

I. Naštej komponente glavnega okna aplikacije kot jih predlagajo navodila za načrtovanje grafičnih uporabniških vmesnikov in razloži njihove funkcije.

Komponente oken:

- okvir (meja okna)
- naslov (uporabnik lahko takoj prepozna kateri aplikaciji pripada in kakšen namen ima okno)
- sistemski menu (osnovne sistemske funkcije za delo z oknom – zapri, minimiziraj...)
- gumbi za velikost okna (hiter dostop do manipulacij za velikost delovnega okna)
- vrstični menu (hierarchicalno nanizane funkcije aplikacije)
- delovno področje
- vrstica za izpis statusa (pogled nad delovanjem aplikacije)
- vrstica za obvestila (pogled nad delovanjem aplikacije)
- vrstica za ukaze (nekater ukaze lahko bistveno hitreje izvršimo, če jih preprosto odtipkamo)
- drsniki (enostavno premikanje po oknu)
- pregrade (ločevanje vsebinsko različnih pojmov)
- polje za prikaz permanentnih ukazov in opcij (hitrejši dostop do ukazov)

2. Razloži koncept direktne manipulacije pri objektno usmerjenih uporabniških vmesnikih.

Za direktno manipulacijo je značilno:

- vizualizacija vseh objektov (omizje, ikone – datoteke, direktoriji, programi) sistem je podaljšek realnega sveta
- hitre, reverzibilne akcije
- zvezna vidljivost objektov in akcij
- prepoznavanje (razbremenjevanje hitrega spomina)

Slabosti:

- težko načrtovanje, zahtevno programiranje, zmogljiva strojna oprema
- ikone niso vedno razumljive, vse naloge se ne dajo opisati s konkretnimi objekti in direktno akcijo
- delovna domena je sedanost

Prednosti:

- hitro učenje
- uporabnik neposredno vidi, če akcije vodijo k cilju
- uporabnik pridobi samozaupanje in kontrolo

3. Sediš za računalnikom z imenom miki. Napiši in razloži vse potrebne sistemske ukaze za delo z X grafično aplikacijo, ki je na oddaljenem računalniku z imenom pluton. Na katerem od obeh računalnikov je strežnik in na katerem odjemalec ?

Na lokalnem računalniku omogočim exportanje prikazovalnika: miki> xhost +pluton, potem naredim telnet ali remote login na oddaljeni računalnik: telnet pluton ali rlogin pluton in tam izvozim prikazovalnik: export DISPLAY=miki:0.

Glede na to, da je po terminologiji odjemalec aplikacija, ki teče na oddaljenem stroju na mreži je pluton odjemalec; strežnik pa je programski proces, ki komunicira z odjemalci X in jim daje potrebne vire za tekoče delo in potemtakem je to miki.

4. V okolju Motif napiši grafični vmesnik za risanje trikotnikov. Vmesnik naj temelji na podobi za risanje DrawingArea dimenzije 300 X 400 pikslov. Trije zaporedni kliki (katerakoli tipka na miški) v polju za risanje predstavljajo oglišča trikotnika. Vsak naslednji trikotnik naj se nariše po vsakem tretjem kliku.

//na začetek programa dodamo datoteko z zaglavji:

```
#include <Xm/DrawingA.h>
```

Widget drawing_a; //deklariramo globalno spemeljivko

int korak=1; // stevec korakov za risanje □ (globalno)

...

//v main proceduro vpisemo:

```
drawing_a=XtVaCreateManagedWidget(>drawing_a«, xmDrawingAreaWidgetClass, main_w, XmNwidth, 300, HmNheight, 400,NULL); //predpostavimo da je main_w
```

predhodnik

```
XtAddCallback(drawing_a, XmNexposeCallback, draw, NULL); //dodamo odzivno proceduro
```

...

```
XtAppMainLoop(app); //gremo v zanko dogodkov...
```

...

```
void draw(widget, client_data, call_data)
```

```
Widget widget;
```

```
XtPointer client_data;
```

```
XtPointer call_data;
```

```
{
```

```
    static Position x1,x2,y1,y2;
```

```

XmDrawingAreaCallbackStruct *cbs(XmDrawingAreaCallbackStruct *) call_data;
XEvent *event = cbs->event;
Display *dpy = event->xany.display;

if (korak==1){ //prvi korak, prve koordinate
    x1=event->xbutton.x;
    y1=event->xbutton.y;
}
else if (korak==2){ //drugi korak, druge koordinate
    x2=event->xbutton.x;
    y2=event->xbutton.y;
}
else if (korak==3){ // tretji korak
    korak=1; //resetiramo stevec korakov
    XDrawLine(dpy, cbs->window, gc, x1, y1, x2, y2);
    XDrawLine(dpy, cbs->window, gc, x2, y2, event->xbutton.x, event->xbutton.y);
    XDrawLine(dpy, cbs->window, gc, event->xbutton.x, event->xbutton.y,
        x1, y1); // narisemo vse tri crte □
}
}
}
predpostavimo da je gc ze kreiran graficni kontekst

```

5. V okolju Motif napiši grafični vmesnik, ki temelji na upravniku tipa RowColmn in vsebuje štiri stikala tipa RadioButton ter en gumb tipa PushButton. Stikala in gumb morajo biti razvrščeni v eno vrstico, najprej stikala, nato gumb. Stikala naj ponujajo ekskluzivno izbiro ene od štirih možnosti. Gumb naj v ukazno okno izpiše stanje stikal v danem trenutku. Stikala naj nosijo oznake Ena, Dva, Tri in Stiri, gumb pa oznako Preveri.

```

#include <Xm/ToggleBG.h>
#include <Xm/RowColumn.h>
#include <Xm/PushB.h>

Widget Ena, Dva, Tri, Stiri, Preveri;
int radiostate=0;

void main
Widget rc;
...

rc=XtVaCreateManagedWidget(»rc«, xmRowColumnWidgetClass, toplevel, XmNorientation, XmHORIZONTAL, XmNradioBehavior, True, NULL); //rc dolocimo horizontalno
poravnavo in dolocimo, da bodo gumbi na njem tipa radiobutton

Ena=XtVaCreateManagedWidget (»ena«, xmToggleButtonGadgetClass, rc, NULL);
XtAddCallback(Ena, XmNvalueChangedCallback, toggled, 1);

Dva=XtVaCreateManagedWidget (»dva«, xmToggleButtonGadgetClass, rc, NULL);
XtAddCallback(Dva, XmNvalueChangedCallback, toggled, 2);

Tri=XtVaCreateManagedWidget (»tri«, xmToggleButtonGadgetClass, rc, NULL);
XtAddCallback(Tri, XmNvalueChangedCallback, toggled, 3);

Stiri=XtVaCreateManagedWidget (»stiri«, xmToggleButtonGadgetClass, rc, NULL);
XtAddCallback(Stiri, XmNvalueChangedCallback, toggled, 4);

Preveri=XtVaCreateManagedWidget(»preveri«, xmPushButtonWidgetClass, rc, NULL);
XtAddCallback(Preveri, XmNactivateCallback, button_pushed);
...
XtAppMainLoop(app); //gremo v zanko dogodkov...
...

void toggled(widget, client_data, call_data)
Widget widget;
XtPointer client_data;
XtPointer call_data;
{
int which = (int) client_data;
radiostate=which;
}

void button_pushed(widget, client_data, call_data)
Widget widget;
XtPointer client_data;

```

```
XtPointer call_data;
{
fprintf(stderr, »Prizgan je %d. gumb/n«, radiostate);
}
```

I. Naštej faze načrtovanja grafičnih uporabniških vmesnikov.

- razumevanje uporabnika
- strojna oprema in grafične sposobnosti sistema
- razumevanje zeljene aplikacije
- načrtovanje oken
- načrtovanje menujev
- interakcija uporabnika s sistemom
- interakcija uporabnika s sistemom preko oken
- aranžiranje gr. gradnikov za komunikacijo
- izbor barv
- izbor in načrtovanje ikon
- povratna informacija: obvestila, dialogi, navodila, pomoč
- testiranje

2. Povej kaj je to metafora, nato pa razloži vlogo in način uporabe metafor pri načrtovanju uporabniških vmesnikov.

Ikone so metafore, ki opisujejo in predstavljajo objekte in akcije. Ikone so slike.

3. V okolju Motif načrtuj vrstični menu tipa MenuBar. Vrstični menu naj vsebuje opcije File, Edit in Help ter naj je vezan na glavno okno tipa MainWindow. Menu Edit naj vsebuje opciji Forward in Backward. Aplikaciji dodaj datoteko resursov, ki opcijama menuja Edit doda pospeševalnike (Ctrl-F in Ctrl-B).

```
#include <Xm/CascadeB.h>
#include <Xm/PushB.h>
...
Widget MenuBar;
...

void main
...
Widget MainWindow;
XmString file, edit, help, forward, backward;
...
XtAppContext app;
toplevel=XtVaAppInitialise(&app, »my_app«, NULL, 0,&argc, argv, NULL, NULL);
...

file=XmStringCreateLcalized(»File«);
edit=XmStringCreateLcalized(»Edit«);
help=XmStringCreateLcalized(»Help«);
MenuBar=XmVaCreateSimpleMenuBar(MainWindow, »menubar«,
    XmVACASCADEBUTTON, file, 'F',
    XmVACASCADEBUTTON, edit, 'E',
    XmVACASCADEBUTTON, help, 'H',NULL);
XmStringFree(file);
XmStringFree(edit);
XmStringFree(help);

forward=XmStringCreateLcalized(»Forward«);
backward=XmStringCreateLcalized(»Backward«);
XmVaCreateSimplePulldownMenu(MenuBar, »edit_menu«, 1, edit_cb,
    XmVaPUSHBUTTON, forward, 'F', NULL, NULL,
    XmVaPUSHBUTTON, backward, 'B', NULL, NULL,NULL);

XmStringFree(forward);
XmStringFree(backward);

XtManageChild(MenuBar); //upravljamo menu
...
XtRealizeWidget(toplevel);
XtAppMainLoop(app);
```

datoteka resursov (my_app):

```
*edit_menu.button_0.accelerator: Ctrl<Key>F
```

```
*edit_menu.button_0.acceleratorText: Ctrl+F
```

```
*edit_menu.button_1.accelerator: Ctrl<Key>B
```

```
*edit_menu.button_1.acceleratorText: Ctrl+B
```

4. V okolju Motif napiši grafični vmesnik za risanje kvadratov in križev dimenzije 10 X 10 pikslov. Vmesnik naj temelji na podobi za risanje tipa DrawingArea dimenzije 300 X 400 pikslov. Vmesnik naj vsebuje dvižni menu z dvema opcijama: Rectangle in Cross. Opcija Rectangle naj nariše kvadrat na mestu kjer je bil zahtevan dvižni menu, opcija Cross pa križ.

```
#include <Xm/MainW.h>
#include <Xm/DrawingA.h>

int Xpos, Ypos;

void main
...
drawing_a=XtVaCreateManagedWidget(»drawing_a«, xmDrawingAreaWidgetClass, main_w,
    XmNwidth, 300, XmNheight, 400, NULL);
popup_menu=xmVaCreateSimplePopupMenu(drawing_a, »popup«, popup_cb,
    XmVaPUSHBUTTON, XmStringCreateLocalized(»Rectangle«, 'R', NULL, NULL,
    XmVaPUSHBUTTON, XmStringCreateLocalized(»Cross«, 'C', NULL, NULL, NULL);
XtAddCallback(drawing_a, XmNinputCallback, input, popup_menu);
...
XtAppMainLoop(app);

void input (menu_item, client_data, call_data)
Widget menu_item;
XtPointer client_data;
XtPointer call_data;
{
Widget popup=(Widget) client_data;

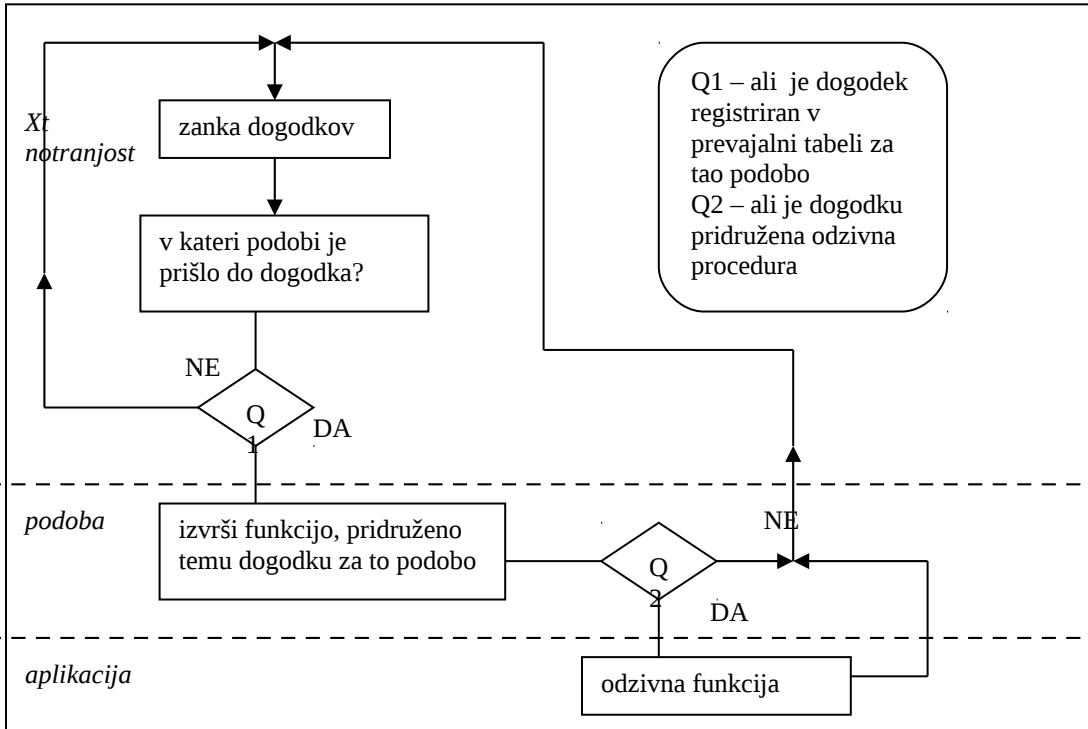
XmDrawingAreaCallbackStruct *cbs=(XmDrawingAreaCallbackStruct *) call_data;
if (cbs->event->xany.type !=ButtonPress || cbs->event->xbutton.button !=3) return;
Xpos=cbs->event->xbutton.x;
Ypos=cbs->event->xbutton.y;
XmMenuPosition(popup, (XButtonPressedEvent *) (cbs->event));
XtManageChild(popup);
}

void popup_cb(menu_item, client_data, call_data)
Widget menu_item;
XtPointer client_data;
XtPointer call_data;
{
int item_no=(int) client_data;
if (item_no==0){ //risemo kvadrat
    XDrawLine(XtDisplay(drawing_a), XtWindow(drawing_a),gc,
        Xpos-5, Ypos-5, Xpos+5, Ypos-5);
    XDrawLine(XtDisplay(drawing_a), XtWindow(drawing_a),gc,
        Xpos-5, Ypos-5, Xpos-5, Ypos+5);
    XDrawLine(XtDisplay(drawing_a), XtWindow(drawing_a),gc,
        Xpos-5, Ypos+5, Xpos+5, Ypos+5);
    XDrawLine(XtDisplay(drawing_a), XtWindow(drawing_a),gc,
        Xpos+5, Ypos+5, Xpos+5, Ypos-5);
}
else if (item_no==1){ //risemo kriz
    XDrawLine(XtDisplay(drawing_a), XtWindow(drawing_a),gc,
        Xpos-5, Ypos, Xpos+5, Ypos);
    XDrawLine(XtDisplay(drawing_a), XtWindow(drawing_a),gc,
        Xpos, Ypos-5, Xpos, Ypos+5);
}
}
```

I. Naštej in razloži bistvene razlike med grafičnimi uporabniškimi vmesniki in objektno usmerjenimi uporabniškimi vmesniki.

	GUV	OUIV
aplikacija vmesnik	aplikacija=ikona, primarno, sekundarna okna	vmesnik=množica sodelujočih objektov in pogledov na njih
ikona	ikone predstavljajo aplikacije	ikone predstavljajo objekte, ki jih lahko direktno manipuliramo
delo	poženemo aplikacijo pred delom z njo	odpremo pogled v objekt pred delom z njim
pozornost	pozornost nad osnovno nalogo kot je ta določena z aplikacijo	pozornost nad vhodom in izhodom za objekte ter nalogami
učenje	koncentrirano na aplikacijo in njene funkcije	koncentrirano na skupne pristope in izgled objektov
vsebina	prikazana s tekstovnimi listami ali dialogi	prikazana z vsebovalniki (folder, notebook)
ena	ena aplikacija je eno opravilo	en objekt lahko uporabimo za različna opravila

2. Dana je aplikacija napisana v okolju Motif. Dogodke obravnavajo Xt notranje funkcije, podobe in odzivne funkcije pridružene podobam. Nariši in razloži diagram poteka dela z dogodki.



Aplikacija se nahaja v zanki dogodkov. Ko se kakšen dogodek zgodi, je treba najprej ugotoviti v kateri podobi je prišlo do dogodka. Če ta dogodek ni registriran v prevajalni tabeli dogodkov za to podobo (recimo klik na label) potem se aplikacija vrne v zanko dogodkov. V nasprotnem primeru se izvrši funkcija, pridružena temu dogodku (klik na gumb – sprememba senčenja objekta), preveriti je treba le še ali je temu dogodku pridružena kakšna odzivna procedura. Če je, se le-ta izvrši (npr. odpre se novo okno, pojavi dialog...) in se aplikacija vrne v zanko dogodkov, sicer se aplikacija vrne v zanko dogodkov.

3. Dan je grafični vmesnik napisan v Motif okolju. Njegovo ime je "Buttons". Sestavljajo ga upravnik tipa MainWindow z imenom "mwindow" kateremu sledita dva upravnik tipa BulletinBoard z imenoma "bbord-left" in "bbord_right". Prvi vsebuje dva gumba tipa PushButton z imenoma "button-1" in "button-2", drugi pa prav tako dva gumba tipa PushButton z imenoma "button_3" in "button_4". Napiši vsebino datoteke resursov, ki realizira naslednje lastnosti podob:

- Vmesnik je širine 300 in višine 170 pikslav;
- Vsi gumbi se nahajajo na odmiku 20 pikslav v smeri x glede na svojega predhodnika;
- Barva ospredja gumbov, ki sledijo podobi "bbord-left" je zelena;
- Tretji gumb, "button_3", je širine 70 ter višine 25 pikslav;
- Font gradnikov, ki sledijo podobi "bbord_right" je tipa "--courier-medium--12-".

```
*mwindow.width: 300          *mwindow.height: 170
*XmPushButton.leftOffset: 20
*bbord-left.XmPushButton.foreground: green
*button_3.width: 70          *button_3.height: 25
*bbord-right*fontList: --curier-medium-12
```

4. V okolju Motif napiši grafični vmesnik za risanje kvadratov dimenzije 10 X 10

pikslov. Vmesnik naj temelji na podobi za risanje tipa DrawingArea dimenzije

300 X 400 pikslov. Na začetku naj vmesnik nastavi barvo ozadja v podobi za risanje na belo. Vmesnik naj vsebuje dvižni menu z dvema opcijama:

Rectangle in Redraw. Opcija Rectangle naj nariše kvadrat na mestu kjer je bil izbran dvižni menu, opcija Redraw pa naj zbriše vsebino polja za risanje in nastavi njegovo ozadje na belo.

```
#include <Xm/MainW.h>
#include <Xm/DrawingA.h>

int Xpos, Ypos;

void main
...
drawing_a=XtVaCreateManagedWidget(»drawing_a«, xmDrawingAreaWidgetClass, main_w,
    XmNwidth, 300, XmNheight, 400, NULL);
XSetBackground(XtDisplay(drawing_a), gc, WhitePixelOfScreen(XtScreen(Drawing_a)));
popup_menu=xmVaCreateSimplePopupMenu(drawing_a, »popup«, popup_cb,
    XmVaPUSHBUTTON, XmStringCreateLocalized(»Rectangle«, 'R', NULL, NULL,
    XmVaPUSHBUTTON, XmStringCreateLocalized(»Redraw«, 'D', NULL, NULL, NULL));
XtAddCallback(drawing_a, XmNinputCallback, input, popup_menu);
...
XtAppMainLoop(app);

void input (menu_item, client_data, call_data)
Widget menu_item;
XtPointer client_data;
XtPointer call_data;
{
Widget popup=(Widget) client_data;

XmDrawingAreaCallbackStruct *cbs=(XmDrawingAreaCallbackStruct *) call_data;
if (cbs->event->xany.type !=ButtonPress || cbs->event->xbutton.button !=3) return;
Xpos=cbs->event->xbutton.x;
Ypos=cbs->event->xbutton.y;
XmMenuPosition(popup, (XButtonPressedEvent *) (cbs->event));
XtManageChild(popup);
}

void popup_cb(menu_item, client_data, call_data)
Widget menu_item;
XtPointer client_data;
XtPointer call_data;
{
int item_no=(int) client_data;
if (item_no==0){ //risemo kvadrat
    XDrawLine(XtDisplay(drawing_a), XtWindow(drawing_a),gc,
        Xpos-5, Ypos-5, Xpos+5, Ypos-5);
    XDrawLine(XtDisplay(drawing_a), XtWindow(drawing_a),gc,
        Xpos-5, Ypos-5, Xpos-5, Ypos+5);
    XDrawLine(XtDisplay(drawing_a), XtWindow(drawing_a),gc,
        Xpos-5, Ypos+5, Xpos+5, Ypos+5);
    XDrawLine(XtDisplay(drawing_a), XtWindow(drawing_a),gc,
        Xpos+5, Ypos+5, Xpos+5, Ypos-5);
}
else if (item_no==1){ //belimo
    XSetForeground(XtDisplay(drawing_a), gc,
        WhitePixelOfScreen(XtScreen(Drawing_a))); //barvali bomo z belo barvo
    XFillRectangle(XtDisplay(drawing_a), XtWindow(drawing_a), gc, 0, 0, 300, 400); //barvamo
    XSetBackground(XtDisplay(drawing_a), gc,
        WhitePixelOfScreen(XtScreen(Drawing_a))); // nastavimo barvo ozadja
}
}
```

I. Razloži delovanje senzornega, kratkotrajnega in dolgotrajnega spomina pri človeku ter prenašanje informacij med njimi.

- senzorni spomin (SM) sprejema ogromno količino informacij, ki se jih niti ne zavedamo
- kratkotrajni spomin (STM) hrani okoli 7 dejstev približno 30 sekund
- dolgotrajni spomin (LTM) – v njem so informacije kodirane. Hrani lahko do 100.000 besed, vendar lahko dostopamo samo do 2000-3000

Prenšanje informacij:

informacija pride preko čutil in procesorjev za zaznavanje v SM, od tu pa v STM. Tu se lahko informacija na podlagi podatkov iz LTM preoblikuje s posredovanjem procesorjev za razumevanje. Ko dostopamo do podatkov v LTM se morajo le-ti dekodirati v procesorjih za razumevanje in prenesti v STM. Tu lahko dostopamo do njih.

2. Naštej prednosti grafičnega prikaza informacij pri uporabniških vmesnikih.

- hitrejše učenje
- boljša preglednost

3. Dan je grafični vmesnik napisan v Motif okolju. Njegovo ime je "Gumbi". Sestavljajo ga upravnik tipa MainWindow z imenom "gokno" kateremu sledita dva upravnik tipa BulletinBoard z imenoma "ogdeska-levo" in "ogdeska-desno". Prvi vsebuje dva gumba tipa PushButton z imenoma "gumb-1" in "gumb-2", drugi pa prav tako dva gumba tipa PushButton z imenoma "gumb-3" in "gumb-4". Napiši vsebino datoteke resursov, ki realizira naslednje lastnosti podob:

- a. Vmesnik je širine 300 in višine 170 pikslov;
- b. Vsi gumbi se nahajajo na odmiku 20 pikslov v smeri x glede na svojega. predhodnika;
- c. Barva ospredja gumbov, ki sledijo podobi "ogdeska-levo" je rdeča;
- d. Četrti gumb, "gumb-4", je širine 50 ter višine 25 pikslov;
- e. Font gradnikov, ki sledijo podobi "ogdeska-desno" je tipa "--courier-medium--12-".

```
*gokno.width: 300
```

```
*gokno.height: 170
```

```
*XmPushButton.leftOffset: 20
```

```
*ogdeska-levo.XmPushButton.foreground: red
```

```
*gumb-4.width: 70
```

```
*gumb-4.height: 25
```

```
*ogdeska-desno*fontList: --curier-medium-12
```

4. V okolju Motif načrtuj grafični vmesnik, ki ga sestavljata glavno okno tipa MainWindow in vrstični menu tipa MenuBar. Vrstični menu naj vsebuje opcije File, Options in Help ter naj je vezan na glavno okno. Menu Options naj vsebuje opciji Option_1 in Option_2. Izbira ene ali druge od teh dveh opcij naj povzroči ustrezen tekstovni izpis v ukazno okno. Vmesnik napišite v celoti. Izjeme so realizacija opcij menujev File in Help ter odzivni funkciji teh dveh menujev.

```
#include <Xm/CascadeB.h>
```

```
#include <Xm/PushB.h>
```

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

```
...
```

```
Widget MenuBar;
```

```
...
```

```
void main()
```

```
...
```

```
Widget MainWindow;
```

```
XmString file, options, help, option1, option2;
```

```
...
```

```
file=XmStringCreateLcalized(»File«);
```

```
options=XmStringCreateLcalized(»Options«);
```

```
help=XmStringCreateLcalized(»Help«);
```

```
MenuBar=XmVaCreateSimpleMenuBar(MainWindow, »menubar«,
```

```
    XmVACASCADEBUTTON, file, 'F',
```

```
    XmVACASCADEBUTTON, options, 'O',
```

```
    XmVACASCADEBUTTON, help, 'H',NULL);
```

```
XmStringFree(file);
```

```
XmStringFree(options);
```

```
XmStringFree(help);
```

```
option1=XmStringCreateLcalized(»Option_1«);
```

```
option2=XmStringCreateLcalized(»Option_2«);
```

```
XmVaCreateSimplePulldownMenu(MenuBar, »option_menu«, 1, options_cb,
```

```
        XmVaPUSHBUTTON, option1, '1', NULL, NULL,  
        XmVaPUSHBUTTON, option2, '2', NULL, NULL,NULL);  
XmStringFree(option1);  
XmStringFree(option2);
```

```
XtManageChild(MenuBar); //upravljamo menu
```

```
...
```

```
XtRealizeWidget(toplevel);  
XtAppMainLoop(app);
```

```
void options_cb(widget, client_data, call_data)
```

```
Widget widget;
```

```
XtPointer client_data;
```

```
XtPointer call_data;
```

```
{
```

```
int item_no=(int)client_data;
```

```
if (item_no==0) fprintf(stderr, »Option_1«);
```

```
else if (item_no==1) fprintf(stderr, »Option_2«);
```

```
},
```