedilom: File → Close
- izhod iz programa: File → Exit

Najpogostejši ukazi za oblikovanje pisave:

- krepki izpis: Format → Character → Bold (ali ikona **B** v orodni vrstici)
- ležeči izpis: Format → Character → Italic (ali ikona *i* v orodni vrstici)
- podčrtani izpis: Format → Character → Font effects → Underlining (ali ikona U v orodni vrstici)
- obojestranska poravnava besedila: Format → Paragraph → Alignment (ali ikona  v orodni vrstici)
- leva poravnava besedila: Format → Paragraph → Alignment (ali ikona  v orodni vrstici)
- desna poravnava besedila: Format → Paragraph → Alignment (ali ikona  v orodni vrstici)
- sredinska poravnava besedila: Format → Paragraph → Alignment (ali ikona  v orodni vrstici)

2. LaTeX

LaTeX je programsko orodje za logično pisanje besedil, pravzaprav skupek makrojev za programsko orodje TeX, ki nam samo delo s TeXom precej olajša. Za samo pisanje besedil lahko uporabljamo katerikoli enostane urejevalnik besedil (npr. Notepad, WordPad, Emacs, ...), ostali pripomočki nam omogočajo izpis besedil, pripravo lastnih naborov znakov, ipd.

Najpomembnejši ukazi za izdelavo končne oblike dokumenta:

- latex

- dvips
- xdvi

Osnovni ukazi pri definiranju novega dokumenta z besedilom:

- documentclass
- usepackage
- begin{document} ... end{document}

Najpogostejši ukazi za oblikovanje pisave:

- pisanje šumnikov z navednicami ("c) samo s paketom babel, sicer: `\v{c}` ali `\v c`
- krepka pisava: `\bf`
- ležeča pisava: `\it`

V novejših LaTeX paketih se lahko šumnike piše brez posebnih znakov. Primer klicanja paketov s slovenskimi črkami je:

```
\documentclass[a4paper,twocolumn]{article}
\usepackage{graphicx,epsfig}
\usepackage[slovene]{babel}
\usepackage[utf-8]{inputenc}
```

Poravnava besedila (kaj je lepilo in kako deluje)

- središčna poravnava: `\begin{center} ... \end{center}`
- desna poravnava: `\begin{flushright} ... \end{flushright}`
- pomen ukazov `\hfill` in `\vfill`

Windows orodja: MikTeX (Latex za Windows) in urejevalniki Texpad, WinEdt
Linux/Unix orodja: urejevalniki Kile, TexMaker, Lyx

3. HTML

HTML ali HyperText Markup Language je enostaven jezik za pisanje in izdelavo spletnih dokumentov neodvisno. Čeprav na trgu že obstajajo različna programska orodja, ki nam pisanje v HTML obliki precej olajšajo, v osnovi ne potrebujemo ničesar razen razumevanja posameznih ukazov, ki so specifični za HTML, in enostavnega urejevalnika besedil (npr. Notepad, WordPad, Emacs, ...). Napotki pri izdelavi spletnih strani v HTML obliki so dostopni na svetovnem spletu.

Osnovni ukazi pri definiranju novega dokumenta z besedilom in nekaj osnovnih ukazov:

- `<html> <head> ... </head> <body> ... </body> </html>`
- `
 ...` prelom vrstice
- `<p> ...` odstavek

Najpogostejši ukazi za oblikovanje pisave:

- `<h?> ...` velikosti pisav
- `` velikosti pisave
- krepka pisava: ` ... `
- poševna pisava: `<i> ... </i>`
- pisalni stroj: `<tt> ... </tt>`

Poravnava besedila (kaj je lepilo in kako deluje)

- Levo poravnano: `<p align=left> ... </p>`
- Desno poravnano: `<p align=right> ... </p>`
- Sredinsko poravnano: `<p align=center> ... </p>`

Pri urejanju HTML besedil se zadnja leta uporabljajo tudi CSS (cascading style sheets). S tem se vsebina in oblika dokumentov ločita. Vsebina se nahaja v HTML dokumentu in oblika v CSS dokumentu. Ena CSS datoteka lahko definira obliko večjemu številu HTML dokumentov. S tem se oblikovanje velikega števila HTML zelo dokumentov poenostavi.

Zanimive povezave:

1. OpenOffice Writer

- http://documentation.openoffice.org/HOW_TO/index.html (predstavitev in navodila za delo),
- http://www.tutorialsforopenoffice.org/category_index/wordprocessing.html
- <http://www.firstpr.com.au/book-design/> (razvoj, problemi, novosti, ...),

2. LaTeX

- [The Not So Short Introduction to L^AT_EX 2 \$\epsilon\$](#)
- <http://www.silmaril.ie/latex.html>,
- <http://vlado.fmf.uni-lj.si/texceh/texceh.htm> (slovenska skupina uporabnikov TeX-a),
- <http://tug.org/> (TeX Users Group Home Page),
- <http://www.tex.ac.uk/cgi-bin/texfaq2html?introduction=yes> (FAQs),
- <http://members.tripod.com/~upem/tex.html> (TEX and LATEX Resources),
- <http://www.tex.ac.uk/tex-archive/> (CTAN Arhiv (Comprehensive TeX Archive Network)),
- <http://members.tripod.com/~upem/miktex.html> (Links and documentation related to LaTeX, TeX and MikTeX),
- <http://www.ora.com/homepages/jasper/texhelp/BibTeX.html> (Help On BibTeX),
- <http://nl.ijs.si/GNUsl/trubar/trubar.html> (Trubar Spelling Checker),
- http://www.amazon.com/exec/obidos/ASIN/0201398257/o/qid=984389269/sr=8-1/ref=aps_sr_b_1_1/102-6867470-9453721 (A Guide to L^AT_EX),
- http://www.amazon.com/exec/obidos/ASIN/0201529831/qid=984392516/sr=1-2/ref=sc_b_2/102-6867470-9453721 (LaTeX: A Documentation Preparation System User's Guide and Reference Manual).

3. HTML

- http://colos1.fri.uni-lj.si/~sis/computing/HTML/primer_1.html (uvod v HTML),
- <http://www.w3schools.com/html/> (HTML tutorial),
- <http://www.w3schools.com/css/> (CSS tutorial),
- www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html (najpogosteje zastavljena vprašanja o HTML in osnove HTML),
- www.davesite.com/webstation/html/ (osnove HTML),
- <http://www.w3.org/>,
- www.geocities.com/siliconvalley/campus/1924/index.html ,
- members.aol.com/teachemath/create.htm ,
- www.jmarshall.com/easy/html ,
- htmlprimer.com ,

- http://www.ncsu.edu/it/edu/html_trng/html_basics.html ,
- <http://freespace.virgin.net/r.cawley/> .

DELO Z BESEDILI - Urejanje dokumenta

1. OpenOffice Writer

Oblikovanje naslovov (pomen različnih stilov): Format → Stylist

Definiranje avtorja in drugih lastnosti dokumenta:

- File → properties
- Tools → Options → User data

Oblikovanje noge in glave (datumi, številčenje strani)

- Insert → Header, Insert → Footer
- Insert → Fields

Uporaba kazal: Insert → Index and Tables

Shranjevanje besedila v druge oblike (html): File → Save as

2. LaTeX

Priprava dokumenta z: author, title, date:

- `\title{...}`
- `\author{...}`
- `\date{...}`
- `\maketitle`
- `\begin {titlepage} \maketitle \end {titlepage}`

Oblikovanje naslovov:

- `\section in \subsection`
- `\part {...}, \chapter {...}, \section {...}, \paragraph {...}`

Uporaba kazal: `\tableofcontents`

Shranjevanje besedila v druge oblike (pdf, html)

- `pdflatex`
- `latex2html`

3. HTML

Priprava dokumenta (oblikovanje glave):

```
<head> ... <title> ... </title>
<meta name="Author" content="podamo avtorja">
<meta name="Keywords" content="podamo ključne besede dokumenta">
<meta name="Description" content="Kratek opis dokumenta">

<link rel=stylesheet href="oblika.css">
</head>
```

```
<body>  
...  
</body>
```

DELO Z BESEDILI - Urejanje dokumenta

1. OpenOffice Writer

Pisanje matematičnih izrazov: Insert → Object → Formula

Naštevanje elementov: Format → Numbering/Bullets (tudi v orodni vrstici)

Oblikovanje tabel: Insert → Table; Format → Table

Vstavljanje slik (predmetov): Insert → Graphics/Object

Sklicevanje: Insert → Cross-reference

Druge vrste kazal: Insert → Indexes and Tables

2. LaTeX

Pisanje matematičnih izrazov:

- formula v besedilu: $\$ \dots \$$
- formula samostojno: $\$ \$ \dots \$ \$$

Naštevanje elementov

- navadni spisek: $\backslash\begin\{itemize\} \dots \backslash\end\{itemize\}$
- naštevalni spisek: $\backslash\begin\{enumerate\} \dots \backslash\end\{enumerate\}$

Oblikovanje tabel: $\backslash\begin\{tabular\} \{stolpci\} \dots \backslash\end\{tabular\}$

Vstavljanje slik (predmetov): $\backslash\begin\{figure\} [\dots] \dots \backslash\end\{figure\}$

Druge vrste kazal

- Kazala slik: $\backslash\listoffigures$
- Kazala tabel: $\backslash\listoftables$
- Literatura: $\backslash\thebibliography$

Sklicevanje:

- na tabele in predmete: $\backslash\ref\{\dots\}$
- na literaturo: $\backslash\cite\{\dots\}$

3. HTML

Oblikovanje spletne strani (ozadje): $\langle\text{body bgcolor=barva1 text=barva2}\rangle$

Vnos in oblikovanje tabel

$\langle\text{table } \dots \rangle$

$\langle\text{tr}\rangle \dots$ vrstica

$\langle\text{th } \dots \text{ stolpec}\rangle\langle/\text{th}\rangle$

$\langle\text{th } \dots \text{ stolpec}\rangle\langle/\text{th}\rangle$

$\langle\text{th } \dots \text{ stolpec}\rangle\langle/\text{th}\rangle$

```
</tr>
<tr> ... vrstica
    <th ... stolpec</th>
    <th ... stolpec</th>
    <th ... stolpec</th>
</tr>
<tr> ... vrstica
    <th ... stolpec</th>
    <th ... stolpec</th>
    <th ... stolpec</th>
</tr>
</table>
```

Vnos in oblikovanje slik:

Sklicevanje: opis

DELO S PREGLEDNICAMI - Razumevanje dela s preglednicami

OpenOffice Calc

OpenOffice Calc je tako kot OpenOffice Writer del pisarniškega paketa OpenOffice, ki omogoča vizuelno urejanje in obdelavo podatkov v preglednici. Podoben je drugim komercialnim izdelkom na področju pisarniškega poslovanja (npr. Microsoft Office), ki ga lahko namestimo na različne operacijske sisteme (npr. Microsoft Windows, Linux, Solaris).

Uporabljamo ga preko zmogljivega uporabniškega vmsnika, ki nam omogoča dostop do posameznih funkcij za oblikovanje, spreminjanje skratka delo z besedilom. Do posameznih ukazov lahko dostopamo preko menija ali gumbov – ikon v orodni vrstici.

Najpomembnejši ukazi za delo z dokumentom:

- odpiranje datoteke: File → Open
- ustvarjanje nove datoteke: File → New
- shranjevanje datoteke: File → Save ali File → Save as
- zapiranje datoteke: File → Close
- izhod iz programa: File → Exit

Sprehajanje po preglednici nam omogočajo tipke ←, ↑, →, ↓.

Vnos podatkov

- Številске in opisne podatke vnesemo neposredno v celico (niso potrebni kakšni posebni znaki)
- Omogočen je vnos nizvo (fill range).

Sklicevanje

- Relativni sklic: =a1
- Absolutni klic: =\$a\$1
(znak \$ fiksira ali stolpec ali vrstico)

Formule

- Vnos formul začnemo z znakom =
- Ciklanje formul ni dovoljeno, izpiše se opozorilo o napaki.

DELO S PREGLEDNICAMI - Razumevanje dela s preglednicami

OpenOffice Calc

- Oblikovanje celic: preko glavnega menija (Format Cell oz. Uredi Celico) ali pomožnega menija (desni gumb miške)
- Delo s podatki: meni Data oz. Podatki
 - a. Urejanje podatkov
 - b. Filtriranje podatkov
 - c. Delne vsote: zahteva predhodno urejene podatke
- Risanje grafikonov: meni Insert Object Graph

SLIKE – Vektorske slike

Strukturirane (vektorske) slike

Strukturirane slike so predstavljene z množico vektorjev, ki opisujejo posamezne krivulje na sliki. Velikost tovrstnih slik je poljubna.

Programska orodja, ki jih imamo na voljo so:

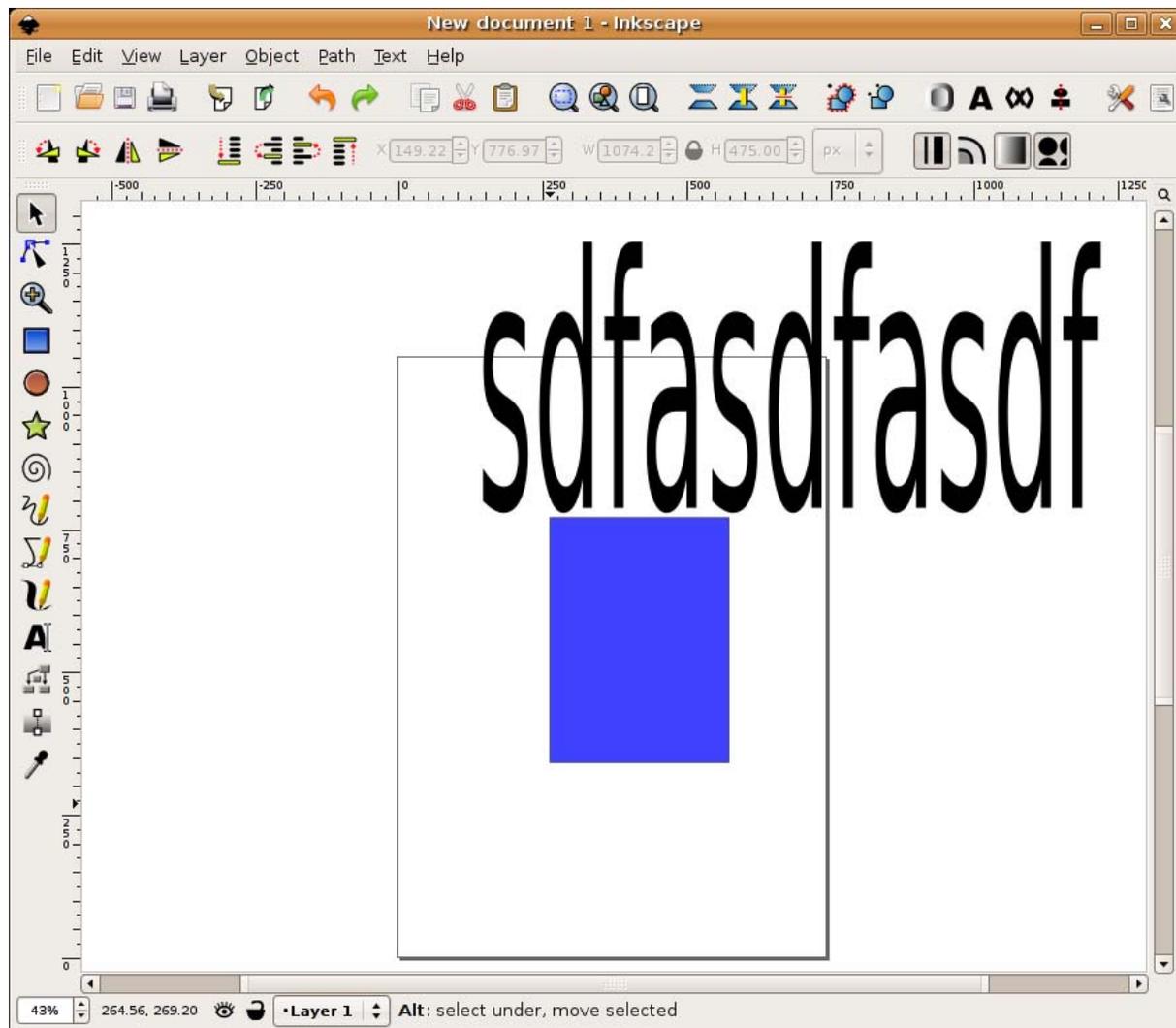
- Xfig – program za risanje, ki zna sliko zapisati tudi v LaTeX formatu
- Inkscape
- OpenOffice Fig
- Corel Draw

Kratke pregled možnosti, ki ji ponuja programsko orodje Xfig

1. Osnovno o programu:
 - a. Zagon: preko pripravljene bližnjice ali iz ukazne lupine z ukazom XFig
 - b. Ukazi dostopni preko menija na vrhu okna
2. Osnovno risanje (ukazi oz. ikone v orodni vrstici)
 - a. Risanje črt
 - b. Risanje poligonov (kvadrati, večkotniki...)
 - c. Risanje krogov in elips
 - d. Vstavljanje besedila
 - e. Vstavljanje zunanjih (drugih) slik
3. Funkcije nad narisanimi objekti (ukazi oz. ikone v orodni vrstici)
 - a. Premikanje
 - b. Brisanje
 - c. Kopiranje
 - d. Vrtenje
 - e. Zrcaljenje
 - f. Dodajanje in odvzemanje točk
 - g. Poravnava (Align – samo znotraj združenega objekta)
 - h. Povečave (Scale – z miško)
 - i. Združevanje objektov
4. Lastnosti risanih objektov
 - a. Določanje lastnosti pred risanjem objekta
 - i. Izberemo funkcijo risanja
 - ii. Določimo lastnosti v orodni vrstici na dnu
 - iii. Rišemo
 - b. Določanje lastnosti že narisane objekta
 - i. Uporabimo funkcijo EDIT
 - ii. Izberemo objekt
 - iii. Spreminjamo lastnosti (Apply, Reset, Ok)
 - c. Lastnosti so različne pri različnih objektih (Debelina črte, Barva črte, Barva ploskve, Način stikanja črt, Oblika puščic, Velikost črt, Barva črk, Poravnava in naklon, ...)

Podobne operacije omogočajo vsi programi za delo z vektorsko grafiko. Vektorska grafika se uporablja predvsem za tiskane dokumente (brošure, logotipi, celostne grafične podobe, ...), ker se kvaliteta z večanjem dokumenta ne spreminja.

Primeri sodobnih profesionalnih orodij sta Adobe Illustrator in Inkscape:



SLIKE – Delo s slikami – rastrske slike

Bitne (rastrske) slike

Bitne slike so predstavljene z matriko števil, ki opisujejo posamezne piksele slike. Vsaka točka je predstavljena z nekim številom ali množico števil. Velikost slike je fiksna.

Programska orodja, ki jih imamo na voljo so:

- Xview – program za delo z bitnimi slikami (image processing)
- GIMP
- OpenOffice Draw
- Adobe Photoshop
- PaintShop PRO
- Windows Paint

Kratkek pregled možnosti, ki ji ponuja programsko orodje XView

1. OSNOVNO OKNO

- a. Srednji gumb (informacije o pixlu-točki: x,y; RGB, #RGB, HSV; dx,dy)
- b. Levi gumb (izbor območja za obdelavo)
- c. Povečave
 - i. CTRL+levi gumb miške (ZoomIn)
 - ii. CTRL+srednji gumb miške (Pan)
 - iii. CTRL+desni gumb miške (ZoomOut)
- d. Editiranje točk
 - i. Izbira barve (srednji gumb miške)
 - ii. Editiranje točke oz. risanje (SHIFT + srednji gumb miške)
 - iii. Megljenje slike (SHIFT + desni gumb miške)

2. KONTROLNO OKNO

- a. Odpiranje in pregledosnovnih funkcij (Load, Save, Delete, Print)
- b. Določanje velikosti slike (ImageResize)
- c. Prikazovanje slike (Display)
- d. Predstavitev slike v oknu ali na namizju (Root)
- e. Obseg barv slike (meni 24/8 BIT) (bitne slike ali 24bitni RGB)
- f. Specialne funkcije (Algoritmi)
- g. Oblikovaje slike (meni Windows)
- h. Druge funkcije
 - i. Zrcaljenje, obračanje
 - ii. Izrezi
 - iii. Kopiranje, lepljenje
 - iv. Dodajanje roba
 - v. Dodajanje texta

3. KONTROLNO OKNO – OBDELAVA BARV

- a. Colormap Editing (spreminjanje trenutno izbrane barve)
 - i. ColUndo (preklic zadnjega ukaza)
 - ii. Revert (preklic vsega)
 - iii. RGB/HSV (preklo na določanje HSV iz RGB in obratno)
 - iv. Grey (sivi odtenki)

- v. RevVid (obrne paleta)
- vi. Random (gen. poljubne palete)
- b. Colormap Editing – ukazi (kontrola nad celo sliko)
- c. HSV Modification – spreminjamo celo sliko na osnovi spreminjanja HSV barv
 - i. Na vrhu: nastavitve barve
 - ii. Srednji del: nastavitve svetlosti in bledosti barve
 - iii. Spodnji del: nastavitve intenzivnosti barve
- d. RGB Modification – spreminjamo celo sliko na osnovi spreminjanja RGB barv

MATEMATIČNI PROBLEMI – Octave

1. Razumevanje delovanja octave:
 - a. `>>` ... input
 - b. `ans` ... rezultat
 - c. Posebne oznake:
 - i. Komentar do konca vrstice: `%`
 - ii. Spremno besedilo: `"`
 - d. Občutljivost na velike in male črke
2. Oklepaji
 - a. Funkcijska odvisnost oz. parametri funkcije: `()` (npr. `f(x)`)
 - b. Vektorji. Matrike, polinomi: `[]`
 - c. Indeksiranje: `()` (npr. `v(i)` – i-ti element vektorja `p`)
3. Pomoč: `help` ukaz
4. Uporaba enakosti.
 - a. enostavno prirejanje: `>> a=4`
 - b. logično primerjanje: `>> a == 1`
5. Osnovne operacije s skalarji
 - a. Seštevanje: `n+m`
 - b. Odštevanje: `n-m`
 - c. Množenje: `n*m`
 - d. Deljenje: `n/m`
 - e. Potenciranje: `n^m`
6. Kompleksna števila
 - a. `a=2+3i`
 - b. Oznaki `i` in `j` sta rezervirani za kopleksni enoti (če ju uporabljamo drugače, potem ju ne moremo več uporabljati kot kompleksni enoti)
 - c. Funkcije nad kompleksnimi števili:
 - i. `imag(x)` ... imaginarni del števila `x`
 - ii. `real(x)` ... realni del števila `x`
 - iii. `conj(x)` ... konjugirano število k številu `x`
7. Vektorji in matrike
 - a. Vektor:
 - i. Zapis: `v1=[1 2 3]` ali `v1=[1,2,3]`
 - ii. Generiranje vektorjev:
 1. `X=1:5` `X=[1 2 3 4 5]`
 2. `X=1:2:10` `X=[1 3 5 7 9]` vektor z vrednostmi od 1 do 10 s korakom 2
 - b. Matrika: `A=[1 2 3;3 4 5; 5 6 7]`
 - i. Elementi matrike: `A(2,3)` ... element v 2.vrstici in 3.stolpcu
 - ii. Izrez dela matrike
 1. podmatrika `A(1:2,2:3)` ... 1. in 2. vrstica ter 2. in 3. stolpec
 2. stolpec matrike: `A(:,2)` ... 2. stolpec
 3. vrstica matrike: `A(2,:)` ... 2. vrstica

- iii. Funkcije nad matrikami
 - 1. size(A) ... dimenzija matrike
 - 2. A' ... transponirana matrika A
 - 3. inv(A) ... inverz kvadratne matrike A
 - 4. det(A) ... determinanta kvadratne matrike A
 - 5. eig(A) ... lastne vrednosti matrike A

- 8. Reševanje sistema enačb
 - a. Podamo sistem z matrikami:
 - i. B matrika koeficientov spremenljivk,
 - ii. A matrika vrednosti enačb
 - iii. B\A vektor rešitev sistema enačb

- 9. Polinomi
 - a. Polinom podamo kot vektor koeficientov od najvišje prosti najnižji potenci
P=[1 3 0 7] ... predstavlja polinom $p(x) = x^3 + 3x^2 + 7$.
 - b. Funkcije nad polinomi:
 - i. polyder(P) ... odvod polinoma
 - ii. roots(P) ... ničle polinoma
 - iii. poly(X) ... vrne polinom, čigar ničle so vrednosti vektorja X
 - iv. length(P) vrne dolžino (stopnjo) polinoma

- 10. Grafika
 - a. Funkcija PLOT
 - i. plot([1 1 1 1]) ... nariše točke s koordinatami (1,1), (2,1), (3,1), (4,1)
 - ii. plot(X,Y) ... nariše točke, kjer je prva koordinata vrednost iz vektorja X, druga pa istoležeča vrednost v vektorju Y
 - iii. plot ima seveda svoje opcije → help Plot

- 11. Programiranje:
 - a. Svoje funkcije shranimo v datoteki s končnico .m (test.m)

 - b. Deklaracija:
 - i. Funkcija brez rezultata (npr. samo prikaže graf) (npr. fun1)
Function test(parametri)
.... %vsebina
return; %konec funkcije

 - ii. Funkcija z enim rezultatom (npr. fun2)
Function t=test(parametri)
.... %vsebina, ki vključuje prireditev t=neka vrednost
return; %konec funkcije

 - iii. Funkcija z več rezultati (npr. fun3)
Function [t,k]=test(parametri)
.... %vsebina, ki vključuje prireditev t=neka vrednost, k=neka vrednost
return; %konec funkcije

- c. Standardni stavki
 - i. IF pogoj stavki; ELSE(ELSEIF)stavki; END
 - ii. FOR števec stavki; END
 - iii. WHILE pogoj stavki; END

- d. Logični operatorji:
 - i. AND ... &
 - ii. OR ... |
 - iii. NOT ... ~

12. Primer: risanje polinoma (pol.m)

13. Primer: iskanje ničel polinoma z bisekcijo (bisek.m)

14. Primer: risanje kroga

15. Koristne povezave

- <http://www.octave.org/>
- <http://www.ro.feri.uni-mb.si/predmeti/navodila/matlab/matlabnav.html>
- <http://www.math.utah.edu/lab/ms/matlab/matlab.html>

