SECTION 4. Computer science, computer engineering and automation.

Shevtsov Alexandr Nikolayevich
candidate of technical Sciences,
President, Theoretical & Applied Science, LLP,
associate Professor of the Department «Applied mathematics»
Taraz State University named after M.Kh. Dulati,
Kazakhstan

ABOUT SOME ALGORITHMS, AUTOMATED DOCUMENT MANAGEMENT PROGRAMS

In the article there are considered some issues of programming on Delphi workflow automation.

Keyword: documents, automation.

Currently there are different workflow systems:ELMA, СЭД, Docsvision, 1С, Е1 Евфрат, Directum, ИНТАЛЕВ, Escom, Optima-WorkFlow, Гран-Док, Летограф, Documetr and many others [1-10].

Selecting one of these systems is rather complicated, because each of them is a document repository, and can be rated by different parameters [7, С.1]:
• card documents;
• security;
• storage and tracking multiple versions of documents;
• search function;
• automatic notification;
• routes of passage of documents and setting the tasks to people;
• user-friendly interface;
• scanning and recognition of documents processed.

An important aspect of the selection can be and the price system:

Table 1.
The cost of the systems of electronic document circulation

System name	The price in rubles, including VAT
Directum	22,800 rubles for installation of the module on the server and 165,300 rubles - automation 20 jobs. Note: for more system functions are implemented in separate modules, you have to pay separately.
Escom.doc	80,000 roubles. Note: The delivery includes a license to use the program on the server and up to 20 simultaneous connections.
Optima-WorkFlow	just 260,000 rubles for the basic version, which will run for 10 users Note: 60,000 rubles'll have to pay for automation of 10 additional jobs. As is the case with the previous system, some modules have to purchase separately, as they are not included in the basic set.
PayDox	90,000 rubles per basic version There is a reference document types, where you set up the possible types of cards. There are reserved fields that you can use at its discretion, and you can also add your own fields directly in the database, or use additional data sources, including direct access to other information systems.
«Гран-Док»	30,000 rubles for the basic version for 5 users. Note: 675,00 rubles Supplement for automation additional 15 jobs.
«ИНТАЛЕВ: Корпоративные документы и процессы»	45,000 rubles per basic version for 5 users. Note: 4,000 rubles will cost every additional workplace.
«ЕВФРАТ-Документооборот»	110,000 rubles Note: the complex on 20 users include license

	«EUPHRATES-document circulation Customer and EUPHRATES-document circulation Server».
«Летограф»	The cost of one license is 10995 rubles. Note: the License gives the right to use all the features of the platform. The license fee includes the cost of the license on the database that is installed on the server.

The average price of the system is 100 thousand rubles, or about $ 3 thousand, and the cheapest development - about 300-400 dollars.

Table 2.

The results of the testing of the systems of electronic document circulation

The functionality of the system	Directum	Escom.doc	Optima-WorkFlow	PayDox	Гран-Док	ИНТАЛЕВ: КДИП	ЕВФРАТ	Летограф
Possibility of card document	5	5	5	4	4	4	4	5
Security	4	5	5	4	5	5	4	5
The ability to store versions	5	4	5	5	4	5	нет	3
Searchable repository of documents	5	5	5	4	5	5	5	5
The possibility of notification during document processing	3	4	5	4	4	5	3	3
Routing capabilities and forming tasks users	5	4	5	4	1	4	3	4
Possibility of integration with e-mail	5	3	4	4	4	4	5	5
Archiving obsolete documents	4	3	4	4	1	3	4	2
The possibility of creating a distributed storage structure	5	нет	4	2	3	4	4	4
The convenience of the interface	5	3	5	3	5	5	5	3
The possibility of scanning and recognition of documents	5	3	3	2	3	3	5	3
GPA	**4,6**	**3,5**	**4,5**	**3,6**	**3,5**	**4,2**	**3,8**	**3,7**

One point - functionality is present in the minimum amount, 5 points - the function is implemented to the fullest.

Moreover, the continuous updating of the cards of documents, and certain types of documents the programs may not have absolutely.

Card - is a file with the settings, determining the source form, built-in labels, as well as changing information and data to be filled in each document individually.

Consider the process of creation of an automated document management system on the example of filling out the forms in the mail.

We need the original form:

Set the two existing size forms of C4 and C5, 20x30 and 16x23 respectively.

code: Delphi

```
procedure TForm1.FormCreate(Sender: TObject);
begin
h:=Printer.PageWidth;
w:=Printer.PageHeight;

bmp:=tbitmap.Create;
bmp.Width:=w;
bmp.Height:=h;
DragAcceptFiles(Handle,True);
Button1.Click;
end;

//*****************************************
case combobox1.ItemIndex of
0:begin
  ww:= trunc(w*21/21);
  hh:=trunc(h*30/30);
end;
1:begin
  ww:= trunc(w*16/21);
  hh:=trunc(h*23/30);
end;
end;
//*****************************************
```

Fill in the blank background information (details), as well as the necessary lines and signs.

code: Delphi

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
begin
bmp.Canvas.Brush.Color:=clwhite;
bmp.Canvas.FillRect(rect(0,0,bmp.Width,bmp.Height));

bmp.Canvas.Pen.Width:=8;

case combobox1.ItemIndex of
0:begin
py1:=0;
py2:=0;
px1:=0;
px2:=0;

py3:=0;
px3:=0;
 zz:=1;
end;
1:begin
py1:=3;
py2:=10;
px1:=20;
px2:=2;

py3:=4;
px3:=4;
zz:=0.9;
bmp.Canvas.Pen.Width:=8;
 bmp.Canvas.MoveTo(0,bmp.Height);
 bmp.Canvas.LineTo(20,bmp.Height);
  bmp.Canvas.MoveTo(bmp.Width-50,bmp.Height);
  bmp.Canvas.LineTo(bmp.Width,bmp.Height);
  bmp.Canvas.LineTo(bmp.Width,bmp.Height-50);

  bmp.Canvas.MoveTo(bmp.Width,50);
 bmp.Canvas.LineTo(bmp.Width,0);
end;

end;
 for I := 0 to 4 do
 begin
 bmp.Canvas.MoveTo(trunc(w*0/295),trunc(h*(py2+15+(15-py1)*i)/210));
 bmp.Canvas.LineTo(trunc(w*(140-px1)/295),trunc(h*(py2+15+(15-
```

```
py1)*i)/210));

 bmp.Canvas.MoveTo(trunc(w*(148+px1/2-px2)/295),trunc(h*(py2+127+(15-
py1)*i)/210));
 bmp.Canvas.LineTo(trunc(w*(295-px1/2-px2)/295),trunc(h*(py2+127+(15-
py1)*i)/210));

 end;

 bmp.Canvas.MoveTo(trunc(w*(140-px1-(12.5-
px3)*6)/295),trunc(h*(py2+15+(15-py1)*5+3-py3)/210));
 bmp.Canvas.LineTo(trunc(w*(140-px1)/295),trunc(h*(py2+15+(15-py1)*5+3-
py3)/210));

 bmp.Canvas.MoveTo(trunc(w*(148+px1/2-px2)/295),trunc(h*(py2+127+(15-
py1)*5+3-py3)/210));
 bmp.Canvas.LineTo(trunc(w*(148+(12.5-px3)*6+px1/2-
px2)/295),trunc(h*(py2+127+(15-py1)*5+3-py3)/210));

 for I := 0 to 6 do
 begin
 bmp.Canvas.MoveTo(trunc(w*(140-(12.5-px3)*i-
px1)/295),trunc(h*(py2+15+(15-py1)*4)/210));
 bmp.Canvas.LineTo(trunc(w*(140-(12.5-px3)*i-
px1)/295),trunc(h*(py2+15+(15-py1)*5+3-py3)/210));

 bmp.Canvas.MoveTo(trunc(w*(148+(12.5-px3)*i+px1/2-
px2)/295),trunc(h*(py2+127+(15-py1)*4)/210));
 bmp.Canvas.LineTo(trunc(w*(148+(12.5-px3)*i+px1/2-
px2)/295),trunc(h*(py2+127+(15-py1)*5+3-py3)/210));
end;

image1.Picture.Bitmap:=bmp;
end;

procedure TForm1.Button2Click(Sender: TObject);
var s0:string;
begin
bmp.Canvas.Font.Size:=trunc(110*zz);
  bmp.Canvas.Font.Style:=[fsbold];

 bmp.Canvas.TextOut(trunc(w*(148+px1/2-
```

```
px2+25)/295),trunc(h*(py2+127+(15-py1)*3-9)/210),s2+' '+Edit1.Text);
 bmp.Canvas.TextOut(trunc(w*(148+px1/2-
px2+25)/295),trunc(h*(py2+127+(15-py1)*4-9)/210),s1+' '+Edit2.Text);

image1.Picture.Bitmap:=bmp;
end;
```

Data about the client will receive from an external source (the questionnaire of the client, Microsoft Exel), highlight, and automatically put them on the card.

code: Delphi

```
procedure TForm1.WMDropFiles(var Msg: TMessage);
Var
   Filename: array[0..256] of char;
begin

  DragQueryFile(THandle(Msg.WParam),0,Filename,SizeOf(Filename));

label1.caption:=FileName;
label2.caption:=extractfilename(label1.caption);
BitBtn1.Click;
    DragFinish(THandle(Msg.WParam));
end;
```

Open access to the server Exel, for direct work with the document, getting data from certain cells and load it into the card.

code: Delphi

```
procedure TForm1.BitBtn1Click(Sender: TObject);
begin
button1.Click;
//begin
if p=1 then ExcelApp.Workbooks.Close;p:=1;
  // а потом и ворд
//ExcelApp.Quit;
//end;

try   // Create a Word Instance
ExcelApp := CreateOleObject('Excel.Application');
except
   // Error...
```

```
   Exit;
end;
ExcelApp.Workbooks.Open(label1.Caption,0,readOnly:=false);//<имя
файла>,0,True);
ExcelApp.Visible := false;
ExcelApp.DisplayAlerts := False;

  bmp.Canvas.Font.Size:=trunc(80*zz);
  bmp.Canvas.Font.Style:=[fsItalic];


  bmp.Canvas.TextOut(trunc(w*0/295),trunc(h*(py2+15+(15-py1)*0-
4)/210),'От кого  ');
  bmp.Canvas.TextOut(trunc(w*0/295),trunc(h*(py2+15+(15-py1)*2-
4)/210),'Откуда  ');
  bmp.Canvas.Font.Size:=trunc(60*zz);
  bmp.Canvas.TextOut(trunc(w*(140-px1-(12.5-
px3)*6)/295),trunc(h*(py2+15+(15-py1)*4-5)/210),'Индекс  ');

  bmp.Canvas.Font.Size:=trunc(80*zz);
  bmp.Canvas.TextOut(trunc(w*(148+px1/2-px2)/295),trunc(h*(py2+127+(15-
py1)*0-4)/210),'Кому  ');
  bmp.Canvas.TextOut(trunc(w*(148+px1/2-px2)/295),trunc(h*(py2+127+(15-
py1)*2-4)/210),'Куда  ');

  bmp.Canvas.Font.Size:=trunc(60*zz);
  bmp.Canvas.TextOut(trunc(w*(148+px1/2-px2)/295),trunc(h*(py2+127+(15-
py1)*4-5)/210),'Индекс ');


  bmp.Canvas.Font.Size:=trunc(110*zz);
  bmp.Canvas.Font.Style:=[fsbold];

  bmp.Canvas.TextOut(trunc(w*25/295),trunc(h*(py2+15+(15-py1)*0-
9)/210),'Шевцов А.Н.');
  bmp.Canvas.TextOut(trunc(w*25/295),trunc(h*(py2+15+(15-py1)*2-
9)/210),'ул.Джамбулская 128');
  bmp.Canvas.TextOut(trunc(w*25/295),trunc(h*(py2+15+(15-py1)*3-
9)/210),'г.Тараз, Жамбылская обл.');
  bmp.Canvas.TextOut(trunc(w*25/295),trunc(h*(py2+15+(15-py1)*4-9)/210),'
Казахстан');
  bmp.Canvas.Font.Style:=[];
```

```
  bmp.Canvas.Font.Size:=trunc(200*zz*zz*zz);
s:='080014'  ;

 for I := 1 to 6 do
 begin
 bmp.Canvas.TextOut(trunc(w*(140-px1-(12.5-px3)*(i)+3-
px3/5)/295),trunc(h*(py2+15+(15-py1)*4+1-py3/5)/210),copy(s,7-i,1));
 end;

 s1:=ExcelApp.Range['C9'];
 s2:=ExcelApp.Range['D9'];
 s3:=ExcelApp.Range['E9'];


  bmp.Canvas.Font.Size:=trunc(110*zz);
  bmp.Canvas.Font.Style:=[fsbold];

  bmp.Canvas.TextOut(trunc(w*(148+px1/2-
px2+25)/295),trunc(h*(py2+127+(15-py1)*0-9)/210),s1);
  bmp.Canvas.TextOut(trunc(w*(148+px1/2-
px2+25)/295),trunc(h*(py2+127+(15-py1)*1-9)/210),s2+' '+s3);

 s:=ExcelApp.Range['B17'];
 s1:=ExcelApp.Range['C17'];
 s2:=ExcelApp.Range['D17'];
 s3:=ExcelApp.Range['E17'];
 s4:=ExcelApp.Range['F17'];
 s5:=ExcelApp.Range['G17'];

s0:='   ;   if checkbox1.Checked then  s0:='ул. '    ;
if s4<>'' then  d0:='д.' else d0:='   ;

if s5<>'' then
bmp.Canvas.TextOut(trunc(w*(148+px1/2-
px2+25)/295),trunc(h*(py2+127+(15-py1)*2-9)/210)-20,s0+s3+'         '+d0+s4+'
кв.'+s5) else
bmp.Canvas.TextOut(trunc(w*(148+px1/2-
px2+25)/295),trunc(h*(py2+127+(15-py1)*2-9)/210),s0+s3+'  '+d0+s4);

 bmp.Canvas.TextOut(trunc(w*(148+px1/2-px2+25-
strtoint(edit3.Text))/295),trunc(h*(py2+127+(15-py1)*3-9)/210),s2);
 bmp.Canvas.TextOut(trunc(w*(148+px1/2-
```

```
px2+25)/295),trunc(h*(py2+127+(15-py1)*4-9)/210),s1);
//
//s:='080014'
  bmp.Canvas.Font.Style:=[];
  bmp.Canvas.Font.Size:=trunc(200*zz*zz*zz);

 for I := 1 to 6 do
 begin
 bmp.Canvas.TextOut(trunc(w*(148+(12.5-px3)*(i-1)+3+px1/2-px2-
px3/5)/295),trunc(h*(py2+127+(15-py1)*4+1-py3/5)/210),copy(s,i,1));
end;

image1.Picture.Bitmap:=bmp;
      Edit1.Text:='';
      Edit2.Text:='';

end;
```

Built-in function print a document converts card in the DIB format and sends it directly to the printer with the size of the card.

code: Delphi

```
procedure TForm1.BitBtn2Click(Sender: TObject);
var ww,hh:integer;
begin
case combobox1.ItemIndex of
0:begin
  ww:= trunc(w*21/21);
  hh:=trunc(h*30/30);
end;
1:begin
  ww:= trunc(w*16/21);
  hh:=trunc(h*23/30);
end;
end;

Printer.Orientation:=poLandscape;
Printer.BeginDoc;
BltTBitmapAsDib(Printer.Canvas.Handle,0,0,ww,hh,
//Printer.PageWidth,Printer.PageHeight,
Bmp);
```

```
Printer.EndDoc;

end;
```

The process of conversion is specified algorithm:

code: Delphi

```
procedure BltTBitmapAsDib(DestDc : hdc;   {Handle of where to blt}
                  x : word;       {Bit at x}
                  y : word;       {Blt at y}
                  Width : word;   {Width to stretch}
                  Height : word;  {Height to stretch}
                  bm : TBitmap);  {the TBitmap to Blt}
var
OriginalWidth :LongInt;             {width of BM}
dc : hdc;                       {screen dc}
IsPaletteDevice : bool;             {if the device uses palettes}
IsDestPaletteDevice : bool;         {if the device uses palettes}
BitmapInfoSize : integer;           {sizeof the bitmapinfoheader}
lpBitmapInfo : PBitmapInfo;         {the bitmap info header}
hBm : hBitmap;                      {handle to the bitmap}
hPal : hPalette;                    {handle to the palette}
OldPal : hPalette;                  {temp palette}
hBits : THandle;                    {handle to the DIB bits}
pBits : pointer;                    {pointer to the DIB bits}
lPPalEntriesArray : PPalEntriesArray; {palette entry array}
NumPalEntries : integer;            {number of palette entries}
i : integer;                    {looping variable}
begin
{$IFOPT R+}
{$DEFINE CKRANGE}
{$R-}
{$ENDIF}
OriginalWidth := bm.Width;dc := GetDc(0);IsPaletteDevice :=
GetDeviceCaps(dc, RASTERCAPS) and RC_PALETTE = RC_PALETTE;dc :=
ReleaseDc(0, dc);
if IsPaletteDevice then    BitmapInfoSize := sizeof(TBitmapInfo) +
(sizeof(TRGBQUAD) * 255) else    BitmapInfoSize := sizeof(TBitmapInfo);
GetMem(lpBitmapInfo, BitmapInfoSize);FillChar(lpBitmapInfo^,
BitmapInfoSize, #0);
lpBitmapInfo^.bmiHeader.biSize :=
sizeof(TBitmapInfoHeader);lpBitmapInfo^.bmiHeader.biWidth :=
OriginalWidth;
```

```
lpBitmapInfo^.bmiHeader.biHeight :=
bm.Height;lpBitmapInfo^.bmiHeader.biPlanes := 1;
if IsPaletteDevice then    lpBitmapInfo^.bmiHeader.biBitCount := 8
else    lpBitmapInfo^.bmiHeader.biBitCount := 24;
lpBitmapInfo^.bmiHeader.biCompression :=
BI_RGB;lpBitmapInfo^.bmiHeader.biSizeImage :=
((lpBitmapInfo^.bmiHeader.biWidth *
longint(lpBitmapInfo^.bmiHeader.biBitCount)) div 8) *
lpBitmapInfo^.bmiHeader.biHeight;
lpBitmapInfo^.bmiHeader.biXPelsPerMeter :=
0;lpBitmapInfo^.bmiHeader.biYPelsPerMeter := 0;
if IsPaletteDevice then begin
  lpBitmapInfo^.bmiHeader.biClrUsed := 256;
lpBitmapInfo^.bmiHeader.biClrImportant := 256;
end else begin                    lpBitmapInfo^.bmiHeader.biClrUsed := 0;
lpBitmapInfo^.bmiHeader.biClrImportant := 0;end;
hBm := bm.ReleaseHandle;hPal := bm.ReleasePalette;
dc := GetDc(0);if IsPaletteDevice then begin
  OldPal := SelectPalette(dc, hPal, TRUE);   RealizePalette(dc);
end;
GetDiBits(dc,        hBm,        0,        lpBitmapInfo^.bmiHeader.biHeight,
nil,        TBitmapInfo(lpBitmapInfo^),        DIB_RGB_COLORS);
hBits := GlobalAlloc(GMEM_MOVEABLE,
lpBitmapInfo^.bmiHeader.biSizeImage);
pBits := GlobalLock(hBits);
GetDiBits(dc,        hBm,        0,        lpBitmapInfo^.bmiHeader.biHeight,
pBits,        TBitmapInfo(lpBitmapInfo^),        DIB_RGB_COLORS);
if IsPaletteDevice then begin    GetMem(lPPalEntriesArray,
sizeof(TPaletteEntry) * 256);
 {$IFDEF VER100}
   NumPalEntries := GetPaletteEntries(hPal,                    0,
256,                    lPPalEntriesArray^);
 {$ELSE}
   NumPalEntries := GetSystemPaletteEntries(dc,
0,                    256,
lPPalEntriesArray^);
 {$ENDIF}
  for i := 0 to (NumPalEntries - 1) do begin
   lpBitmapInfo^.bmiColors[i].rgbRed :=      lPPalEntriesArray^[i].peRed;
   lpBitmapInfo^.bmiColors[i].rgbGreen :=      lPPalEntriesArray^[i].peGreen;
   lpBitmapInfo^.bmiColors[i].rgbBlue :=      lPPalEntriesArray^[i].peBlue;
  end;
```

```
  FreeMem(lPPalEntriesArray, sizeof(TPaletteEntry) * 256);
end;
if IsPaletteDevice then begin
   SelectPalette(dc, OldPal, TRUE);
   RealizePalette(dc);
end;
dc := ReleaseDc(0, dc);
IsDestPaletteDevice :=   GetDeviceCaps(DestDc, RASTERCAPS) and
RC_PALETTE = RC_PALETTE;
if IsPaletteDevice then begin
   OldPal := SelectPalette(DestDc, hPal, TRUE);   RealizePalette(DestDc);
end;
StretchDiBits(DestDc,          x,          y,          Width,          Height,
0,          0,          OriginalWidth,
lpBitmapInfo^.bmiHeader.biHeight,          pBits,          lpBitmapInfo^,
DIB_RGB_COLORS,          SrcCopy);
if IsDestPaletteDevice then begin    SelectPalette(DestDc, OldPal, TRUE);
RealizePalette(DestDc);end;
GlobalUnLock(hBits);GlobalFree(hBits);FreeMem(lpBitmapInfo,
BitmapInfoSize);
bm.Handle := hBm;
bm.Palette := hPal;
{$IFDEF CKRANGE}
{$UNDEF CKRANGE}
{$R+}
{$ENDIF}
end;
```

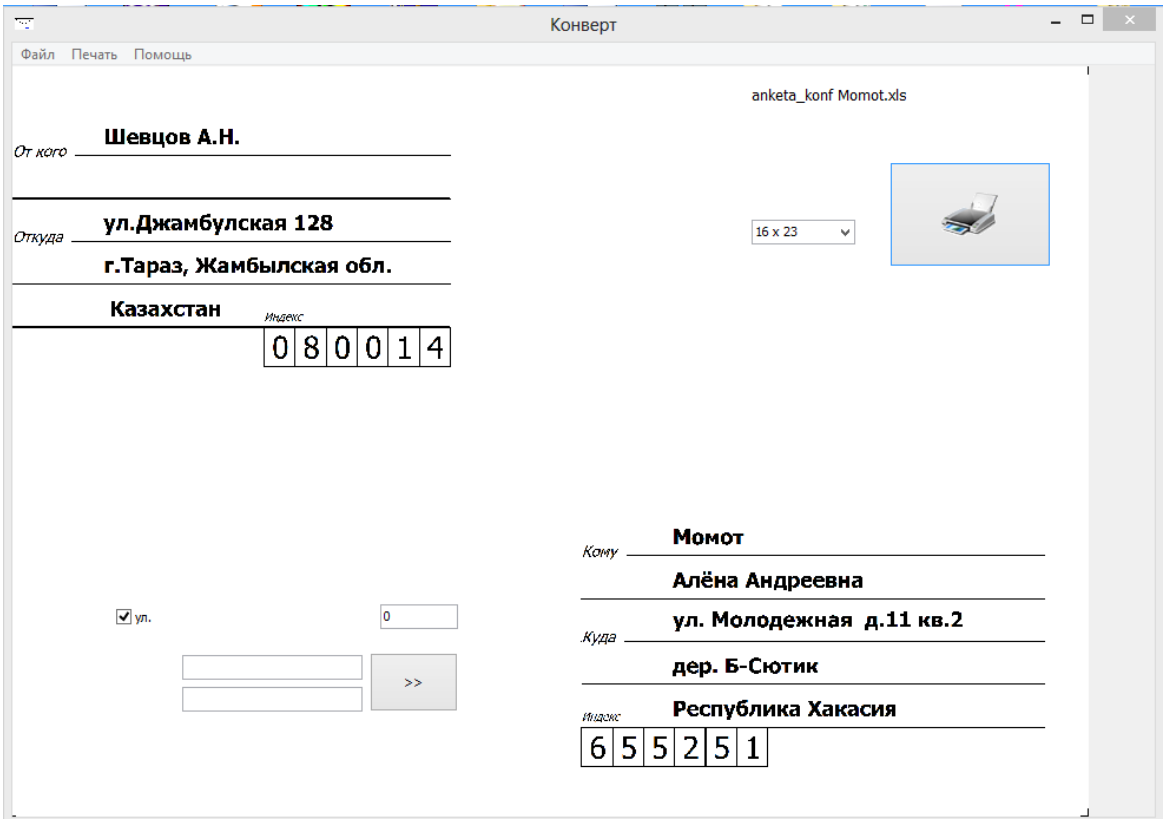

Figure 1 - The Automated system creation blanks.

The obtained system, and algorithms demonstrate an example of creating an automated system to create documents on the basis of initial and current data as well as certain design standards (Fig.1), and databases created in different formats. The system can be implemented in various organizations faced with the necessity of sending letters by mail.

Bibliography

1. ELMA: Электронный документооборот. [Электронный ресурс]. URL : http://www.elma-bpm.ru/product/e-doc/ (дата обращения: 21.07.2013).
2. Система электронного документооборота. [Электронный ресурс]. URL: http://www.prostoy.ru/145.html (дата обращения: 21.07.2013).
3. Семейство программных продуктов «Канцлер» для созданияСЭД. [Электронный ресурс]. URL : http://www.kancler.by/ (дата обращения: 21.07.2013).
4. Электронный документооборот. [Электронный ресурс]. URL : http://www.docsvision.com/ (дата обращения: 21.07.2013).
5. Система электронного документооборота (СЭД): решаемые задачи. [Электронный ресурс]. URL: http://elektronnij-dokumentooborot.1c-kpd.ru/ (дата обращения: 21.07.2013).

6. Система электронногодокументооборота и автоматизации бизнес-процессов. [Электронный ресурс]. URL : http://www.evfrat.ru/ (дата обращения: 21.07.2013).
7. Маруняк А. Какую программу выбрать для автоматизации документооборота. Тест-драйв популярных систем. [Электронный ресурс]. URL: http://www.intalev.ru/library/articles/article.php?ID=5211 (дата обращения: 21.07.2013).
8. 1С для автоматизации документооборота и бизнес-процессов. [Электронный ресурс]. URL: http://документооборот.net/ (дата обращения: 21.07.2013).
9. Автоматизация электронного документооборота (СЭД) Автоматизированная система делопроизводства и электронного документооборота. [Электронный ресурс]. URL : http://www.motiw.ru/docflow/ (дата обращения: 21.07.2013).
10. Программа управления электронными документами. [Электронный ресурс]. URL: http://www.documetr.ru/anot.html (дата обращения: 21.07.2013).