

Redovisning av projektet "Metod för
driftkostnads- och underhållsplanering -
teoribildning med projektnummer 920527-1

Forsberg Claes och Gustafsson Stig-Inge

Innehåll

1 Förord	5
2 Från Opera till Kaizen	7
2.1 Förändringar i Opera	7
2.2 Förändringar i SORAD	8
2.3 Förändringar i BIVAL	8
2.4 Nyutvecklat program	8
A EXCOPER.C	11
B Nya SOLWIN	13
C KAIZEN.C	23
D RC-fl	43
E Does postpond retrofitting save money?	47

Kapitel 1

Förord

Rubricerat projekt är endast en av två separata delar. Den andra delen kommer att redovisas för sig via Energimarknad Claes Forsberg AB i Skutskär. Då det huvudsakliga rapportskrivandet sker i Skutskärsdelen kommer redovisningen här endast att behandla de program och teorier som legat till grund för arbetet i Linköping och dessutom är framställningen mycket kortfattad och avsedd mera som en orientering om det arbete som utförts än som en pedagogisk framställning där vunna erfarenheter publiceras. Avsikten är att man skall läsa rapporten från Skutskär först och sedan använda denna rapport mera som en bilaga eller ett appendix. Nedan beskrivs de olika programmen och hur de handhas.

Kapitel 2

Från Opera till Kaizen

2.1 Förändringar i Opera

Hela programsystemet, vilket tills vidare har arbetsnamnet "System Kaizen", vilar på den s.k. OPERA-modellen. Denna används för att finna den precisa avvägningen mellan installations-, bygg- och ventilationsåtgärder, så att en byggnads framtida livscykelkostnad skall kunna minimeras. OPERA har beskrivits i detalj i andra publikationer, bl. a. i Referens [1] vilken publicerats av rådet. Någon närmare beskrivning av OPERA sker därför inte här. För att OPERA skulle kunna samverka med med den databas som utvecklats i Skutskär, inom System Kaizen, och genom utnyttjandet av programmen Microsoft Access och Excel var vissa förändringar nödvändiga. Vår ambition har varit att hela programpaketet skall kunna användas i den s.k. Windowsmiljön på en vanlig IBM-kompatibel PC. Detta har varit ett förhållandevis stort arbete. OPERA är skrivet i FORTRAN vilket språk är bra för tekniska beräkningar men mindre bra för programmering av fönster och dialogboxar. Vi har därför valt att kompilera om programmet som en s.k. QuickWin applikation vilket gör att det kan köras som en vanlig Windowsapplikation i ett fönster. Det program som exekveras heter OPEWINC.EXE. Tyvärr har man därvid endast begränsad rådighet över val i menyer och liknande. Vidare var det opraktiskt att behöva svara på "frågor" från OPERA när man redan bestämt alla indata i det nyutvecklade programpaketet. OPERA läser därför alltid, i denna version, en fil som heter "OPIN.DAT". Denna fil måste dessutom ligga i ett bibliotek som heter C:\EXCEL\OP. Man måste därför se till att "OPIN.DAT" finns där innan man försöker köra programmet. Normalt sker detta dock helt automatiskt.

Från System Kaizen startas OPERA med hjälp av ett s.k. macro. Detta macro exekveras automatiskt i programmet Excel som via samma macro redan skapat en textfil som heter "INDATA.TXT". Denna fil kan inte läsas direkt av OPERA utan måste omformas av programmet EXCOPER.EXE. Detta lilla program är skrivet i C och koden återfinnes i bilaga A. EXCOPER.EXE ser till att kommatecken m.m. sätts in på lämpliga platser i textfilen samt lägger till anföringstecken " " runt text så att OPERA kan läsa filen. Det var inte möjligt, eller lämpligt, att utföra detta direkt i Excel eller Access, varför EXCOPER.EXE måste användas. Operatören behöver dock inte själv göra detta utan allt sker automatiskt via nämnda macro. Filen INDATA.TXT förstörs dess-

utom, också automatiskt, varför man normalt inte ser denna alls. Fortfarande via macrot, startas OPERA. Det är möjligt att starta programmet i ett fönster eller som en ikon i bakgrunden. Det viktiga är dock att man låter programmet köra färdigt innan nästa steg i macrot träder in. OPERA bildar nämligen ett antal filer varav det nyutvecklade systemet behöver två. Dessa filer heter "OPUT.TXT" samt "OPUT1.TXT". Dessa filer har speciellt anpassats så att de kan läsas in av Excel helt automatiskt. Bägge filerna skapas i katalogen C:\EXCEL\OP. I filen "OPUT.TXT" hamnar ett antal energibalanser som OPERA beräknar och framför allt den energibalans som visas för det system som befunnits vara optimalt. I filen "OPUT1.TXT" hamnar alla åtgärder som visats vara optimala. Programmet Excel kräver att alla indata till ett arbetsblad måste vara placerade på exakt rätt ställe för att en automatisk inmatning skall kunna ske. Man skall därför inte ändra i vare sig indata- eller utdatafiler om man inte vet exakt vad man gör. Meningen är istället att man skall ändra allting från System Kaizen's rutor och dialogboxar. Programkoden till OPERA har presenterats i Referens [2], sidorna 117 - 184. Då denna omfattar c:a 70 sidor och då förändringarna i OPERA varit förhållandevis små har inte koden publicerats igen. Den intresserade kan beställa en kopia kostnadsfritt från författaren.

2.2 Förändringar i SORAD

Till OPERA-modellen hörde ett separat program vilket beräknade solinstrålningen genom fönstren i byggnaden. Detta program beskrivs också närmare i [1], sidorna 77 - 94, varför ingen närmare presentation sker här. Programmet var sedan tidigare skrivet i C vilket innebär att det utan alltför stor arbetsinsats gick att omforma detta till en äkta s.k. Windowsapplikation. Programkoden återfinnes i bilaga B. De utdata som man tidigare erhöll har nu begränsats så att endast nödvändig information presenteras på skärmen. I dagsläget måste man själv föra in de solvärden som beräknats i motsvarande celler i det nyutvecklade programmet. Normalt behöver inte detta göras alls då lämpliga värden redan finns i den indatafil som levereras med System Kaizen.

2.3 Förändringar i BIVAL

Till OPERA hör dessutom ett program för s.k. diskret optimering av isoleråtgärder och bivalenta värmesystem, se [1], sidorna 95 - 106. Detta program är skrivet i FORTRAN varför vi även här tillgripit metoden med en sk QuickWin applikation. Inga förändringar har skett i koden varför vi hänvisar till [2], sidorna 202 - 206.

2.4 Nyutvecklat program

Till System Kaizen hör dessutom ett helt nyutvecklat program, kallat KAIZEN.EXE. Detta är en Windowsapplikation skriven i C++. Detta programspråk är objektorienterat vilket innebär att man kan ärva redan färdigskrivna klasser och programstrukturer och inkorporera dessa i sitt eget programsystem. Det klassbibliotek vi använt kommer från Borland och den version vi använt heter

3.1. Detta innebär att man förhållandevis enkelt kan skapa egna Windowsapplikationer utan att man för den skull behöver vara helt insatt i Windowsprogrammeringens mysterier. Trots detta har en avsevärd tid lagts ner på att utveckla denna applikation. Programmet har utvecklats så att det kan användas som ett skal varifrån de andra programmen i systemet kan startas. Dessutom finns en enklare texteditor inbyggd vilket innebär att man kan ändra filer ungefär på samma sätt som i "Windows anteckningar". Vidare har man möjlighet att erhålla s.k. on line hjälp för OPERA modellen. Hela referensen [1] har således omformats till en hjälpfil som kan nås via hjälpmenyn i programmet. Då programmet är nyutvecklat återfinnes hela koden i två bilagor, bilaga C och bilaga D. Förutom att programmet kan tjänstgöra som ett överordnat skal kan det användas för att studera hur livscykelkostnaden förändras för tidigare- och senareläggning av olika ROT-åtgärder. Anledningen till att detta arbete utförts är att husägaren mera sällan helt och hållet följer de råd denne får genom att använda OPERA. Exempelvis kan det vara optimalt att avstå från att vara ansluten till fjärrvärmenätet och istället bygga ett nytt eget värmesystem. Då det är vanligt att kommunen själv är fastighetsägare, via ett kommunalägt bolag, är detta en inte särdeles populär strategi även om fastighetsägaren skulle tjäna pengar på en sådan åtgärd. Pengarna skulle ju förloras på en annan plats i den kommunala ekonomin. OPERA antar vidare att man skall applicera alla åtgärder på samma gång vilket inte alltid passar in i de kommunala budgetarna. En fastighetsägare kanske vill vänta med en tilläggsisolering några år tills finansieringen blir lättare oavsett om detta är optimalt eller inte. System Kaizen var därför tvunget att kunna hantera sådana överväganden.

I bilagan E redovisas ett utkast till den artikel som kommer att utgöra en delredovisning till detta projekt. Notera att artikeln inte i dagsläget är publicerad varför vi helst ser att den ännu inte mångfaldigas. Vi kommer inom några dagar att sända artikeln till någon tidskrift inkluderad i Current Contents eller Science Citation Index. I artikeln undersöks om det skulle vara möjligt att erhålla en lägre livscykelkostnad genom att senarelägga ROT-åtgärder. Svaret är i många fall ja! De minskade kostnader man erhåller p.g.a. en minskad energianvändning kan många gånger ej balansera kostnaden för att tidigarelägga åtgärden ifråga. Gränsen för senareläggningen går dock, vad vi förstått, vid det årtal då byggnadsdelen ändå måste bytas ut, alltså av andra skäl en energiskäl. I övrigt hänvisas till bilagan.

Programmet KAIZEN.EXE läser samma fil som OPERA dvs OPIN.TXT. Som utdata erhålles kostnaderna redovisade för alla de år som projektet omfattar, ett värde som sätts vid indataproceduren. Vidare levererar också KAIZEN.EXE en fil som heter KAIZEN.TXT där energianvändning mm redovisas för de åtgärder operatören valt. Denna senare fil läses sedan från det övergripande programsystemet "System Kaizen" via de macron som tidigare omtalats och en rapport skrivs ut.

Bilaga A

EXCOPER.C

```
//Detta skall bli en konverterare för excel till Opera
#include "stdio.h"
#include "stdlib.h"
#include "ctype.h"

void main()

{
FILE *fi,*fo;
char varde;
int tecken=0,kont=0;
if ((fi=fopen("c:\\excel\\op\\indata.txt","r"))==NULL)
{
printf("Filen indata.txt kan inte hittas\n");
exit(0);
}
if ((fo=fopen("c:\\excel\\op\\opin.txt","w"))==NULL)
{
printf("Filen opin.txt kan inte hittas\n");
exit(0);
}
do
{
varde=getc(fi);
if(varde=='\n' && tecken==1)
{
if(kont==1)
{
putc(39,fo);
kont=0;
}
putc(varde,fo);
continue;
}
/*if(varde=='S')
{
printf("Varde\n");
} */
//Är det en bokstav?
if (isprint(varde) || varde=='å' || varde=='ä' || varde=='ö' || varde=='Å'
|| varde=='Ä' || varde=='Ö' || varde=='ä' )
{
if(varde!=' ')
{
if (isalpha(varde) || varde=='å' || varde=='ä' || varde=='ö' || varde=='Å'
|| varde=='Ä' || varde=='Ö' || varde==228 )
{
if(kont==0)
{
putc(39,fo);
kont=1;
}

```

```
    }
    putc(varde,fo);
    tecken=1;
  }
  else
  {
    putc(',',fo);
  }
}
else
{
  if(tecken==1)
  {
    tecken=0;
    if(kont==1)
    {
      //Text slut in med fnutt
      putc(39,fo);
      kont=0;
    }
    varde=getc(fi);
    /*if(varde=='S')
    {
      printf("Varde\n");
    }*/
    if(varde==EOF)break;
    if(varde=='\t')
    {
      putc('\n',fo);
    }
    else
    {
      if(varde!='\n')
      {
        tecken=1;
      }
      putc(',',fo);
      putc(varde,fo);
    }
  }
  else
  {
    putc('\n',fo);
  }
}
}
}
}
fflush(fo);
}
while(varde!=EOF);
fclose(fi);
fclose(fo);
}
```

Bilaga B

Nya SOLWIN

```
#define STRICT
#define WIN31
#include "owl.h"
#include "filewnd.h"
#include "string.h"
#include "stdio.h"
#include "math.h"
#include "process.h"
#define CM_ABOUT 101
#define CM_EX 201
#define CM_FILER 102
#define CM_KOR 103
#define CM_EXI 104
#define CM_INDATA 400
#define PI 3.14159265
#define DAG 5
#define SOL_HLP 301

/*****/
class TStigApp : public TApplication
{
public:
TStigApp(LPSTR Name, HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine,
int nCmdShow) : TApplication( Name, hInstance, hPrevInstance, lpCmdLine, nCmdShow)
{
};
virtual void InitMainWindow();
virtual void InitInstance();
};
/*****/

_CLASSDEF(TMyWindow)
class TSolWindow : public TWindow
{
public:
TSolWindow(PtWindowsObject AParent, LPSTR ATitle); //Declaration
virtual void GetWindowClass(WNDCLASS _FAR & AWndClass);
virtual void CMAbout(RTMessage Msg) = [CM_FIRST + CM_ABOUT];
virtual void CMFiler(RTMessage Msg) = [CM_FIRST + CM_FILER];
virtual void CMExi(RTMessage Msg) = [CM_FIRST + CM_EXI];
virtual void CMKor(RTMessage Msg) = [CM_FIRST + CM_KOR];
void ShowEditWin(PtWindowsObject AParent, LPSTR EdTitle, LPSTR FileName);
virtual void CMHjalp(RTMessage Msg) = [CM_FIRST + CM_INDATA];
virtual void Paint(HDC PaintDC, PAINTSTRUCT&);
};
/*****/

class TEdWindow : public TFileWindow
{
```

```

public:
    TEdWindow(PWindowsObject AParent, LPSTR EdTitle, LPSTR FileName); //Declaration
    virtual void CMEx(RTMessage Msg) = [CM_FIRST + CM_EX];

};

/*****/

HWND MinRuta, HWin;
PAINTSTRUCT ps;
HDC StigDC;
char lager[80], lager1[1000], text[100];
RECT rect;
float summan1=0.;

float direct(float lat, float azyta, float b, int forsta, int sista, float
fakt1, float fakt2, float fakt3, float ha, float rm, float rc);
float *transm(float rikt, float absco);

void TStigApp::InitInstance()
{
    TApplication::InitInstance();
    if ( Status == 0 )
        HAccTable = LoadAccelerators(hInstance, "FILECOMMANDS");
}

TSolWindow::TSolWindow(PWindowsObject AParent, LPSTR ATitle) : TWindow(AParent, ATitle)
{
    Attr.Style=WS_OVERLAPPEDWINDOW|WS_VSCROLL|WS_HSCROLL;
    Attr.X=100;
    Attr.Y=100;
    Attr.W=700;
    Attr.H=500;
    AssignMenu("About");
    Scroller = new TScroller(this, 5, 5, 200, 200);
}

void TStigApp::InitMainWindow()
{
    MainWindow=new TSolWindow(NULL,Name); //NULL för att det är ett main window
}

TEdWindow::TEdWindow(PWindowsObject AParent, LPSTR EdTitle, LPSTR FileName):
TFileWindow(AParent,EdTitle,FileName)//Constructor
{
    Attr.Style=WS_VISIBLE|WS_OVERLAPPEDWINDOW;
    Attr.X=300;
    Attr.Y=300;
    Attr.W=500;
    Attr.H=300;
    AssignMenu("FILECOMMANDS");
}

void TSolWindow::CMAbout(RTMessage)
{
    GetApplication()->ExecDialog(new TDialog(this, "AboutBox"));
}

void TSolWindow::CMEx( RTMessage Msg )
{
    WinHelp(HWindow, "c:\\opera\\ophlp.hlp", OL, HELP_QUIT);
    TWindow::CMExit( Msg );
}

void TSolWindow::Paint(HDC StigDC, PAINTSTRUCT&) //Här ritas text vid scrolling

```

```

{
    DrawText(StigDC,lager1,-1,&rect,DT_LEFT | DT_WORDBREAK | DT_EXPANDTABS);
}

void TSolWindow::CMHjalp( RTMessage)
{
WinHelp(HWindow,"c:\\opera\\ophlp.hlp",HELP_CONTEXT,SOL_HLP);
    return;
}

void TSolWindow::CMFiler(RTMessage)
{
    ShowEditWin(this,"Edit","");//Visa edit-fönster
    MinRuta=HWindow;
}

void TSolWindow::GetWindowClass(WNDCLASS _FAR & AWndClass)
{
    TWindow::GetWindowClass(AWndClass);
    AWndClass.hIcon = LoadIcon(GetApplication()->hInstance, MAKEINTRESOURCE(1));
}

void TSolWindow::ShowEditWin( PWindowsObject AParent,LPSTR ATitle, LPSTR FileName)

{
GetApplication()->MakeWindow( new TEDWindow( AParent, ATitle, FileName) );
}

void TEDWindow::CMEEx(RTMessage)
{
    CloseWindow();
    SetFocus(MinRuta);
}

void TSolWindow::CMKor(RTMessage)
{
float varde[30],fakt1,fakt2,fakt3,ha,rm,mansum[3][12],mansu1[3][12],
typer[2][12],resultat[12];
static int forsta=1,sista=0,dagtyp=0;
static int dagar[12] = {31,28,31,30,31,30,31,31,30,31,30,31};
static float rco[12]={1.0315,1.0235,1.0103,0.9913,0.9757,0.968,
0.968,0.9757,0.9898,1.0087,1.0238,1.0318};
int manad,i,strlangd,res;
FILE *in;

    StigDC=GetDC(HWindow);
    HWin=HWindow;
if((in = fopen("c:\\opera\\SUN.DAT","r")) != NULL)
{
int i=0;
while(fscanf(in,"%f",&varde[i]) != EOF)
{
i++;
}
fclose(in);
ha=0.;
rm=0.2;
for(dagtyp=0;dagtyp<=2;dagtyp++)
{
/*Här kommer värden för olika dagtyper*/
switch(dagtyp)
{
case 0:
/*Här sätts faktorer för en helklar dag*/
fakt1=0.9;
fakt2=0.2;
fakt3=0.04;
GetClientRect(HWindow,&rect);
InvalidateRect(HWindow,&rect,TRUE);
BeginPaint(HWindow,&ps);
res=sprintf(text,"Calculates for CLEAR DAYS");
TextOut(StigDC,0,0,text,res);
}
}
}
}

```

```

                                EndPaint(HWindow,&ps);
break;
    case 1:
        /*Här sätts faktorer för en halvklar dag*/
        fakt1=0.52;
        fakt2=0.38;
fakt3=0.032;

                                InvalidateRect(HWindow,&rect,TRUE);
                                BeginPaint(HWindow,&ps);
                                res=sprintf(text,"Calculates for HALF-CLEAR DAYS");
TextOut(StigDC,0,0,text,res);
break;
    case 2:
        /*Här kommer värden för en mulen dag*/
        fakt1=0.1;
        fakt2=0.35;
fakt3=0.016;

                                InvalidateRect(HWindow,&rect,TRUE);
                                BeginPaint(HWindow,&ps);
                                res=sprintf(text,"Calculates for OVERCAST DAYS");
TextOut(StigDC,0,0,text,res);
                                EndPaint(HWindow,&ps);
break;
}

/*Här skall man se till att rätt antal dagar används*/
for(manad=0;manad<=11;manad++)
{
    sista+=dagar[manad];
    mansum[dagtyp][manad]=direct(varde[0],varde[1],
varde[2],forsta,sista,fakt1,fakt2,fakt3,ha,rm,rc[manad]);
    mansu1[dagtyp][manad]=summan1;
    /*printf("Den totala solstrålningen i månad %2d blev = %5.2f\n",manad+1,mansum[dagtyp][manad]);*/
    /*printf("The total radiation through a window month %2d became: %6.2f\n",manad+1,summan1);*/
    forsta=sista+1;
    summan1=0.;
}
forsta=1;
sista=0;
}
else
{
MessageBox(HWindow,"Could not find the input data file!!","ERROR",MB_ICONEXCLAMATION);
return;
}
//printf("\n\n");
/*printf("Nu har programmet kommit till slutet\n");*/
/*printf("      SOLAR RADIATION IN kWh/SQ.M. AT THE OUTSIDE AND THE INSIDE OF THE WINDOW \n");
printf("-----\n");
printf("Month      Clear days      Half-clear days      Overcast days\n");
printf(" nr          Outside  Inside      Outside  Inside      Outside  Inside\n");
for(manad=0;manad<=11;manad++)
printf(" %3d      %6.1f      %6.1f      %6.1f      %6.1f      %6.1f      %6.1f\n",
manad+1,mansum[0][manad]/1000.,mansu1[0][manad]/1000.,mansum[1][manad]/1000.,mansu1[1][manad]/1000.,
mansum[2][manad]/1000.,mansu1[2][manad]/1000.);*/

/* Här ska nu tas hänsyn till tvåglasfönster samt antal klara, halvklara
samt mulna dagar*/
i=3;
for(dagtyp=0;dagtyp<=1;dagtyp++)
{
for(manad=0;manad<=11;manad++)
{
typer[dagtyp][manad]=varde[i];
    i++;
}
}

wsprintf(lager,"SOLAR ENERGY THROUGH A TWO PANE WINDOW\n");
strcpy(lager1,lager);
wsprintf(lager,"=====\n");
strcat(lager1,lager);
sprintf(lager,"The normal of the window directed %5.1f degrees clockwise from the south\n",varde[1]);

```



```

strcat(lager1,lager);
    strcat(lager1,"\n");
wsprintf(lager,"\t\tMonth\t\tEnergy \n");
strcat(lager1,lager);
wsprintf(lager,"\t\tno\t\tkWh/sq.m\n");
strcat(lager1,lager);
    for(manad=0;manad<=11;manad++)
{
resultat[manad]=
0.9*((mansu1[0][manad]*typer[0][manad])+(mansu1[1][manad]*(dagar[manad]-typer[0][manad]-typer[1][manad]))+
(mansu1[2][manad]*typer[1][manad]))/dagar[manad];
sprintf(lager,"\t\t%3d\t\t%6.2f\n",manad+1,resultat[manad]/1000.);
    strcat(lager1,lager);
}

    strcat(lager1,"\0");
    GetClientRect(HWindow,&rect);
    InvalidateRect(HWindow,&rect,TRUE);
    BeginPaint(HWindow,&ps);
DrawText(StigDC,lager1,-1,&rect,DT_LEFT | DT_WORDBREAK | DT_EXPANDTABS);
    EndPaint(HWindow,&ps);
    ReleaseDC(TSolWindow::HWindow,StigDC);
//printf("The program has now come to its end\n");
}

```

```

/* Här kommer en funktion för att beräkna den direkta solinstrålningen*/

```

```

float direct(float lat,float azyta,float b,int forsta,int sista,float fakt1,float fakt2,float fakt3,
float ha,float rm,float rc)

```

```

{
float dekl,timv,c1,c2,azim,idn,latr,timvun,tupp,tner,hojdv;
float azimd,sumidn,sumidn1,azytar,irad,igrad,brad;
float idn1,azres,azimpos,idirh,idiffh,threl,rikt,absor,absco,trans;
float cv,itotal,itoth,idiffb,idiffm,idn2,summan,iehim,iemark,idn3=0.;
float *pektrans,itot=0.;
float midn3=0.,himdiff,absorhim,transhim,absorm,transma,markdiff;
float totstral=0.;
double uttr;
int koll,flagga;
register int kll,dagnr;
unsigned int kl;
int res;
char text[100];
HDC StigDC;

StigDC=GetDC(HWin);
/*Här sätts absorbtionscoefficienten till 0.07*/
absco=0.07;
/*printf("Latituden blev = %5.2f grader N\n",lat);*/
/*printf("Ytans azimut = %5.2f grader räknat från söder och medurs.\n",azyta);*/
/*printf("Ytans lutning mot horisontalplanet = %5.2f grader \n",b);*/

latr=lat*2.*PI/360.;/* Latituden ska vara i radianer */
summan=0.;
summan1=0.;
/*printf("I början av direkt är summan1= %6.2f\n",summan1);*/
/*printf("                midn3= %6.2f\n",midn3);*/
/*printf("                idn3= %6.2f\n",idn3);*/

/*Här börjar beräkningarna för varje månad*/

if(forsta>365)MessageBox(HWin,"Utänför året!!!","ERROR",MB_ICONECLAMATION);

for(dagnr=forsta;dagnr<=sista;dagnr++)
{
//MessageBox(HWin,"Kontroll!!!","ERROR",MB_ICONECLAMATION);
res=sprintf(text,"Now calculating on day no");
TextOut(StigDC,0,30,text,res);
res=sprintf(text,"%d",dagnr+1);
TextOut(StigDC,300,30,text,res);
}
}

```

```

c2=0.017214*(dagnr-2.8749);
c1=c2+0.0334*sin(c2)+1.78128;
dekl=asin(-0.3979*sin(c1));

/* Först måste beräknas när solen går upp och ner*/
/*if(dagnr==DAG)
{
printf("                CALCULATIONS FOR DAY NO %3d\n",dagnr);
printf("\n");
printf("                Tidsperiod = från dag nr %4d till %4d \n",forsta,sista);
printf("                =====\n");
}*/
timvun=acos(-(sin(latr)*sin(dekl)/(cos(latr)*cos(dekl))));
tupp=12.-timvun*360./(2.*PI*15.);
tner=12.+timvun*360./(2.*PI*15.);
/*printf(" Timvinkel vid solnedgång i radianer = %5.2f\n",timvun);*/
/*printf(" Solen går upp kl %5.2f och går ner kl %5.2f\n",tupp,tner);*/
kl=(int)tupp;
kl1=(int)tner;
/*printf(" Heltalsdelen av soluppgången = %5d \n",kl); */
kl++;
/*if(dagnr==DAG)
{
printf("True Eleva- Solar Solar Beta Solar Diffuse Diffuse Total Total\n");
printf("tion azimuth radia. angle radia. radia. radia. radia. radia.\n");
printf("time angle IDN IDN1 sky ground outside inside\n");
printf("[ h ] [deg] [ deg ] [W/sq.m] [deg] [W/sq.m] [W/sq.m] [W/sq.m] [W/sq.m] [W/sq.m]\n");
printf("-----\n");
}*/

sumidn=0.;
itotal=0.;
sumidn1=0.;
koll=2;
while(kl<=kl1)
{
/*printf(" Nu är klockan = %5d \n",kl);*/
timv=(kl-12.)*15.*2.*PI/360.; /* Timvinkel i radianer */
hojdv=asin(sin(latr)*sin(dekl)+cos(latr)*cos(dekl)*cos(timv));
azim=asin(cos(dekl)*sin(timv)/cos(hojdv));
uttr=(sin(dekl)*cos(latr)-cos(dekl)*sin(latr)*cos(timv))/cos(hojdv);
if(uttr<=-1.) uttr=-0.999999;
//azim1=acos(uttr);
/*printf(" Azimuten beräknad på enkla sättet blev = %6.4f radianer. \n", azim);*/
/*printf(" Detta är latr,c2,c1,dekl = %5.2f , %5.2f , %5.2f , %5.2f
\n",latr,c2,c1,dekl);*/
/*printf(" Detta är timv,höjdv,azim = %5.2f , %5.2f , %5.2f\n",timv,hojdv,azim);*/
if(hojdv>=0.001)
idn=fakt1*rc*991.64*exp(-0.09143/sin(hojdv));
else
idn=0.;
idn2=idn;
/*printf("Direkt strålning mot normalen = %7.2f \n",idn);*/
/* Här ska jag ta hänsyn till hur ytan lutar.*/
brad=b*2.*PI/360.;
azytar=azyta*2.*PI/360.;
azres=azim-azytar;
azimd=azim*360./(2.*PI);
if(azimd<=0.)
{
azimpos=360.+azimd;
/*printf("azimpos = %7.4f grader\n",azimpos);*/
if(azimpos>=azyta+90.)
{
if(azimpos>=azyta+270.)
{
/*printf("Framsidan.Bidrag.\n");*/
flagga=1;
}
else
{
/*printf("Baksidan.Inget bidrag\n");*/
idn=0.;
azres=(azimpos-azyta)*2.*PI/360.;
}
}
}
}
}

```

```

}
else
{
/*printf("Nu lyser solen på framsidan. Bidrag.\n");*/
flagga=1;
}
}
else
{
if(azimd<=(azyta-90.))
{
/*printf("Här lyser solen på baksidan. Inget bidrag.\n");*/
idn=0.;
azres=azytar-azim;
}
else
{
/*printf("Här blir det bidrag. \n");*/
flagga=1;
}
}
}
irad=acos(sin(hojdv)*cos(brad)+cos(hojdv)*sin(brad)*cos(azres));
igrad=irad*360./(2.*PI);
/*printf(" irad = %7.4f , igrad = %7.4f \n",irad,igrad);*/
/*printf(" Här i funktionen direkt ( ) blev idn = %5.2f\n",idn);*/
idn1=idn*cos(irad);
/*Här kommer direkt strålning mot horisontell yta.*/
idirh=fakt1*idn2*sin(hojdv);
/*printf("Dir strålning mot horisontell yta = %7.2f W/kv.m\n",idirh);*/
idiffh=(fakt2*sin(hojdv)+fakt3)*idn2;
/*printf("Diff strål mot horisontell yta = %7.2f W/kv.m\n",idiffh);*/
/*Threlkelds faktor*/
/*printf("solazimut-ytazimut i radianer = %5.3f\n",azres);*/
cv=cos(hojdv)*cos(azres);
threl=0.55+0.437*cv+0.313*cv*cv;
/*printf("cv = %7.3f threl = %7.3f \n",cv,threl);*/
if(threl<=0.45) threl=0.45;
idiffb=(1+fakt1*(2.*threl-1.)*sin(brad)*sin(brad))*0.5*(cos(ha)+cos(brad))*idiffh;
/*printf("Diffus strålning mot lutande yta är = %7.2f \n",idiffb);*/
itoth=idirh+idiffh;
idiffm=rm*0.5*(2.-cos(ha)-cos(brad))*itoth;
/*printf("Strålning reflekterad från marken = %7.2f \n",idiffm);*/
/*Total strålning blir direkt+diffus+mark */
itotal=idn1+idiffb+idiffm;
/*printf("Total strålning blev = %7.2f W/kv.m \n",itotal);*/
//hojdv=hojdv*360./(2.*PI);
/*Här ska beräknas strålningen genom ett fönster*/
/*Först ny vinkel Se medd. 146 sid 8*/
iehim=59.68-0.1388*b+0.001497*b*b;
iemark=90.-0.5788*b+0.002693*b*b;
/*printf("Vinkeln = %5.2f iehim=%5.2f iemark= %5.2f \n"
,b,iehim,iemark);*/
/*Här kommer transmissionsberäkning. Se B&I sid 162 */
/*Först beräknas för direkt instrålning*/
if(flagga==1)
{
/*printf("Här lyser det på framsidan. Alltså direkt strålning\n");*/
rikt=igrad;
pektrans=transm(rikt,absco);
trans=*pektrans;
absor=(pektrans+1);
/*printf("trans = %6.2f absor = %6.2f \n",trans,absor);*/
idn3=idn1*trans+0.3*idn1*absor; /*Se medd. 146 sid 9*/
/*printf("idn = %6.2f idn1 = %6.2f idn3 = %6.2f\n",idn,idn1,idn3);*/
}
/* Här skall diffusa strålningen in genom fönstret*/
/* Först tar jag himmelsstrålning */
rikt=iehim;
pektrans=transm(rikt,absco);
transhim=*pektrans;
absorhim=(pektrans+1);
/*printf("Transmissionsfaktor vid himmelsstrålning blev = %6.2f\n",transhim);*/
/*printf("Absorptionsfaktor vid him.stråln. = %6.2f\n",absorhim);*/
/* Här kommer markreflekterad strålning*/

```

```

rikt=iemark;
pektrans=transm(rikt,absc0);
transma=*pektrans;
absorm=(pektrans+1);
himdiff=idiffb*(transhim+0.3*absorhim);
/*printf("Den resulterande him.stråln. blev = %6.2f\n",himdiff);*/
markdiff=idiffm*(transma+0.3*absorm);
/*printf("Transma = %6.2f Absorm = %6.2f\n",transma,absorm);*/
/*printf("Den resulterande markstrålningen blev = %6.2f\n",markdiff);*/
/* Här kommer total strålning som transmitterats genom fönstret*/
totstral=himdiff+markdiff+idn3;
/*printf("Total strålning genom fönstret blev = %6.2f\n",totstral);
    if(dagnr==DAG)
    {
printf(" %2d %4.1f %6.1f %6.1f %6.1f %6.1f %5.1f %5.1f %5.1f %5.1f\n",
kl,hojvd,azimd,idn,igrad,idn1,idiffb,idiffm,itotal,totstral);
    }
printf(" koll = %2d \n",koll);*/
    if(koll == 2)
    {
sumidn=-(tupp-kl)*itotal/2.;
/*printf("tupp,kl,idn = %5.2f %5d %5.2f \n",tupp,kl,idn);*/
/*printf("sumidn = %5.2f \n",sumidn);*/
koll = 1;
    }
sumidn+=itotal;
kl++;
flagga=0;
/*printf("idn3 = %6.2f midn3 = %6.2f\n",idn3,midn3);*/
/*printf("pekm = %6.2f\n",*pekm);*/
midn3+=totstral;
/*printf("Nu efter summan är idn3= %6.2f\n",idn3);*/
/*printf("Omväg via pekare ger efter summan = %6.2f\n",*pekm);*/
/*printf("Adressen är nu = %6d\n",&midn3);*/
/*printf("midn3 efter summan = %6.2f\n",midn3);*/
idn3=0.;
totstral=0.;
itot+=itotal;
    }
sumidn+=-(tner-kl)*itotal/2.;
/*printf("tner = %4.2f kl = %3d \n",tner,kl);*/
/*printf("Bidrag på slutet blir %7.4f \n",- (tner-kl)*idn1/2.);*/
sumidn1=sumidn;
summan+=sumidn1;
/*printf("Före summering är summan1 = %6.2f och midn3= %6.2f\n",summan1,midn3);*/
summan1+=midn3;

/*if(dagnr==DAG)
{
printf("-----\n");
printf("\n");
printf("The sum of the total outside radiation day no. %3d equals %4.1f Wh/sq.m.\n",dagnr,itot);
printf("The sum of the total inside radiation day no. %3d equals %4.1f Wh/sq.m.\n",dagnr,midn3);
printf("Dag nr %3d är summan = %7.2f W/kv.m\n",dagnr,summan);
printf("Dag nr %3d är midn3 = %7.2f W/kv.m\n",dagnr,midn3);
printf("Dag nr %3d är summan1 = %7.2f W/kv.m\n",dagnr,summan1);
}*/
sumidn1=0.;
midn3=0.;
itot=0.;
    }
/*printf("summan1 = %6.2f\n",summan1);*/
if(sista==365)
{
res=sprintf(text," ");
TextOut(StigDC,300,30,text,res);
}
ReleasedC(HWin,StigDC);
return(summan);
}

```

/*Här kommer en funktion för att beräkna transmission och
absorbtion i glasrutan*/

```

float *transm(float rikt,float absco)

//float rikt,absco;

{

float *pektrans;
float riktrad,bryti=1.52,taljpar,nammpar,taljvin,namvin,riktpar,riktvin;
float faktor,alfpar,alfvin,r1par,r1vin,alf1vin,alf1par,atot,ttot,sum1;
float sum2,sum3,tpar,tvin,refpar,refvin,reftot,apar,avin;
double roten;

/*printf("Riktning resp abs.coeff blev %6.2f %6.4f \n",rikt,absco);*/
/*Här ska först reflektionsfaktorerna beräknas. Par = parallell
och vin = vinkelrät, se 1:47 och 1:48 i B&I */
/*Om rikt är större än 90 grader blir det fel. Minska med 180 grader*/
if(rikt>90.) rikt = 180.-rikt;
riktrad=rikt*2.*PI/360.;
roten=sqrt(bryti*bryti-sin(riktrad)*sin(riktrad));
/*printf("rikt = %6.2f riktrad = %6.4f roten = %6.4f \n",rikt,riktrad,roten);*/
taljpar=bryti*bryti*cos(riktrad)-roten;
nammpar=bryti*bryti*cos(riktrad)+roten;
riktpar=(taljpar/nammpar)*(taljpar/nammpar);
taljvin=cos(riktrad)-roten;
namvin=cos(riktrad)+roten;
riktvin=(taljvin/namvin)*(taljvin/namvin);
/*printf("riktpar = %6.4f riktvin = %6.4f \n",riktpar,riktvin);*/

/*Här kommer formel 3:8 i B&I */
faktor=1.-(exp(-absco*bryti/roten));
alfpar=(1.-riktpar)*faktor;
alfvin=(1.-riktvin)*faktor;
/*printf("faktor= %6.4f alfpar = %6.4f alfvin = %6.4f \n",faktor,alfpar,alfvin);*/

/*Här kommer formel 3:9 i B&I */
r1par=1.-riktpar;
r1vin=1.-riktvin;
alf1par=1.-alfpar;
alf1vin=1.-alfvin;

refpar=riktpar+(riktpar*(r1par*r1par*alf1par*alf1par)/(1.-riktpar*riktpar*alf1par*alf1par));
refvin=riktvin+(riktvin*(r1vin*r1vin*alf1vin*alf1vin)/(1.-riktvin*riktvin*alf1vin*alf1vin));
reftot=(refpar+refvin)/2.;
/*printf("refpar = %6.4f refvin = %6.4f reftot = %6.4f\n",refpar,refvin,reftot);*/

/*Här kommer formel 3:10 i B&I*/
apar=(alfpar*r1par*(1.+riktpar*alf1par))/(1.-riktpar*riktpar*alf1par*alf1par);
avin=(alfvin*r1vin*(1.+riktvin*alf1vin))/(1.-riktvin*riktvin*alf1vin*alf1vin);
atot=(apar+avin)/2.;
/*printf("apar =%6.4f avin = %6.4f atot = %6.4f \n",apar,avin,atot);*/

/*Här kommer formel 3:11 i B&I */
tpar=(1.-riktpar)*(1.-riktpar)*alf1par/(1.-riktpar*riktpar*alf1par*alf1par);
tvin=(1.-riktvin)*(1.-riktvin)*alf1vin/(1.-riktvin*riktvin*alf1vin*alf1vin);
ttot=(tpar+tvin)/2.;
/*printf("tpar = %6.4f tvin = %6.4f ttot = %6.4f\n",tpar,tvin,ttot);*/

/*Här ska summorna kollas*/
sum1=refpar+apar+tpar;
sum2=refvin+avin+tvin;
sum3=reftot+atot+ttot;
/*printf("sum1 = %4.2f sum2 = %4.2f sum3 = %4.2f\n",sum1,sum2,sum3);*/
if(sum1>1.05||sum1<.95||sum2>1.05||sum2<.95||sum3>1.05||sum3<.95)
printf("Här blev det fel i transmissionsberäkningen\n");

pektrans=&ttot;
*(pektrans+1)=atot;
return(pektrans);
}

int PASCAL WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow)

```

```
{
TStigApp StigApp("SOLWIN",hInstance,hPrevInstance,lpCmdLine,nCmdShow);
StigApp.Run();
return StigApp.Status;
}
```

Bilaga C

KAIZEN.C

```
//Detta program skall utgöra en "äkta" Windowsapplikation
//Programmet skall användas som bas för system Kaizen
//Projektet är finansierat av Byggeforskningsrådet och samarbete har skett
//med Energimarknad Claes Forsberg AB i Skutskär som svarar för den egentliga rapporten
//Programmet skrivet i C++ anpassat för klasser i Borland C++ 3.1
```

```
#define STRICT
#define WIN31
#include "owl.h"
#include "filewnd.h"
#include "string.h"
#include "stdio.h"
#include "io.h"
#include "fcntl.h"
#include "radiobut.h"
#include "math.h"
#define CM_ABOUT 101
#define CM_EX 201
#define CM_FILER 102
#define CM_EXI 104
#define IDM_CLAES 103
#define IDM_OPERA 109
#define IDM_SORAD 110
#define IDM_BIVAL 111
#define CM_INDATA 400
#define IDM_SPEC 105
#define CM_ANDRA 106
#define CM_KLART 107
#define CM_SKIPPA 108
#define CM_VIND 300
#define CM_GOLV 301
#define CM_YTTER 302
#define CM_INNER 303
#define CM_TATNING 304
#define CM_TVAG 306
#define CM_TREG 307
#define CM_ENERGI 308
#define CM_GAS 309
#define CM_FRAN 310
#define CM_VENT 305
#define ID_CLAESDIA 200
#define ID_ISO 201
#define ID_PANNA 202
#define ID_ISOEDIT 122
#define ID_ISOAAAR 123
#define CM_BEFINT 320
#define CM_NYOLJA 321
#define CM_ELVARME 322
#define CM_FJAERRV 323
#define CM_VAERMEP 324
#define CM_NATURGAS 325
#define CM_DISDIFF 326
```

```

#define CM_ELDIFF 327
#define CM_BIVTYP1 328
#define CM_BIVTYP2 329
#define CM_SKIP 330
#define ID_PANNA1 210
#define ID_PANNAR 211
#define ID_PANNEF 212
#define ID_TIDPKT 215
#define ID_TID 216
#define ID_NORR 218
#define ID_OST 219
#define ID_SYD 220
#define ID_VAST 221
#define ID_WINAR 222
#define ID_FONSTER 224
#define CM_KLART1 225
#define CM_SKIPPA1 226

struct TKnappStruct {
    BOOL Vind;
    BOOL Golv;
    BOOL Ytter;
    BOOL Inner;
    BOOL Tatning;
    BOOL Tvag;
    BOOL Treg;
    BOOL Energi;
    BOOL Gas;
    BOOL Vent;
    BOOL Fran;
};

struct TPannaStruct{
    BOOL Befint;
    BOOL Nyolja;
    BOOL Elvaerme;
    BOOL Fjaerrv;
    BOOL Vaermep;
    BOOL Naturgas;
    BOOL Disdiff;
    BOOL Eldiff;
    BOOL Bivtyp1;
    BOOL Bivtyp2;
};

struct TIsoStruct
{
    char VindIso[3];
    char Aar[3];
};

struct TPanna1Struct
{
    char Pannar[3];
    char Pannef[3];
};

struct TTidStruct
{
    char Tid[3];
};

struct TRotStruct
{
    float Aartal;
    int Aatgaerd;
    float Transdiff;
    float Ventdiff;
};

struct TFonStruct
{

```



```

        BOOL Norr;
        BOOL Ost;
        BOOL Syd;
        BOOL Vast;
        char WinAar[3];
    };

    /*****/
class TStigApp : public TApplication
{
public:
TStigApp(LPSTR Name, HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine,
        int nCmdShow) : TApplication( Name, hInstance, hPrevInstance, lpCmdLine, nCmdShow)
    {};
    virtual void InitMainWindow();
    virtual void InitInstance();
};
    /*****/

    _CLASSDEF(TMyWindow)
class TMyWindow : public TWindow
{
public:

    TKnappStruct TransferStruct;
    TIsoStruct IsoStruct;
    TPannaStruct PannaStruct;
    TPanna1Struct Panna1Struct;
    TTidStruct TidStruct;
    TFonStruct FonStruct;
    TMyWindow(PtWindowsObject AParent, LPSTR ATitle); //Declaration
    virtual void GetWindowClass(WNDCLASS _Far & AWndClass);
    virtual void CMAbout(RTMessage Msg) = [CM_FIRST + CM_ABOUT];
    virtual void CMFiler(RTMessage Msg) = [CM_FIRST + CM_FILER];
    virtual void CMExi(RTMessage Msg) = [CM_FIRST + CM_EXI];
    void ShowEditWin(PtWindowsObject AParent, LPSTR EdTitle, LPSTR FileName);
    virtual void CMOpera(RTMessage Msg) = [CM_FIRST + IDM_OPERA];
    virtual void CMSorad(RTMessage Msg) = [CM_FIRST + IDM_SORAD];
    virtual void CMBival(RTMessage Msg) = [CM_FIRST + IDM_BIVAL];
    virtual void CMClaes(RTMessage Msg) = [CM_FIRST + IDM_CLAES];
    virtual void CMHjalp(RTMessage Msg) = [CM_FIRST + CM_INDATA];
    virtual void CMSpec(RTMessage Msg) = [CM_FIRST + IDM_SPEC];
    virtual void Paint(HDC PaintDC, PAINTSTRUCT&);

};
    /*****/

class TEdWindow : public TFileWindow
{
public:

    TEdWindow(PtWindowsObject AParent, LPSTR EdTitle, LPSTR FileName); //Declaration
    virtual void CMEx(RTMessage Msg) = [CM_FIRST + CM_EX];

};

    /*****/
    _CLASSDEF(TTransferDialog)
class TTransferDialog : public TDialog {
public:
    TTransferDialog(PtWindowsObject AParent, int ResourceId);
    virtual void CMFardigt(RTMessage Msg)= [ID_FIRST + CM_KLART];
    virtual void CMSkippa(RTMessage Msg)= [ID_FIRST + CM_SKIPPA];
};

    /*****/
    _CLASSDEF(TIsoDialog)
class TIsoDialog : public TDialog
{
public:

    TIsoDialog(PtWindowsObject AParent, int ResourceId);

```

```

};

/*****/

_CLASSDEF(TPannaDialog)
class TPannaDialog : public TDialog {
public:
    TPannaDialog(PWindowsObject AParent, int ResourceId);
    virtual void CMSkip(RTMessage Msg)= [ID_FIRST + CM_SKIP];
};

/*****/

_CLASSDEF(TPannaDialog)
class TPannaDialog : public TDialog
{
public:

    TPannaDialog(PWindowsObject AParent, int ResourceId);
};

/*****/

_CLASSDEF(TTidDialog)
class TTidDialog : public TDialog
{
public:

    TTidDialog(PWindowsObject AParent, int ResourceId);
};

_CLASSDEF(TFonDialog)
class TFonDialog : public TDialog
{
public:
    virtual void CMFardigt1(RTMessage Msg)= [ID_FIRST + CM_KLART1];
    virtual void CMSkip1(RTMessage Msg)= [ID_FIRST + CM_SKIPPA1];
    TFonDialog(PWindowsObject AParent, int ResourceId);
};

/*****/
//Funktioner

void SkipLine(int to),EneBal(char Btyp[50]),GradTim(float Klim[12]);
void Livstidskostnad(int Vsys,float Area[8],float In[8],float NL[5],float L[5],float Optb,
float R1,float NuvIsoKost,float BrPris[12],float Vgvar,float Q,float Lva,
float PannKost[5][6],float Ef,float Restv,int k,float tin,float tdim,
float Fjv[6],int Befpanna,float RestvSkal);
float ReadLine(int to),SkalRot(float Area[8],float In[8],float NL[5],float L[5],float Optb,float R1);
float Nuvarde(float S0,float L1,float L2,float Opt,float R1);
float Taxor(int Vsys,float BrPris[12],float Fjv[6],int Bytt);

//Andra deklarationer
HWND MinRuta;
FILE *fh;
float vekt[12],grad[12],tin,enem[12];

float trans,vent,Area[8],Oms,Tv[12],
Gratis[12],Sol[4][12],eneres[12];

PAINTSTRUCT ps;
HDC StigDC;
RECT rect;
char text[5000];
int Fardigt=0,Fardigt1=0,sca=0;//För dialogbox
TRotStruct RotStruct[10];

void TStigApp::InitInstance()
{
    TApplication::InitInstance();
    if ( Status == 0 )
        HAccTable = LoadAccelerators(hInstance, "FILECOMMANDS");
}

```

```

TEdWindow::TEdWindow(PWindowsObject AParent, LPSTR EdTitle, LPSTR FileName):
TFileWindow(AParent,EdTitle,FileName)//Constructor
{
Attr.Style=WS_VISIBLE|WS_OVERLAPPEDWINDOW;
Attr.X=300;
Attr.Y=300;
Attr.W=500;
Attr.H=300;
AssignMenu("FILECOMMANDS");
}

TMyWindow::TMyWindow(PWindowsObject AParent, LPSTR ATitle) : TWindow(AParent, ATitle)
{
Attr.Style=WS_VISIBLE|WS_OVERLAPPEDWINDOW|WS_VSCROLL|WS_HSCROLL;

Attr.X=100;
Attr.Y=100;
Attr.W=600;
Attr.H=400;
AssignMenu("About");
Scroller = new TScroller(this, 15, 15, 80, 200);
TransferStruct.Vind = FALSE;
TransferStruct.Golv = FALSE;
TransferStruct.Ytter = FALSE;
TransferStruct.Inner = FALSE;
TransferStruct.Tatning = FALSE;
TransferStruct.Fran = FALSE;
TransferStruct.Tvag = FALSE;
TransferStruct.Treg = FALSE;
TransferStruct.Energi = FALSE;
TransferStruct.Gas = FALSE;
TransferStruct.Vent = FALSE;
strcpy(IsoStruct.VindIso,"0");
strcpy(IsoStruct.Aar,"0");
PannaStruct.Befint=TRUE;
PannaStruct.Nyolja=FALSE;
PannaStruct.Elvaerme=FALSE;
PannaStruct.Fjaerrv=FALSE;
PannaStruct.Vaermep=FALSE;
PannaStruct.Naturgas=FALSE;
PannaStruct.Disdiff=FALSE;
PannaStruct.Eldiff=FALSE;
PannaStruct.Bivtyp1=FALSE;
PannaStruct.Bivtyp2=FALSE;
strcpy(TidStruct.Tid,"0");
FonStruct.Norr = FALSE;
FonStruct.Ost=FALSE;
FonStruct.Syd=FALSE;
FonStruct.Vast=FALSE;
strcpy(FonStruct.WinAar,"0");
MinRuta=HWindow;
}

TTransferDialog::TTransferDialog(PWindowsObject AParent, int ResourceId)
: TDialog(AParent, ResourceId)
{
//MessageBox(HWindow,"Nu är jag här 1","!!!!OBS!!!!",MB_OK|MB_ICONEXCLAMATION);
new TRadioButton(this, CM_VIND, NULL);
new TRadioButton(this, CM_GOLV, NULL);
new TRadioButton(this, CM_YTTER, NULL);
new TRadioButton(this, CM_INNER, NULL);
new TRadioButton(this, CM_TATNING, NULL);
new TRadioButton(this, CM_TVAG, NULL);
new TRadioButton(this, CM_TREG, NULL);
new TRadioButton(this, CM_ENERGI, NULL);
new TRadioButton(this, CM_GAS, NULL);
new TRadioButton(this, CM_VENT, NULL);
new TRadioButton(this, CM_FRAN, NULL);
TransferBuffer = (void far*)&((TMyWindow *)Parent)->TransferStruct);
//MessageBox(HWindow,"Nu är jag här 2","!!!!OBS!!!!",MB_OK|MB_ICONEXCLAMATION);
}

```

```

TPannaDialog::TPannaDialog(PtWindowsObject AParent, int ResourceId)
    : TDialog(AParent, ResourceId)
{
    new TRadioButton(this, CM_BEFINT, NULL);
    new TRadioButton(this, CM_NYOLJA, NULL);
    new TRadioButton(this, CM_ELVAERME, NULL);
    new TRadioButton(this, CM_FJAERRV, NULL);
    new TRadioButton(this, CM_VAERMEP, NULL);
    new TRadioButton(this, CM_NATURGAS, NULL);
    new TRadioButton(this, CM_DISDIFF, NULL);
    new TRadioButton(this, CM_ELDIFF, NULL);
    new TRadioButton(this, CM_BIVTYP1, NULL);
    new TRadioButton(this, CM_BIVTYP2, NULL);
    TransferBuffer = (void far*)&(((TMyWindow *)Parent)->PannaStruct);
}

TIsoDialog::TIsoDialog(PtWindowsObject AParent, int ResourceId)
    : TDialog(AParent, ResourceId)
{
    new TEdit(this, ID_ISOEDIT, sizeof(((TMyWindow *)Parent)->IsoStruct.VindIso));
    new TEdit(this, ID_ISOAAAR, sizeof(((TMyWindow *)Parent)->IsoStruct.Aar));
    TransferBuffer = (void far*)&(((TMyWindow *)Parent)->IsoStruct);
}

TPannaDialog::TPannaDialog(PtWindowsObject AParent, int ResourceId)
    : TDialog(AParent, ResourceId)
{
    new TEdit(this, ID_PANNAR, sizeof(((TMyWindow *)Parent)->PannaStruct.Pannar));
    new TEdit(this, ID_PANNEF, sizeof(((TMyWindow *)Parent)->PannaStruct.Pannef));
    TransferBuffer = (void far*)&(((TMyWindow *)Parent)->PannaStruct);
}

TTidDialog::TTidDialog(PtWindowsObject AParent, int ResourceId)
    : TDialog(AParent, ResourceId)
{
    new TEdit(this, ID_TIDPKT, sizeof(((TMyWindow *)Parent)->TidStruct.Tid));

    TransferBuffer = (void far*)&(((TMyWindow *)Parent)->TidStruct);
}

//Dialogruta för fönster
TFonDialog::TFonDialog(PtWindowsObject AParent, int ResourceId)
    : TDialog(AParent, ResourceId)
{
    new TRadioButton(this, ID_NORR, NULL);
    new TRadioButton(this, ID_OST, NULL);
    new TRadioButton(this, ID_SYD, NULL);
    new TRadioButton(this, ID_VAST, NULL);
    new TEdit(this, ID_WINAR, sizeof(((TMyWindow *)Parent)->FonStruct.WinAar));

    TransferBuffer = (void far*)&(((TMyWindow *)Parent)->FonStruct);
}

void TStigApp::InitMainWindow()
{
    MainWindow=new TMyWindow(NULL,Name);//NULL för att det är ett main window
}

void TMyWindow::GetWindowClass(WNDCLASS _FAR & AWndClass)
{
    TWindow::GetWindowClass(AWndClass);
    AWndClass.hIcon = LoadIcon(GetApplication()->hInstance, MAKEINTRESOURCE(1));
}

void TMyWindow::CMFiler(RTMessage)
{
    ShowEditWin(this,"Edit","");//Visa edit-fönster
    MinRuta=HWindow;
}

void TMyWindow::CMAbout(RTMessage)
{

```

```

    GetApplication()->ExecDialog(new TDialog(this, "AboutBox"));
}

void TMyWindow::CMOpera(RTMessage)
{
    WinExec("c:\\for\\er7c.exe",SW_SHOW);
}

void TMyWindow::CMBival(RTMessage)
{
    WinExec("c:\\OPERA\\Bivwin.exe",SW_SHOW);
}

void TMyWindow::CMSorad(RTMessage)
{
    WinExec("c:\\OPERA\\nysol.exe",SW_SHOW);
}

void TMyWindow::CMClaes(RTMessage)
{
    WinExec("c:\\EXCEL\\excel.exe",SW_SHOW);
}

void TMyWindow::CMExi( RTMessage Msg )//Här slutar huvudappen
{
    CloseWindow();
    CMExit(Msg);
    return;
}

void TMyWindow::CMHjalp( RTMessage)//Här slutar huvud-appen
{
    WinHelp(HWindow,"c:\\opera\\ophlp.hlp",HELP_INDEX,NULL);
    return;
}

void TMyWindow::Paint(HDC StigDC,PAINTSTRUCT&)//Här ritas text vid scrolling
{
    DrawText(StigDC,text,-1,&rect,DT_LEFT | DT_WORDBREAK | DT_EXPANDTABS);
}

void TTransferDialog::CMFardigt(RTMessage Msg)
{
    Fardigt=1;
    Cancel(Msg);
}

void TFonDialog::CMFardigt1(RTMessage Msg)
{
    Fardigt1=1;
    Cancel(Msg);
}

void TTransferDialog::CMSkippa(RTMessage Msg)
{
    Fardigt=2;
    //MessageBox(HWindow,"Tryckt på Avbryt","!!!OBS!!!",MB_OK|MB_ICONEXCLAMATION);
    Cancel(Msg);
}

void TFonDialog::CMSkippa1(RTMessage Msg)
{
    Fardigt=2;

    //MessageBox(HWindow,"Tryckt på Avbryt","!!!OBS!!!",MB_OK|MB_ICONEXCLAMATION);
    Cancel(Msg);
}

void TPannaDialog::CMSkip(RTMessage Msg)
{
    Cancel(Msg);
}

void TMyWindow::CMSpec( RTMessage Msg )//Här börjar specialberäkningarna

```

```

{
int i,to,lgh,j,Vsys=0,resul=0,k=-1;
int Ann,Ano,Ans,Anv,fo,Aop,Befpanna;
float tdim,fltemp, Klim[12],An,Ao,As,Av,hojd,ba,LambdaT,NyttK,IsoT,L[5],NL[5],Opta,R,R1,Q,IsoKost;
float LambdaG,LambdaVy,LambdaVi,Lva,In[8],Ko[4],Ak[4],Intv,Aktv,Intr,Aktr,Infy,Akfy,Infe,Akfe;
float Optb,Nuv,IsoKost=0.,BrPris[12],Vgvar,OljePris=0.,Shade[3],PannKost[5][6],Ef,Effekt;
float Bkt,Bkg,Bky,Mk2,Mk3,Mk4,Mk5,Mkn,Mko,Mks,Mkv,RestvSkal=0.,Senlagg;
float Instpanna,Restv=0.,Omdiff,Lita,Kpo,trans1,vent1,trans2,vent2,Elpris[3],Natgas[2],Fjv[6],Fjvdiff[12];
char Vatyp[25],Tecken,IsoAar[3],Btyp[50],text2[3];
MinRuta=HWindow;
//MessageBox(HWindow,"Nu är jag här 1","!!!OBS!!!",MB_OK|MB_ICONEXCLAMATION);
fh=fopen("c:\\excel\\op\\opin.txt","r");
if(fh=NULL)
{
    MessageBox(HWindow,"Filen opin.txt hittas ej","!!!OBS!!!",MB_OK|MB_ICONEXCLAMATION);
    return;
}
fo=open("c:\\excel\\op\\claes.txt",O_TRUNC | O_TEXT | O_WRONLY);
if(fo==1)
{
    MessageBox(MinRuta,"Filen claes.txt kan ej skapas","!!!OBS!!!",MB_OK|MB_ICONEXCLAMATION);
    return;
}
close(fo);
//Här läses alla värden som behövs in
//MessageBox(HWindow,"Nu är jag här 2","!!!OBS!!!",MB_OK|MB_ICONEXCLAMATION);
to=1;
Skipline(to);
ReadLine(4);
Area[0]=vekt[0];//Tak, golv, ytterv, bruksarea resp
Area[1]=vekt[1];
Area[2]=vekt[2];
Area[3]=vekt[2];
ba=vekt[3];
Skipline(1);
sca+=fscanf(fh,"%f,%d,%f,%d,%f,%d,%f,%d",&An,&Ann,&Ao,&Ano,&As,&Ans,&Av,&Anv);
Area[4]=An*Ann;
Area[5]=Ao*Ano;
Area[6]=As*Ans;
Area[7]=Av*Anv;
ReadLine(4);
Bkt=vekt[0];
Bkg=vekt[1];
Bky=vekt[2];
Mk2=vekt[3];
Mkn=Mk2;
Mko=Mk2;
Mks=Mk2;
Mkv=Mk2;
sca+=fscanf(fh,"%f,%f,%f,%f,%f",&L[0],&L[1],&L[2],&L[3],&L[4]); //Är kvar
//Skipline(1);
sca+=fscanf(fh,"%f",&Oms);
i=0;
Skipline(1);
do
{
Tecken=fgetc(fh);
    Vatyp[i]=Tecken;//Läser text på befintligt värmesystem
    i++;
}
while(Tecken!='\0');
Vatyp[i-1]='\0';

sca+=fscanf(fh,"%f,%f,%f",&Ef,&Vgvar,&Lva);
sprintf(text2,"%2.0f",Lva);
strcpy(PannaStruct.Pannar,text2);
Skipline(1);
ReadLine(12);
for(i=0;i<12;i++)Tv[i]=vekt[i];
sca+=fscanf(fh,"%f,%f,%f,%f",&LambdaT,&LambdaG,&LambdaVy,&LambdaVi);
sca+=fscanf(fh,"%f",&Mk3);
sca+=fscanf(fh,"%f",&Mk4);
sca+=fscanf(fh,"%f",&Mk5);
Skipline(1);

```

```

sca+=fscanf(fh,"%f,%f,%f,%f",&NL[0],&NL[1],&NL[2],&NL[3],&NL[4]);
sca+=fscanf(fh,"%f,%f,%f",&Opta,&R,&Q);
R1=1.+R;
Optb=Opta;
SkipLine(1);
sca+=fscanf(fh,"%f,%f,%f",&In[0],&Ko[0],&Ak[0]);
sca+=fscanf(fh,"%f,%f,%f",&In[1],&Ko[1],&Ak[1]);
sca+=fscanf(fh,"%f,%f,%f",&In[2],&Ko[2],&Ak[2]);
sca+=fscanf(fh,"%f,%f,%f,%f",&In[3],&Ko[3],&Ak[3],&hojd);
SkipLine(1);
sca+=fscanf(fh,"%f,%f",&Intv,&Aktv);
sca+=fscanf(fh,"%f,%f",&Intr,&Aktr);
sca+=fscanf(fh,"%f,%f",&Infy,&Akfy);
sca+=fscanf(fh,"%f,%f",&Infe,&Akfe);
In[4]=(Intv/An)+Aktv;
In[5]=(Intv/Ao)+Aktv;
In[6]=(Intv/As)+Aktv;
In[7]=(Intv/Av)+Aktv;
for(i=0;i<5;i++)sca+=fscanf(fh,"%f,%f,%f,%f,%f",&PannKost[i][0],&PannKost[i][1],&PannKost[i][2],
&PannKost[i][3],&PannKost[i][4],&PannKost[i][5]);
SkipLine(2);
ReadLine(12);
for(i=0;i<12;i++)Klim[i]=vekt[i];
SkipLine(3);
sca+=fscanf(fh,"%d,%f,%f,%f",&Aop,&Kpo,&Omdiff,&Lita);
sca+=fscanf(fh,"%d,%f,%f,%f",&lgh,&fltemp,&tin,&tdim);
sprintf(text2,"%2.0f",tdim);
strcpy(Panna1Struct.Pannef,text2);
SkipLine(6);
ReadLine(12);
for(i=0;i<12;i++)Gratis[i]=vekt[i];
for(j=0;j<4;j++)
{
ReadLine(12);
for(i=0;i<12;i++)Sol[j][i]=vekt[i];
}
sca+=fscanf(fh,"%f,%f,%f",&Shade[0],&Shade[1],&Shade[2]);
sca+=fscanf(fh,"%f",&OljePris);

sca+=fscanf(fh,"%f,%f,%f",&Elpris[0],&Elpris[1],&Elpris[2]);
sca+=fscanf(fh,"%f,%f",&Natgas[0],&Natgas[1]);
sca+=fscanf(fh,"%f,%f,%f,%f,%f",&Fjv[0],&Fjv[1],&Fjv[2],&Fjv[3],&Fjv[4],&Fjv[5]);
ReadLine(12);
for(i=0;i<12;i++)Fjvdif[i]=vekt[i];
fclose(fh);
if(sca!=210)MessageBox(HWindow,"Läser fel i indata","!!!OBS!!!",MB_OK|MB_ICONEXCLAMATION);
sca=0;
GradTim(Klim);//Beräknar antalet gradtimmar
trans=Area[0]*Bkt+Area[1]*Bkg+Area[2]*Bky+Area[4]*Mk2+Area[5]*Mk2+Area[6]*Mk2+Area[7]*Mk2;
trans1=trans;
trans2=trans;
vent=ba*Oms*.33*hojd;
vent1=vent;
vent2=vent;
for(i=0;i<12;i++)enem[i]=grad[i]*(trans+vent)/1000.;
strcpy(Btyp,"Utan åtgärder\n");
EneBal(Btyp);
Effekt=(trans+vent)*(tin-tdim)/1000.;
sprintf(text2,"%2.0f",Effekt);
strcpy(Panna1Struct.Pannef,text2);
MinRuta=HWindow;
while(Fardigt!=1&&Fardigt!=2)
{
GetApplication()->ExecDialog(new TTransferDialog(this, ID_CLAESDIA));
if(TransferStruct.Vind)
{
char IsoTjock[3];//Här skall extra isolering hämtas från dialogbox
GetModule()->ExecDialog(new TISODialog(this, ID_ISO));
strcpy(IsoTjock,IsoStruct.VindIso);
IsoT=0.01*atof(IsoTjock);
NyttK=Bkt*LambdaT/(LambdaT+Bkt*IsoT);
Bkt=NyttK;
trans=Area[0]*NyttK+Area[1]*Bkg+Area[2]*Bky+Area[4]*Mkn+Area[5]*Mko+Area[6]*Mks+Area[7]*Mkv;
strcpy(IsoAar,IsoStruct.Aar);//Här hämtas när det skall ske

```

```

Senlagg=atof(IsoAar);
if(Senlagg>L[2])
{
MessageBox(HWindow,"År för stort. Gör om!", "!!!BJÄLKLAGSISO!!!", MB_OK|MB_ICONEXCLAMATION);
return;
}
RestvSkal+=In[0]*Area[0]*(L[0]-Senlagg)/NL[0];
L[0]=Senlagg;
IsoKost=(Ko[0]+Ak[0]*IsoT)*Area[0];
NuvIsoKost+=Nuvarde(IsoKost,L[0],NL[0],Optb,R1);
sprintf(Btyp,"Isolering bjälklag %4.2f m år %3.0f\n",IsoT,L[0]);
for(i=0;i<12;i++)enem[i]=grad[i]*(trans+vent)/1000.;
EneBal(Btyp);
//Här skall ny effekt beräknas, se även nedanstående fall
//Effekt=(trans+vent)*(tin-tdim)/1000.;
//sprintf(text2,"%2.0f",Effekt);
//strcpy(Panna1Struct.Pannef,text2);
TransferStruct.Vind=FALSE;
k++;
RotStruct[k].Aartal=L[0];
RotStruct[k].Aatgaerd=0;
RotStruct[k].Transdiff=trans1-trans;
trans1=trans;
RotStruct[k].Ventdiff=0.;
continue;
}
}
if(TransferStruct.Golv)
{
char IsoTjock[3]; //Här skall extra isolering hämtas från dialogbox
GetModule()->ExecDialog(new TisoDialog(this, ID_ISO));
strcpy(IsoTjock,IsoStruct.VindIso);
IsoT=0.01*atof(IsoTjock);
NyttK=Bkg*LambdaG/(LambdaG+Bkg*IsoT);
trans=Area[0]*Bkt+Area[1]*NyttK+Area[2]*Bky+Area[4]*Mkn+Area[5]*Mko+Area[6]*Mks+Area[7]*Mkv;
Bkg=NyttK;
strcpy(IsoAar,IsoStruct.Aar); //Här hämtas när det skall ske
Senlagg=atof(IsoAar);
if(Senlagg>L[1])
{
MessageBox(HWindow,"År för stort. Gör om!", "!!!YTTERISO!!!", MB_OK|MB_ICONEXCLAMATION);
return;
}
RestvSkal+=In[1]*Area[1]*(L[1]-Senlagg)/NL[1];
L[1]=Senlagg;
IsoKost=(Ko[1]+Ak[1]*IsoT)*Area[1];
NuvIsoKost+=Nuvarde(IsoKost,L[1],NL[1],Optb,R1);
sprintf(Btyp,"Isolering golv %4.2f m år %3.0f\n",IsoT,L[1]);
for(i=0;i<12;i++)enem[i]=grad[i]*(trans+vent)/1000.;
EneBal(Btyp);
/*Effekt=(trans+vent)*(tin-tdim)/1000.;
sprintf(text2,"%2.0f",Effekt);
strcpy(Panna1Struct.Pannef,text2);*/
TransferStruct.Golv=FALSE;
k++;
RotStruct[k].Aartal=L[1];
RotStruct[k].Aatgaerd=1;
RotStruct[k].Transdiff=trans1-trans;
trans1=trans;
RotStruct[k].Ventdiff=0.;
continue;
}
}
if(TransferStruct.Ytter)
{
char IsoTjock[3]; //Här skall extra isolering hämtas från dialogbox
GetModule()->ExecDialog(new TisoDialog(this, ID_ISO));
strcpy(IsoTjock,IsoStruct.VindIso);
IsoT=0.01*atof(IsoTjock);
NyttK=Bky*LambdaVy/(LambdaVy+Bky*IsoT);
trans=Area[0]*Bkt+Area[1]*Bkg+Area[2]*NyttK+Area[4]*Mkn+Area[5]*Mko+Area[6]*Mks+Area[7]*Mkv;
Bky=NyttK;
strcpy(IsoAar,IsoStruct.Aar); //Här hämtas när det skall ske
Senlagg=atof(IsoAar);
if(Senlagg>L[2])
{

```



```

        k++;
        RotStruct[k].Aartal=Vtid;
        RotStruct[k].Aatgaerd=8;
        RotStruct[k].Transdiff=0.;
        RotStruct[k].Ventdiff=vent1-vent;
        vent1=vent;
    }
    if(TransferStruct.Fran) MessageBox(HWindow,"Finns ej ännu","!!!!KNAPPAR!!!!",MB_OK|MB_ICONEXCLAMATION);
    if(TransferStruct.Tvag) MessageBox(HWindow,"Finns ej ännu","!!!!KNAPPAR!!!!",MB_OK|MB_ICONEXCLAMATION);
        if(Fardigt==2)//=Tryckt på avbryt
        {
            TransferStruct.Treg=FALSE;
        }
    continue;
    }
    if(TransferStruct.Treg)
    {
        while(Fardigt1!=1)
        {
            GetModule()->ExecDialog(new TFonDialog(this, ID_FONSTER));
            if(Fardigt==1)//=Tryckt på färdigt
            {
                TransferStruct.Treg=FALSE;
            }
        }
    continue;
    }
    strcpy(IsoAar,FonStruct.WinAar);//Här hämtas när det skall ske
    Senlagg=atof(IsoAar);
    if(Senlagg>L[4])
    {
        MessageBox(HWindow,"Är för stort. Gör om!",!!!!YTTERISO!!!!,MB_OK|MB_ICONEXCLAMATION);
        return;
    }
    L[4]=Senlagg;
    if(FonStruct.Norr)
    {
        MessageBox(HWindow,"Finns ej ännu","!!!!KNAPPAR!!!!",MB_OK|MB_ICONEXCLAMATION);
    }
    if(FonStruct.Ost)
    {
        Mko=Mk3;
        IsoKost=Ano*(Intr+Ao*Aktr);
        NuvIsoKost+=Nuvarde(IsoKost,L[4],NL[4],Optb,R1);
        IsoKost=Ano*(Intv+Ao*Aktv);//Dra bort för tvåglasrot
        NuvIsoKost-=Nuvarde(IsoKost,L[4],NL[4],Optb,R1);
        RestvSkal+=IsoKost*(L[4]-Senlagg)/NL[4];
    }
    trans=Area[0]*Bkt+Area[1]*Bkg+Area[2]*Bky+Area[4]*Mkn+Area[5]*Mko+Area[6]*Mks+Area[7]*Mkv;
    sprintf(Btyp,"Treglasfönster åt öster är %3.Of\n",L[4]);
    for(i=0;i<12;i++)
    {
        enem[i]=grad[i]*(trans+vent)/1000.;
        //Här ska tas hänsyn till skuggeffekten
        Sol[1][i]*(1.-Shade[0]);
    }
    EneBal(Btyp);
    FonStruct.Ost=FALSE;
    k++;
    RotStruct[k].Aartal=L[4];
    RotStruct[k].Aatgaerd=5;
    RotStruct[k].Transdiff=trans1-trans;
    RotStruct[k].Ventdiff=0.;
    trans1=trans;
    continue;
    }
    if(FonStruct.Syd)
    {
        MessageBox(HWindow,"Finns ej ännu","!!!!KNAPPAR!!!!",MB_OK|MB_ICONEXCLAMATION);
    }
    if(FonStruct.Vast)
    {
        Mkv=Mk3;
        IsoKost=Anv*(Intr+Av*Aktr);
        NuvIsoKost+=Nuvarde(IsoKost,L[4],NL[4],Optb,R1);
        IsoKost=Anv*(Intv+Av*Aktv);//Dra bort för tvåglasrot
        NuvIsoKost-=Nuvarde(IsoKost,L[4],NL[4],Optb,R1);
        RestvSkal+=IsoKost*(L[4]-Senlagg)/NL[4];
    }

```

```

trans=Area[0]*Bkt+Area[1]*Bkg+Area[2]*Bky+Area[4]*Mkn+Area[5]*Mko+Area[6]*Mks+Area[7]*Mkv;
printf(Btyp,"Treglasfönster åt väster är %3.Of\n",L[4]);
for(i=0;i<12;i++)
    {
enem[i]=grad[i]*(trans+vent)/1000.;
        Sol[3][i]*(1.-Shade[0]);
    }
    EneBal(Btyp);
        FonStruct.Vast=FALSE;
    k++;
    RotStruct[k].Aartal=L[4];
    RotStruct[k].Aatgaerd=7;
    RotStruct[k].Transdiff=trans1-trans;
    RotStruct[k].Ventdiff=0.;
    trans1=trans;
    continue;
    }
} //Whilesats slutar här
}
if(TransferStruct.Energi) MessageBox(HWindow,"Finns ej ännu","!!!!KNAPPAR!!!!",MB_OK|MB_ICONEXCLAMATION);
if(TransferStruct.Gas) MessageBox(HWindow,"Finns ej ännu","!!!!KNAPPAR!!!!",MB_OK|MB_ICONEXCLAMATION);
if(TransferStruct.Vent) MessageBox(HWindow,"Finns ej ännu","!!!!KNAPPAR!!!!",MB_OK|MB_ICONEXCLAMATION);
if(Fardigt==2)
{
    Fardigt=0;
return;
}
    Fardigt=0;
    //MessageBox(HWindow,"Nu är jag här","!!!!KNAPPAR!!!!",MB_OK|MB_ICONEXCLAMATION);
    for(i=0;i<12;i++)enem[i]=grad[i]*(trans+vent)/1000.;
    strcpy(Btyp,"Alla åtgärder tillsammans\n");
    EneBal(Btyp);
    trans=trans2;
    vent=vent2;

    //Här börjar beräkningar för pannor
    resul=strcmp("'OIL-BOILER'",Vatyp);
    if(resul==0)Befpanna=0;
    GetModule()->ExecDialog(new TPannaDialog(this, ID_PANNA));
    if(PannaStruct.Befint)
    {
        if(fabs(Effekt-Ef)>5.)
    {
resul=MessageBox(HWindow,"Ber. effekt ej lika bef. effekt. ÄNDRA?"
, "!!!!PANNOR!!!!",MB_YESNO|MB_ICONEXCLAMATION);
        if(resul==IDYES)
    {
GetModule()->ExecDialog(new TPannaDialog(this, ID_PANNA1));
            Ef=atof(PannaStruct.Pannef);
        }
        resul=strcmp("'OIL-BOILER'",Vatyp);
        if(resul==0)Vsys=0;
        for(i=0;i<12;i++)BrPris[i]=0ljePris;
        Restv=0.;
    }
        if(PannaStruct.Nyolja) MessageBox(HWindow,"Finns ej ännu","!!!!PANNA!!!!",MB_OK|MB_ICONEXCLAMATION);
        if(PannaStruct.Elvaerme) MessageBox(HWindow,"Finns ej ännu","!!!!PANNA!!!!",MB_OK|MB_ICONEXCLAMATION);
if(PannaStruct.Fjaerrv)
{
    Vsys=2;//Fjärrvärmesystem vad gäller pannkostnader
    GetModule()->ExecDialog(new TPannaDialog(this, ID_PANNA1));
    Instpanna=atof(PannaStruct.Pannar);
    if(Instpanna>Lva+0.5)
{
MessageBox(HWindow,"Är för stort. Gör om!","!!!!PANNA!!!!",MB_OK|MB_ICONEXCLAMATION);
        return;
    }
    resul=strcmp("'OIL-BOILER'",Vatyp);
    if(resul==0)Restv=(PannKost[0][0]+PannKost[0][1]*Ef)*(Lva/PannKost[0][3]);
    Restv=Restv*(Lva-Instpanna)/Lva;
    Restv=Restv*pow(R1,-(Instpanna));
}
}
}

```

```

Lva=Instpanna;
Ef=atof(Panna1Struct.Pannef);
for(i=0;i<12;i++)BrPris[i]=Fjv[0];
}
if(PannaStruct.Vaermep) MessageBox(HWindow,"Finns ej ännu","!!!!PANNA!!!!",MB_OK|MB_ICONEXCLAMATION);
if(PannaStruct.Naturgas)
{
    MessageBox(HWindow,"Finns ej ännu","!!!!PANNA!!!!",MB_OK|MB_ICONEXCLAMATION);
    /*Vsys=4;//Naturgassystem vad gäller pannkostnader
    GetModule()->ExecDialog(new TPanna1Dialog(this, ID_PANNA1));
    Instpanna=atof(Panna1Struct.Pannar);
    if(Instpanna>Lva+0.5)
{
    MessageBox(HWindow,"Är för stort. Gör om!","!!!!PANNA!!!!",MB_OK|MB_ICONEXCLAMATION);
    return;
    }
    resul=strcmp("'OIL-BOILER'",Vatyp);
    if(resul==0)Restv=(PannKost[0][0]+PannKost[0][1]*Ef)*(Lva/PannKost[0][3]);
    Restv=Restv*(Lva-Instpanna)/Lva;
    Restv=Restv*pow(R1,-(Instpanna));
    Lva=Instpanna;
    Ef=atof(Panna1Struct.Pannef);
    for(i=0;i<12;i++)BrPris[i]=Natgas[1];*/
}

if(PannaStruct.Disdiff)
{
    Vsys=6;//Fjärrvärmesystem, Vad gäller pannkostnader = nr 2
    GetModule()->ExecDialog(new TPanna1Dialog(this, ID_PANNA1));
    Instpanna=atof(Panna1Struct.Pannar);
    if(Instpanna>Lva+0.5)
{
    MessageBox(HWindow,"Är för stort. Gör om!","!!!!PANNA!!!!",MB_OK|MB_ICONEXCLAMATION);
    return;
    }
    resul=strcmp("'OIL-BOILER'",Vatyp);

    if(resul==0)Restv=(PannKost[0][0]+PannKost[0][1]*Ef)*(Lva/PannKost[0][3]);
    Restv=Restv*(Lva-Instpanna)/Lva;
    Restv=Restv*pow(R1,-(Instpanna));
    Lva=Instpanna;
    Ef=atof(Panna1Struct.Pannef);
    for(i=0;i<12;i++)BrPris[i]=Fjvdiff[i];
}

if(PannaStruct.Eldiff) MessageBox(HWindow,"Finns ej ännu","!!!!PANNA!!!!",MB_OK|MB_ICONEXCLAMATION);
if(PannaStruct.Bivtyp1) MessageBox(HWindow,"Finns ej ännu","!!!!PANNA!!!!",MB_OK|MB_ICONEXCLAMATION);
if(PannaStruct.Bivtyp2) MessageBox(HWindow,"Finns ej ännu","!!!!PANNA!!!!",MB_OK|MB_ICONEXCLAMATION);

    Livstidskostnad(Vsys,Area,In,NL,L,Optb,R1,Nuv,IsoKost,BrPris,Vgvar,Q,Lva,PannKost,Ef,Restv,
k,tin,tdim,Fjv,Befpanna,RestvSkal);
    return;
}

void Livstidskostnad(int Vsys,float Area[8],float In[8],float NL[5],float L[5],float Optb,float R1,
float Nuv,IsoKost,float BrPris[12],float Vgvar,float Q,float Libp,
float PannKost[5][6],float Eff,float Restv,int k,float tin,float tdim,
float Fjv[6],int Befpanna,float RestvSkal)
{
    float TotKost,SkalR,EneKost,R2,EneKostNuv=0.,NuvPanna=0.,PannSum,NuvSlang,Slang,Vg,
RestvPanna=0.,RestvSlang=0.,Effe,diff,diff1,Ans1Avg=0.,diff2=0.;
    int i,j,RotVekt[10],Snabbast,RotForst[10],k1=0,Aatgaerder[10],k2,koll=0,Panntyp,Koll,Bytt
,Koll1=0;
    char Text2[50],text3[5000],text4[20];

    SkalR=SkalRot(Area,In,NL,L,Optb,R1);
    R2=(1.+Q)/(R1);
    //Sortera RotStruct
    Snabbast=1000;
    for(i=0;i<10;i++)RotForst[i]=0;
    for(i=0;i<=k;i++)RotForst[i]=RotStruct[i].Aartal;
    for(j=0;j<=k;j++)
    {
for (i=0;i<=k;i++)

```

```

{
    if(RotForst[i]<Snabbast)
    {
        Snabbast=RotForst[i];
        if(k1<=i)k1=i;
    }
    RotVekt[j]=Snabbast;
    RotForst[k1]=1000;
    Snabbast=1000;
    Aatgaerder[j]=RotStruct[k1].Aatgaerd;
}

j=0;
Vg=Vgvar;
Panntyp=Befpanna;//Sätt pannan till befintlig panna
Bytt=0;//Ej bytt panna
Koll=0;//Anslutningsavgift?
strcpy(text3,"Energikostnad år och Kr samt diff\n");
for(i=0;i<Optb;i++)
{
    if(i>=RotVekt[j])
    {
        for(k=0;k<=k1;k++)
        {
            if(RotStruct[k].Aatgaerd==Aatgaerder[j])
            {
                diff=trans+vent;
                trans-=RotStruct[k].Transdiff;
                vent-=RotStruct[k].Ventdiff;
                diff-=trans+vent;
                diff2+=diff;
                if(abs(RotStruct[k].Aartal-RotStruct[k+1].Aartal)<0.005)
                {
                    Koll1=1;
                }
            }
        }
        if(Koll1==0)j++;
        Koll1=0;
    }
    if(i==Libp)
    {
        Panntyp=Vsys;//Byt till ny panntyp
        if(Vsys==6)Panntyp=2;

        Vg=PannKost[Panntyp][2];
        Effe=(trans+vent)*(tin-tdim)/1000.;
        if(fabs(Eff-Effe)>0.5 || koll==1)
        { //Här har man inte ändrat effekt enligt förslag!
            Eff-=diff2*(tin-tdim)/1000.;
            koll=1;
        }
        if(fabs(Eff-Effe)<0.5)Eff=Effe;
        PannSum=PannKost[Panntyp][0]+PannKost[Panntyp][1]*Eff;
        if(Panntyp==2&&Koll==0)
        { //Fjärrvärme ansl avg
            AnslAvg=Fjv[1]*Eff;
            Koll=1;
        }

        //NuvPanna=Nuvarde(PannSum,Libp,PannKost[Vsys][3],Optb,R1);
        NuvPanna+=PannSum*pow(R1,-(i));
        Slang=PannKost[Panntyp][4]*Eff;
        //NuvSlang=Nuvarde(Slang,Libp,PannKost[Vsys][5],Optb,R1);
        if(Bytt==0)NuvSlang=Slang*pow(R1,-(i));
        if(Vsys==0)NuvSlang=0;
        Libp+=PannKost[Panntyp][3];
        diff=0;
        Bytt=1;//Bytt panna
    }
}

```

```

        for(k2=0;k2<12;k2++)enem[k2]=grad[k2]*(trans+vent)/1000.;
        sprintf(Text2,"Energianvändning år %d\n",i);
        EneBal(Text2);
        if(Vsys==6)Panntyp=6;
        EneKost=Taxor(Panntyp,BrPris,Fjv,Bytt);//Energikostnad för ett år
        EneKostNuv+=pow((1/R2),-(i+1))*EneKost/Vg;
        diff1=EneKost/Vg;
        sprintf(Text2,"%d = \t%6.1f \t%6.1f\n",i+1,EneKostNuv,diff1);
        strcat(text3,Text2);
    }
    if(Vsys==6)Panntyp=2;
    strcpy(text,text3);
    RestvPanna=(PannSum*(-Optb+Libp)/PannKost[Panntyp][3])*pow(R1,-Optb);
    RestvSlang=(Slang*(-Optb+PannKost[Panntyp][5])/PannKost[Panntyp][5])*pow(R1,-Optb);
    TotKost=SkalR+NuvIsoKost+EneKostNuv+NuvPanna+NuvSlang+Restv-RestvPanna-RestvSlang+AnslAvg+
RestvSkal;
    sprintf(text4,"TotKost= %8.1f",TotKost);
    MessageBox(MinRuta,text4,"TOTAL KOSTNAD",MB_OK|MB_ICONEXCLAMATION);
    sprintf(Text2,"%6.2f\t%6.2f\t%6.2f\t%6.2f\t%6.2f\t%6.2f\t%6.2f\t%6.2f\t%6.2f\t%6.2f\n",
SkalR,NuvIsoKost,EneKostNuv,NuvPanna,NuvSlang,
Restv,RestvPanna,RestvSlang,AnslAvg,RestvSkal);
    strcat(text,Text2);
    StigDC=GetDC(MinRuta);
    GetClientRect(MinRuta,&rect);
    InvalidateRect(MinRuta,&rect,TRUE);
    int res;
    res=DrawText(StigDC,text,-1,&rect,DT_LEFT | DT_WORDBREAK | DT_EXPANDTABS);
    //TextOut(StigDC,200,200,text3,strlen(text3));
    ReleaseDC(MinRuta,StigDC);
    rect.bottom=1100;
    return;
}

float Taxor(int Vsys,float BrPris[12],float Fjv[6],int Bytt)
{
    float EneKostAr=0,EffKost,DebEff,FastAvg;
    int i;
    for (i=0;i<12;i++)
    {
        EneKostAr+=(BrPris[i]*eneres[i]);
    }
    /* DETTA GÅLLER FÖR GÄVLE!!!!
if(Vsys==7&&Bytt==0)
{
        DebEff=(eneres[0]+eneres[2])/1422.;
        if(DebEff<=100.)Fjv[4]=714.;
        if(DebEff>100.&&DebEff<=500.)Fjv[4]=508.;
        if(DebEff>500.)Fjv[4]=408.;
    EffKost=DebEff*Fjv[4];
        if(DebEff<=100.)FastAvg=5578.;
        if(DebEff>100.&&DebEff<=500.)FastAvg=26217.;
        if(DebEff>500.)FastAvg=76420.;
        EneKostAr+=(EffKost+FastAvg);
    }
    if(Vsys==4&&Bytt==1)
{
        DebEff=(eneres[0]+eneres[2])/1422.;
        if(DebEff<=100.)Fjv[4]=679.;
        if(DebEff>100.&&DebEff<=500.)Fjv[4]=482.;
        if(DebEff>500.)Fjv[4]=387.;
    EffKost=DebEff*Fjv[4];
        if(DebEff<=100.)FastAvg=5300.;
        if(DebEff>100.&&DebEff<=500.)FastAvg=26217.;
        if(DebEff>500.)FastAvg=72600.;
        EneKostAr+=(EffKost+FastAvg);
    }*/
    //DETTA GÅLLER FÖR MALMÖ
    if(Vsys==6)
{
        DebEff=(eneres[0]+eneres[1])/1422.;
        if(DebEff<=800.)FastAvg=Fjv[2];
        if(DebEff>800.)FastAvg=Fjv[3];
    EffKost=DebEff*Fjv[4];
        EneKostAr+=EffKost*Fjv[5]+FastAvg;
}
}

```

```

    }
    return EneKostAr;
}

float SkalRot(float Area[8],float In[8],float NL[5],float L[5],float Optb,float R1)
{
    float SumbR=0.,S0,S1=0.,L1,L2;
    int i;
    for(i=0;i<=7;i++)
    {
        S0=Area[i]*In[i];
        if(i>4)
        {
            L1=L[4];
            L2=NL[4];
        }
        else
        {
            L1=L[i];
            L2=NL[i];
        }
        S1=Nuvarde(S0,L1,L2,Optb,R1);
        SumbR+=S1;
    }
    return SumbR;
}

float Nuvarde(float S0,float L1,float L2,float Opt,float R1)
{
    float S1=0.;
    if(S0<=0.05)return 0.;
    if(L2<=0.05)return 0.;
    if(L1>Opt)return 0.;
    do
    {
        S1=S1+S0*pow(R1,-L1);
        L1+=L2;
    }
    while(L1<Opt);
    S1=S1-(((L1-Opt)/L2)*S0)*pow(R1,-Opt);
    return S1;
}

void SkipLine(int to)
{
    int nr1=0;
    char varde;
    while(nr1<to)
    {
        varde=getc(fh);
        if(varde=='\n')nr1++;
    }
    return;
}

float ReadLine(int to)
{
    int i;
    for(i=0;i<to;i++)
    {
        sca+=fscanf(fh,"%f",&vekt[i]);
    }
    return(vekt[0]);
}

void GradTim(float Klim[12])
{
    int i;
    float tim;
    for (i=0;i<12;i++)
    {
        if(i==0||i==2||i==4||i==6||i==7||i==9||i==11)tim=31.*24.;
    }
}

```

```

        if(i==1)tim=((3.*28.+29.)*24.)/4.;
        if(i==3||i==5||i==8||i==10)tim=30.*24.;
        grad[i]=(tin-Klim[i])*tim;
    }
    return;
}

void EneBal(char Btyp[50])

{
    float totene=0.,solsum=0.,sola[12];
    char text1[100];
    unsigned int j,i,langd;
    int fo,res;

    strcpy(text,Btyp);
    StigDC=GetDC(MinRuta);
    for(i=0;i<12;i++)
    {
        sola[i]=0.;
        for(j=0;j<4;j++)
        {
            sola[i]+=Sol[j][i]*Area[j+4];
        }
        solsum+=sola[i];
        eneres[i]=enem[i]-Gratis[i]-sola[i];
        if(eneres[i]<0.1)eneres[i]=0.;
        eneres[i]+=Tv[i];
        totene+=eneres[i];
    }
    //sprintf(text1,"%d\t%7.1f\t%7.1f\t%7.1f\t%7.1f\n",i+1,enem[i],Tv[i],Gratis[i],sola[i],eneres[i]);
    sprintf(text1,"%d\t%7.1f\n",i+1,eneres[i]);
    strcat(text,text1);
}

fo=open("c:\\excel\\op\\claes.txt",O_APPEND | O_TEXT |O_WRONLY);
if(fo==-1)
{
    MessageBox(MinRuta,"Filen claes.txt kan ej skapas","!!!!OBS!!!!",MB_OK|MB_ICONEXCLAMATION);
    return;
}

langd=strlen(text);
res=write(fo,text,langd);
if(res==-1)
{
    MessageBox(MinRuta,"Kan ej skriva till filen claes.txt.","!!!!OBS!!!!",MB_OK|MB_ICONEXCLAMATION);
    return;
}

if(res!=langd)
{
    MessageBox(MinRuta,"Skriver inte rätt till filen claes.txt.","!!!!OBS!!!!",MB_OK|MB_ICONEXCLAMATION);
    return;
}

close(fo);
GetClientRect(MinRuta,&rect);
InvalidateRect(MinRuta,&rect,TRUE);
//BeginPaint(MinRuta,&ps);
DrawText(StigDC,text,-1,&rect,DT_LEFT | DT_WORDBREAK | DT_EXPANDTABS);
//EndPaint(MinRuta,&ps);
ReleaseDC(MinRuta,StigDC);
return;
}

void TMyWindow::ShowEditWin( PTWindowsObject AParent,LPSTR ATitle, LPSTR FileName)

{
    GetApplication()->MakeWindow( new TEdWindow( AParent, ATitle, FileName) );
}

void TEdWindow::CMEx(RTMessage)
{
    CloseWindow();
    SetFocus(MinRuta);
}

```



```
    }  
  
int PASCAL WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow)  
{  
    TStigApp StigApp("SYSTEM KAIZEN",hInstance,hPrevInstance,lpCmdLine,nCmdShow);  
    StigApp.Run();  
    return StigApp.Status;  
}
```


Bilaga D

RC-fil

```
//Här nedan redovisas den sk RC-filen som ingår i alla Windowsprogram
//Nedstående includefiler är enkla översättningar av Borland C++ motsvarigheter
//eller filer som inte förändrats alls. Detta för att erhålla menyer mm på svenska
```

```
#include <windows.h>
#include <owlrc.h>
#include "filedial.dlg"
#include "filmeny.rc"
#include "stigacc.rc"
#include "inputdia.dlg"
#include "stdwnds.dlg"
#define CM_ABOUT 101
#define IDM_CLAES 103
#define IDM_OPERA 109
#define IDM_SORAD 110
#define IDM_BIVAL 111
#define CM_FILER 102
#define CM_EXI 104
#define CM_INDATA 400
#define IDM_SPEC 105
#define CM_ANDRA 106
#define CM_KLART 107
#define CM_SKIPPA 108
#define CM_VIND 300
#define CM_GOLV 301
#define CM_YTTER 302
#define CM_INNER 303
#define CM_TATNING 304
#define CM_TVAG 306
#define CM_VENT 305
#define CM_TREG 307
#define CM_ENERGI 308
#define CM_GAS 309
#define CM_FRAN 310
#define ID_CLAESDIA 200
#define ID_ISO 201
#define ID_PANNA 202
#define ID_ISOEDIT 122
#define ID_ISOAAAR 123
#define CM_BEFINT 320
#define CM_NYOLJA 321
#define CM_ELVAERME 322
#define CM_FJAERRV 323
#define CM_VAERMEP 324
#define CM_NATURGAS 325
#define CM_DISDIFF 326
#define CM_ELDIFF 327
#define CM_BIVTYP1 328
#define CM_BIVTYP2 329
#define CM_SKIP 330
#define ID_PANNA1 210
#define ID_PANNAR 211
```

```

#define ID_PANNEF 212
#define ID_TIDPKT 215
#define ID_TID 216

#define ID_NORR 218
#define ID_OST 219
#define ID_SYD 220
#define ID_VAST 221
#define ID_WINAR 222
#define ID_FONSTER 224
#define CM_KLART1 225
#define CM_SKIPPA1 226

About MENU
BEGIN
POPUP "&Optioner"
BEGIN
MENUITEM "&Info", CM_ABOUT
MENUITEM "&Ändra filer", CM_FILER
        MENUITEM "&Kör OPERA", IDM_OPERA
        MENUITEM "Kör &SORWIN", IDM_SORAD
        MENUITEM "Kör &BIVWIN", IDM_BIVAL
        MENUITEM "Kör &Excel", IDM_CLAES
        MENUITEM "Kör E&get", IDM_SPEC
MENUITEM "&Exit", CM_EXI
        END
        POPUP "\&Help"
        BEGIN
                MENUITEM "Öppna &Hjälp", CM_INDATA
        END
END

AboutBox DIALOG 20,20,160,80
STYLE DS_MODALFRAME | WS_CAPTION | WS_SYSMENU
CAPTION "Info om System Kaizen"
BEGIN
CTEXT "Version 1.08" -1,0,12,160,8
CTEXT "Stiggan Gustafsson" -1,0,36,160,8
CTEXT "Mars 1994" -1,0,48,160,8
DEFPUSHBUTTON "OK" IDOK,64,60,32,14,WS_GROUP
END

1 ICON "c:\opera\opera.ico"

ID_CLAESDIA DIALOG 15, 23, 184, 95
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION | WS_SYSMENU
CAPTION "Egna förändringar"
BEGIN
PUSHBUTTON "Avbryt",CM_SKIPPA, 149, 12, 26, 17
PUSHBUTTON "OK",IDOK, 148, 39, 27, 17
        PUSHBUTTON "Färdigt",CM_KLART, 148, 64, 27, 17
LTEXT "Typ av förändring:", -1, 6, 7, 64, 8
CONTROL "Isolering vind",CM_VIND, "BUTTON", 9 | WS_CHILD | WS_VISIBLE| BS_AUTORADIOBUTTON, 7, 23, 61, 10
CONTROL "Isolering golv",CM_GOLV, "BUTTON", 9 | WS_CHILD | WS_VISIBLE| BS_AUTORADIOBUTTON, 7, 37, 61, 10
CONTROL "Isolering ytterv",CM_YTTER, "BUTTON", 9 | WS_CHILD | WS_VISIBLE| BS_AUTORADIOBUTTON, 7, 51, 61, 10
CONTROL "Isolering innerv",CM_INNER, "BUTTON", 9 | WS_CHILD | WS_VISIBLE| BS_AUTORADIOBUTTON, 7, 66, 61, 10
CONTROL "Tätning",CM_TATNING, "BUTTON", 9 | WS_CHILD | WS_VISIBLE| BS_AUTORADIOBUTTON, 7, 81, 61, 10
CONTROL "Fönster tvåglas",CM_TVAG, "BUTTON", 9 | WS_CHILD | WS_VISIBLE| BS_AUTORADIOBUTTON, 78, 7, 61, 10
CONTROL "Fönster treglas",CM_TREG, "BUTTON", 9 | WS_CHILD | WS_VISIBLE| BS_AUTORADIOBUTTON, 78, 22, 61, 10
CONTROL "Fönster energi",CM_ENERGI, "BUTTON", 9 | WS_CHILD | WS_VISIBLE| BS_AUTORADIOBUTTON, 78, 36, 61, 10
CONTROL "Fönster gasfylln",CM_GAS, "BUTTON", 9 | WS_CHILD | WS_VISIBLE| BS_AUTORADIOBUTTON, 78, 51, 61, 10
CONTROL "Ventilation",CM_VENT, "BUTTON", 9 | WS_CHILD | WS_VISIBLE| BS_AUTORADIOBUTTON, 78, 66, 61, 10
CONTROL "Frånluftvärmep",CM_FRAN, "BUTTON", 9 | WS_CHILD | WS_VISIBLE| BS_AUTORADIOBUTTON, 78, 81, 61, 10
END

ID_ISO DIALOG 18, 18, 117, 57
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION | WS_SYSMENU
CAPTION "Ändra isolering"
BEGIN
        EDITTEXT ID_ISOEDIT, 62, 20, 15,12
LTEXT "Ange mängd extra", -1, 12, 8, 57, 8
LTEXT "isolering i cm:", -1, 12, 21, 57, 8
CONTROL "OK",IDOK, "BUTTON", BS_PUSHBUTTON | WS_CHILD | WS_VISIBLE | WS_TABSTOP, 86, 38, 26, 12

```

```

CONTROL "Utföres år?", -1, "STATIC", SS_LEFT | WS_CHILD | WS_VISIBLE | WS_GROUP, 12, 40, 40, 8
EDITTEXT ID_ISOAAAR, 62, 37, 15, 12
CONTROL "Avbryt", IDCANCEL, "BUTTON", BS_PUSHBUTTON | WS_CHILD | WS_VISIBLE | WS_TABSTOP, 86, 20, 26, 12
END

ID_PANNA DIALOG 21, 21, 179, 110
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION | WS_SYSMENU
CAPTION "OLIKA VÄRMESYSTEM"
BEGIN
CONTROL "VÄLJ ETT VÄRMESYSTEM:", -1, "STATIC", SS_LEFT | WS_CHILD | WS_VISIBLE | WS_GROUP, 7, 8, 87, 8
CONTROL "Befintligt",CM_BEFINT, "BUTTON", 9 | WS_CHILD | WS_VISIBLE | BS_AUTORADIOBUTTON, 10, 26, 53, 10
CONTROL "Ny oljepanna",CM_NYOLJA, "BUTTON", 9 | WS_CHILD | WS_VISIBLE | BS_AUTORADIOBUTTON, 10, 42, 53, 10
CONTROL "Elvärme, fast",CM_ELVAERME, "BUTTON", 9 | WS_CHILD | WS_VISIBLE | BS_AUTORADIOBUTTON, 10, 58, 53, 10
CONTROL "Fjärrv, fast",CM_FJAERRV, "BUTTON", 9 | WS_CHILD | WS_VISIBLE | BS_AUTORADIOBUTTON, 10, 74, 53, 10
CONTROL "Värmepump",CM_VAERMEP, "BUTTON", 9 | WS_CHILD | WS_VISIBLE | BS_AUTORADIOBUTTON, 10, 90, 53, 10
CONTROL "Naturgas",CM_NATURGAS, "BUTTON", 9 | WS_CHILD | WS_VISIBLE | BS_AUTORADIOBUTTON, 78, 26, 50, 10
CONTROL "Fjärrv, diff",CM_DISDIFF, "BUTTON", 9 | WS_CHILD | WS_VISIBLE | BS_AUTORADIOBUTTON, 78, 42, 50, 10
CONTROL "Elvärme, diff",CM_ELDIFF, "BUTTON", 9 | WS_CHILD | WS_VISIBLE | BS_AUTORADIOBUTTON, 78, 58, 50, 10
CONTROL "Olja-Vp, typ1",CM_BIVTYP1, "BUTTON", 9 | WS_CHILD | WS_VISIBLE | BS_AUTORADIOBUTTON, 78, 74, 50, 10
CONTROL "Olja-Vp, typ2",CM_BIVTYP2, "BUTTON", 9 | WS_CHILD | WS_VISIBLE | BS_AUTORADIOBUTTON, 78, 90, 50, 10
CONTROL "Avbryt",CM_SKIP, "BUTTON", BS_PUSHBUTTON | WS_CHILD | WS_VISIBLE | WS_TABSTOP, 142, 40, 28, 17
CONTROL "Färdigt",IDOK, "BUTTON", BS_PUSHBUTTON | WS_CHILD | WS_VISIBLE | WS_TABSTOP, 142, 69, 27, 17
END

ID_PANNA1 DIALOG 18, 20, 152, 56
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION | WS_SYSMENU
CAPTION "NY VÄRMEANLÄGGNING"
BEGIN
CONTROL "Installeras år?", -1, "STATIC", SS_LEFT | WS_CHILD | WS_VISIBLE | WS_GROUP, 5, 33, 48, 8
EDITTEXT ID_PANNAR, 83, 30, 16, 12
CONTROL "Pannans effekt i kW?", -1, "STATIC", SS_LEFT | WS_CHILD | WS_VISIBLE | WS_GROUP, 5, 15, 68, 8
EDITTEXT ID_PANNEF, 83, 12, 16, 12
CONTROL "Avbryt", IDCANCEL, "BUTTON", BS_PUSHBUTTON | WS_CHILD | WS_VISIBLE | WS_TABSTOP, 120, 9, 24, 14
CONTROL "OK", IDOK, "BUTTON", BS_PUSHBUTTON | WS_CHILD | WS_VISIBLE | WS_TABSTOP, 120, 33, 24, 14
END

ID_TID DIALOG 23, 40, 142, 56
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION | WS_SYSMENU
CAPTION "ÖVRIGA ÅTGÄRDER"
BEGIN
CONTROL "Ange år för åtgärd:", -1, "STATIC", SS_LEFT | WS_CHILD | WS_VISIBLE | WS_GROUP, 6, 11, 64, 8
CONTROL "Avbryt", IDCANCEL, "BUTTON", BS_PUSHBUTTON | WS_CHILD | WS_VISIBLE | WS_TABSTOP, 104, 8, 24, 14
CONTROL "OK", IDOK, "BUTTON", BS_PUSHBUTTON | WS_CHILD | WS_VISIBLE | WS_TABSTOP, 104, 33, 24, 14
EDITTEXT ID_TIDPKT, 34, 30, 16, 12
END

ID_FONSTER DIALOG 22, 21, 142, 89
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION | WS_SYSMENU
CAPTION "BYTE AV FÖNSTER"
BEGIN
CONTROL "Norr", ID_NORR, "BUTTON", BS_AUTORADIOBUTTON | WS_CHILD | WS_VISIBLE | WS_TABSTOP, 10, 10, 28, 12
CONTROL "Öster", ID_OST, "BUTTON", BS_AUTORADIOBUTTON | WS_CHILD | WS_VISIBLE | WS_TABSTOP, 43, 10, 28, 12
CONTROL "Söder", ID_SYD, "BUTTON", BS_AUTORADIOBUTTON | WS_CHILD | WS_VISIBLE | WS_TABSTOP, 10, 25, 28, 12
CONTROL "Väster", ID_VAST, "BUTTON", BS_AUTORADIOBUTTON | WS_CHILD | WS_VISIBLE | WS_TABSTOP, 43, 25, 33, 12
CONTROL "Avbryt",CM_SKIPPA1, "BUTTON", BS_PUSHBUTTON | WS_CHILD | WS_VISIBLE | WS_TABSTOP, 100, 8, 28, 14
CONTROL "OK", IDOK, "BUTTON", BS_PUSHBUTTON | WS_CHILD | WS_VISIBLE | WS_TABSTOP, 100, 38, 27, 14
CONTROL "Färdigt",CM_KLART1, "BUTTON", BS_PUSHBUTTON | WS_CHILD | WS_VISIBLE | WS_TABSTOP, 100, 68, 27, 14
EDITTEXT ID_WINAR, 55, 70, 16, 12
CONTROL "Ange år för", -1, "STATIC", SS_LEFT | WS_CHILD | WS_VISIBLE | WS_GROUP, 8, 59, 36, 8
CONTROL "byte:", -1, "STATIC", SS_LEFT | WS_CHILD | WS_VISIBLE | WS_GROUP, 8, 72, 28, 8
END

```


Bilaga E

Does postpond retrofitting save money?

I det ursprungliga dokumentet fanns en uppsats med i form av ett manuskript.
När denna text skrivs är manuskriptet sedan länge publicerat, se Referens [3]

Litteraturförteckning

- [1] Gustafsson Stig-Inge. A Computer Model for Optimal Energy Retrofits in Multi-Family Buildings. The OPERA model. Technical report, Byggeforskningsrådet, Dokument D21, Stockholm, 1990.
- [2] Gustafsson Stig-Inge. The OPERA model. A computer model for optimal energy retrofits in multi-family buildings. The FORTRAN code. Rapport nr LiTH-IKP-R-613, 1990.
- [3] Gustafsson Stig-Inge. Does Postponed Retrofitting Save Money? *Heat Recovery Systems & CHP*, 15(5):469 – 472, 1995.