

Rīgas Tehniskā universitāte

Datorzinātnes un informācijas tehnoloģijas fakultāte

Informātikas un programmēšanas katedra

Risinājumu algoritmizācija un programmēšana (2.daļa)

Studiju darbs

Numuru uzskaites sistēma viesnīcai

**Datorzinātnes un Informācijas
tehnoloģijas fakultāte
IDBD 1. kurss 14. grupa
Sergejs Terentjevs
Studenta apl. 061RDB140**

Anotācija

Tiek piedāvāta datorizēta viesnīcas numuru uzskaites sistēma jeb datu bāze, kurā ir iespējams apkopot un aplūkot informāciju par viesnīcas numuriem, istabu skaitu numurā, koriģēt un dzēst datus, veikt informācijas kārtošanu, kā arī interesējošo datu atlasīšanu un ierakstu izslēgšanu, u.t.t.

Ar programmas palīdzību ir iespējams manipulēt ar datiem, kurus lietotājs ievada failā, kā arī vēlāk realizēt visas iepriekš minētas darbības.

Programma tika uzrakstīta Borland Turbo Pascal vidē. Programmas mērķis ir veikt vienkāršu un saprotamu saskarsmi starp lietotāju un programmu, kā arī realizēt visas darbības un prasības, lai šo programmu varētu tiešam uzskatīt par viesnīcas numuru uzskaites sistēmu.

Saturs

1.	Darba uzdevums	4
2.	Pasūtītāja prasības	4
3.	Realizācijas projekts.	5
3.1	Apraksts	5
3.2	Programmas uzbūve	5
4.	Programmatūras specifikācija	6
4.1	Interfeisa specifikācija	6
4.2	Moduļa st01 specifikācija	6
5.	Testēšanas metodika	8
6.	Programmas pirmtteksti	9
7.	Instrukcija lietotājam	36
8.	Secinājumi	37

1 Darba uzdevums

Izveidot viesnīcas numuru uzskaites sistēmu, kura tieši strādās ar failu, failam ir jāsaturs informāciju par numuru, istabu skaitu numura, aizņemts vai brīvs numurs, viesu vārdu un uzvārdu, cena par diennakti.

Programmā ir jāparedz faila satura apskati, jaunas informācijas ievadi, ierakstu dzēšanu, ierakstu koriģēšanu, informācijas kārtošanu pēc viesu uzvārdiem vai istabu skaita numurā, brīvo istabu un visdārgāko numuru atrašana.

2 Pasūtītāja prasības

Viesnīcas numuru uzskaites sistēma paredz iespēju reģistrēt līdz pat 100 numuriem, katrs no šiem numuriem glabā unikālo informāciju, kura sniedz aprakstu lietotājam par katru viņam interesējošo numuru.

Datu bāze satur sekojošus datus par katru no pierēģistrētājiem numuriem :
istabu skaits numura, viesu vārds un uzvārds, cena par diennakti, numura numurs, numura stāvoklis (brīvs vai aizņemts).

Numuru uzskaites sistēmas īpašnieks skaitās tas lietotājs, kuram ir tieša pieeja pie datu bāzes, citām personām aplūkot šo programmu neizdosies, jo tā ir aizsargāta ar paroli.

Viesnīcas numuru uzskaites sistēmas īpašnieks var sākt darbu ar tukšu datu bāzi vai arī strādāt ar jau izmantotu datu bāzi, dzēšot, koriģējot informāciju no faila.

Numuru uzskaites sistēmas īpašniekam ir dota iespēja atrast visas brīvas istabas un visdārgāko numuru, kas, savukārt, lietotājam atvieglo un padara ērtāko programmas lietošanu..

Ka arī lietotājs var sakārtot informāciju failā pēc viesu uzvārdiem un istabu skaita numurā, kas ļauj konkrēti un precīzi atlasīt datus par lietotāja interesējošo numuru.

3 Realizācijas projekts.

3.1 Apraksts

Veidosim failu apkalpošanas sistēmu.

Paredzēsim visu datu glabāšanu vienā failā.

Programmas darbība visu laiku notiks ar tiešās pieejas failu, tas nozīmēs, ka pēc katras manipulācijas ar datiem, veiktās izmaiņas tiek saglabātas teksta failā

Veicot mums interesējošo datu atlasī, programma veidos divus vēl papildus teksta failus, kuri glabās informāciju par mūsu atlasītiem datiem

Lai lietotājam būtu ērtāk rediģēt ieraksta saturu, katram ierakstam programmas izpildes laikā tiek piešķirts kārtas numurs, atteicīgi lietotājs vāres izvēleties noteiktu numuru no saraksta.

Viesnīcas numuru uzskaites sistēmas īpašniekam tiks dotas visas tiesības: jaunu datu ievade, informācijas labošana, datu atlasīšana, ierakstu dzēšana, datu bāzes aplūkošana, informācijas kārtošana.

Programma saturēs procedūru, kura ļaus lietotājam izdzēst nepareizi ievadīto burtu, ciparu, pavisamoties pa kreisi, vai pa labi, kontroles lietotāja vadītos datus (nepieļaus nekorektos ierakstus), veidos lietotājam tīkamo un vieglu interfeisu, kas padarīs programmas lietošanu vieglu u.tml.

Datiem uzskaites sistēma būs sekojoša struktūra:

Viesa vārds - 12 baiti (string tipa mainīgais },
Viesa uzvārds - 12 baiti (string tipa mainīgais),
Numura nr. – integer tipa mainīgais,
Istabu sk.- integer tipa mainīgais,
Cena – real tipa mainīgais.
Stāvoklis – string tipa mainīgais

3.2 Programmas uzbūve

Sistēmas pamats ir:

1. Tiešās pieejas fails „hotel” – numuru uzskaites sistēma,
2. Modulis „st01” - satur apakšprogrammas darbam ar šāda veida failu,
3. Modulis „st02” – lietotāja interfeiss, kas padara programmu ērtu lietotājam un veic brīdinājumu izvadi,
4. Galvenā programma „Studio” - veic lietotāja autentifikāciju un moduļu testēšanu un pielietošanu.
5. Papildus faili „holdtext” un „roomdata” - satur informāciju par atlasītiem datiem (tiek veidoti un pārrakstīti bez lietotāja starpības) .

Tiešās pieejas fails satur šāda tipa ierakstus:

```
type
  Data = array [1..Max] of ^Outlaw;
  Outlaw = record
```

```

Vards      : string[12];
Uzvards    : string[12];
Numurs     : integer;
Istabas    : integer;
Stavoklis  : string[12];
Cena       : real;
end;

```

Paredzēsim modulī šādas failu apkalpošanas funkcijas

1. Jauna datu faila izveidošana – parametri: faila identifikators;
2. Datu bāzes jeb faila apskatīšanu – parametri: faila identifikators, faila vārds;
3. jauno ierakstu pievienošanu – parametri: faila identifikators, faila vārds;
4. Ierakstu izslēgšanu – parametri: faila identifikators, faila vārds;
5. Ierakstu izslēgšanu – parametri: faila identifikators, faila vārds;
6. Faila sakārtošanu pēc pazīmes – parametri: faila identifikators, faila vārds;
7. Datu meklēšanu un atlasīšanu: faila identifikators, faila vārds;

4 Programmatūras specifikācija

4.1 Interfeisa specifikācija

Dialogs ar lietotāju tiek realizēts, izmantojot moduļa CRT iebūvētas pamatfunkcijas un procedūras, piemēram, `textbackground (white)` vai `textcolor(blue)`. Programma veic kontroli, kas izpaužas, ka lietotāja datu ievades pārbaude, piemēram, ja lietotājs laukā, kura ir jāievada string tipa simbolu virkne, mēģina ievadīt skaitli, programma neļauj veikt šādu darbību. Tas tiek panākts ar procedūras `InputCheck` palīdzību.

Lai realizētu lietotājam tīkamo programmas saskarsmi, programma satur izvēlni, pa kuru var brīvi pārvietoties, ka arī lietotājs var izvēlēties sev vēlamu operāciju. Jā lietotājs ievada nepareizo informāciju vai fails ir tukšs, vai arī neatvērts programma izvada attiecīgos brīdinājuma paziņojumus. Jebkurā mirklī lietotājs var iziet no izvēlētas darbības nospiežot taustiņu „Esc”.

4.2 Moduļa `st01` specifikācija

```

unit St01;
INTERFACE
uses crt, st02;
Type
  MenuType = (Vertical);
Const
  width = 50;
  Items1 = 8;

  optText1: array [0..Items1-1] of string =
    ('.[ apskatīt datu bāzes saturu ]:.',
     ':[ ierakstīt jaunu informāciju ]:.',
     ':[ dzēst ierakstu ]:.',
     ':[ korigēt informāciju ]:.',
     ':[ sakārtot informāciju faila ]:.',
     ':[ atrast visas brīvas istabas ]:.',
     ':[ atrast informāciju par visdārgāko numuru ]:.',

```

```

' : [ exit ] : .');

optNormal = 1;
optSelected = white;
Const
    Max = 100;
    filename = 'hotel.txt';
type
    Data = array [1..Max] of ^Outlaw;
    Outlaw = record
        Vards    : string[12];
        Uzvards  : string[12];
        Numurs   : integer;
        Istabas  : integer;
        Stavoklis : string[12];
        Cena     : real;
    end;
    MyFile = File of Outlaw;

```

Modulis „st01” satur šādas procedūras darbam ar teksta tipa failu, aplūkosim tās:

1. Procedūra faila stavokļa noteikšanai.

```
Procedure position (var textfile : MyFile);
```

Procedūra nosaka faila stāvokli, vai tas ir tukšs, vai tas ir atrasts un atvērts. Pēc procedūras izpildes mēs iegūstam turpmāko rīcības plānu.

2. Procedūra tekst tipa faila apskatīšanai.

```
procedure content (var textfile: myfile; filename: string);
```

Procedūra satur algoritmus, kuri ļauj lietotājam veikt pilna faila apskati.

3. Procedūra jaunu ierakstu ievadīšanai un saglabāšanai failā.

```
Procedure addrecords (var textfile: MyFile; filename: string);
```

Procedūra satur visu nepieciešamu, lai lietotājs spētu veigti un korekti ievadīt un saglabāt jaunus datus.

4. Procedūra ierakstu izslēgšanai.

```
procedure discharge (var textfile: myfile; filename: string);
```

Procedūra satur algoritmus, kuri ļauj dzēst izvēlto ierakstu.

5. Procedūra esošo ierakstu labošanai.

```
procedure correct (var textfile: myfile; filename: string);
```

Procedūra „correct” nodrošina ierakstu labošanu, izpildes sākumā lietotājam būs jāizvēlas ieraksts, turpmāk tiks piedāvātas izvēles, kuras ļaus lietotājam labot vārdu, bet ja viņš nevēlās

labot uzvārdu, nepiekrītot nosacījumam, procedūra paries uz nākošo ierakstu, bet uzvārds netiks labots.

6. Procedūra ierakstu kārtšanai;

```
procedure Sort (var textfile: myfile; filename: string);
```

Izsaucot šo procedūru, lietotājam būs jāizvēlas, ka tieši viņš grib kārtot failu – vai nu pēc viesu uzvārdiem, vai arī pēc istabu skaita numurā. Sekojoši pēc izpildes iegūsim failu, kura ieraksti tiks sakārtoti atbilstoši kārtšanas kritērijam.

7. Procedūra brīva numura atrašanai esoša ieraksta;

```
Procedure search (var textfile : myfile; filename : string);
```

Procedūra vispirms atrod visus brīvos numurus, tad veic visu brīvo istabu saskaitīšanu. Atlasītos datus lietotājam ir iespējams aplūkot cita teksta tipa failā.

8. Procedūra visdargāka numura atrašanai;

```
Procedure expensive (var textfile : myfile; filename : string);
```

Procedūra „expensive” veic viena vai vairāku dargo numuru un informāciju par šiem numuriem atrašanu. Dati tiek atlasīti cita teksta tipa failā.

Bez tām modulis satur vairākas papildfunkcijas, ka piemēram izvēlnes veidošanai u.t.t.

5 Testēšanas metodika

1. Jaunas viesnīcas numuru uzskaites sistēmas veidošana;
2. Numuru uzskaites sistēmas papildināšana ar jauniem ierakstiem;
3. Izvēlēto ierakstu koriģēšana (pēc lietotāja izvēles tiek mainīts viens ieraksts, piemēram, vārds, bet pārējie ieraksti paliek neskarti (uzvārds, numurs, istabas skaits numurā, u.tml.);
4. Izvēlēto ierakstu dzēšana, ka arī datu atlasīšana un nepieciešamas informācijas sameklēšana;
5. Procedūru testēšana strādājot ar failu, kas satur tikai dažus ierakstus;
6. Procedūru testēšana strādājot ar pilnīgu datu bāzi;
7. Programmas testēšana (brīdinājumu izvades pārbaude), ja lietotājs izdara, ievada nepareizo darbību un informāciju;
8. Programmas testēšana, ja fails netiek atvērts, vai arī nesatur ierakstus;
9. Programmas testēšana, ja lietotājs ievada nepareizo paroli, vai mēģina pārtraukt kādas procedūras darbību, nospiežot taustiņu „Esc” (programma atgriežas pie galvenās izvēlnes).

6 Programmas pirmteksti

Galvenās programmas pirmteksts

```

Program butterfly;

uses crt, St01, St02;

Label Login;
var i,x,sk,k:byte;
    simb: byte;
    pass: string;

begin
  Login:
    TextBackground (White);
    Clrscr;
    TextColor(Black);
    writeln('');
    writeln('      /_____)')';
    writeln('      (____|" -')';
    writeln('      ; _='')';
    writeln('      // _ /')';
    writeln('      /) \ / ) ) |. _____ |')';
    writeln('      // | - - / \ / ; |. : |')';
    writeln('      | / | / \ / |. | /')';
    writeln('      ; ::::: | _____ |')';
    writeln('      (/ // // // \ \ \ \ \ | _ | _ |')';
    writeln('      _ / | _ // // // // / / _ [ _____ ] _')';
    writeln('      | / | /')';
    writeln('      | |')';
    writeln('      ( | ( |')';
    writeln('      , | | | |')';
    writeln('      ; = ; =')';
    gotoXY(21,15);
    showcursor;
    TextColor (White);
    Textbackground(6);
    writeln ('.: [ Numuru uzskaites sistema veisnicai ] :. ');
    Textbackground (White);
    gotoXY(30,19);
    TextColor (white);
    writeln ('Ievadiet Jusu paroli');
    gotoXY(30,20);
    TextColor(white);
    Writeln ('password:');
    gotoXY (40,20);
    TextBackground (6); Write (''); ;
    GotoXY (40,20);
    TextColor (Black);
    Textbackground(white);

    pass:='';

```

```

x:=24;
sk:=0;
K:= 9;

repeat
    simb:=ord(readkey);
    if not (ord(simb) = 13) and (ord(simb) <>10) then
        begin
            If ord(simb) = 27 then Exit;
            if (simb <> 8) and (Length(pass) <=9) then begin
pass:=pass + chr(simb); inc (sk); end;
            If (simb = 8) and (Length(pass) > 0) then
                begin
                    dec (sk);
                    delete (pass,Length(pass),1);
                end;

        Clrscr;

        TextBackground (White);
        Clrscr;
        TextColor (Black);
        writeln('');
        writeln('          /_____)'))';
        writeln('          (____|" - ');
        writeln('          ; _ = ');
        writeln('          //_____/ ');
        writeln('          /)_____\ /)_____) |._____| ');
        writeln('          //| - -/\|/; |. :| ');
        writeln('          | / | / \| | / ');
        writeln('          ; : : : : |_____| ');
        writeln('          (/ // // // \ \ \ \ \ |_____| ');
        writeln('          ____/|_// // // // / / ____ [_____] ');
        writeln('          | / | / ');
        writeln('          | | ');
        writeln('          (| (| ');
        writeln('          ,| | | ');
        writeln('          ; = ; = ');
        gotoXY(21,15);
        TextColor (White);
        Textbackground (6);
        writeln ('.: [ Numuru uzskaites sistema veisnicai ]:.' );
        Textbackground (white);
        gotoXY(30,19);
        TextColor (white);
        writeln ('Ievadiet Jusu paroli');
        gotoXY(30,20);
        TextColor (white);
        Writeln ('password:');
        gotoXY (40,20);
        TextBackground (6); Write (' '); ;
        GotoXY (40,20);

```

```

Textcolor (white);

    if sk <= 9 then for i:=1 to Sk do write ('*') else
        if sk > 9 then
            if simb = 8 then
                begin
                    k:=K-1;
                    for i:=1 to K do write ('*');
                end else
                    for i:=1 to K do write ('*');
Textbackground(white);
    end;
until simb = 13;

    If (pass <>'111') then
begin
    TextBackground (White);
    Clrscr;
    TextColor(Black);
    writeln(' ');
    writeln('      /_____) ) ) ');
    writeln('      (_____) " - ');
    writeln('      ; = ');
    writeln('      //_____/ ');
    writeln('      /)_____\ / ) ) | . _____ | ');
    writeln('      // | - - / \ \ /; | . : | ');
    writeln('      | / | / \ / | . | / ');
    writeln('      ; ::::: | _____ | ');
    writeln('      (/ // // // \ \ \ \ \ | _____ | ');
    writeln('      _ / | _ // // // // / / [ _____ ] - ');
    writeln('      _____ | / / / ');
    writeln('      | | ');
    writeln('      ( | ( | ');
    writeln('      , | | | ');
    writeln('      ; = = ');
    gotoXY(21,15);
    TextColor (white);
    Textbackground (6);
    writeln ('.: [ Numuru uzskaites sistema veisnicai ] :. ');
    Textbackground (white);
    gotoXY(30,19);
    TextColor (white);
    writeln ('Ievadiet Jusu paroli');
    gotoXY(30,20);
    TextColor(white);
    Writeln ('password:');
    gotoXY (40,20);
    TextBackground (6); Write (' '); ;
    GotoXY (15,22);
    TextBackground(white);
    hidecursor;
    Textcolor(31); writeln ('Parole ir nepareiza, spiediet
jebkuru taustinu... ');

```

```

        Readkey;
        goto Login;
    end
else
    begin
        hidecursor;
        menu;
    end;

end.

```

Moduļa St01 pirmteksts

```

unit St01;

INTERFACE
uses crt, st02;
Type
    MenuType = (Vertical);
Const
    width = 50;
    Items1 = 8;

    optText1: array [0..Items1-1] of string =
    ('.[ apskatit datu bazes saturu ].',
    ':[ ierakstit jaunu informaciju ].',
    ':[ dzest ierakstu ].',
    ':[ koriget informaciju ].',
    ':[ sakartot informaciju faila ].',
    ':[ atrast visas brivas istabas ].',
    ':[ atrast informaciju par visdargako numuru ].',
    ':[ exit ].');

    optNormal = 1;
    optSelected = white;

Const
    Max = 100;
    filename = 'hotel.txt';

type
    Data = array [1..Max] of ^Outlaw;
    Outlaw = record
        Vards    : string[12];
        Uzvards  : string[12];
        Numurs   : integer;
        Istabas  : integer;
        Stavoklis : string[12];
        Cena     : real;
    end;

    MyFile = File of Outlaw;

```

```

var
  X, Y,
  selected,
    row : integer;
  _style : menuType;
Textfile : myfile;
chapter : byte;
records : data;
  code : integer;
  ch : char;

procedure content (var textfile: myfile; filename: string);
procedure addrecords(var textfile: MyFile; filename: string);
procedure discharge (var textfile: myfile; filename: string);
procedure position (var textfile : MyFile);
procedure Sort (var textfile: myfile; filename: string);
Procedure expensive (var textfile : myfile; filename : string);
procedure correct (var textfile: myfile; filename: string);
procedure search (var textfile : myfile; filename : string);
procedure menu;

```

IMPLEMENTATION

```

procedure position (var textfile : MyFile);
var
  rec : Outlaw;
begin
  assign (textfile, filename);
  {$I-}
  reset (textfile);
  {$I+}
  Clrscr;
  if (ioresult<>0) then
    begin
      chapter:=1; exit;
    end;
  {$I-}
  read (textfile, REC);
  {$I+}
  if IOresult <> 0 then
    begin
      chapter:=2; Close (textfile); exit;
    end;
  chapter:=0;
  Close (textfile);
end;

(*-----*)

procedure content (var textfile: myfile; filename: string);
var i: integer; REC: Outlaw;

```

```

curr_pos : integer;
refresh : boolean;
begin
  Clrscr;
  {$I-}
  assign (textfile, filename);
  reset (textfile);
  {$I+}
  seek (textfile, 0);
  CLRSCR;
  TextColor (blue);
  Textbackground(White); Clrscr;
  writeln ('-----');
  writeln (' Nr   Vards    Uzvars  Numurs  Brivs numurs  Istabu skaits  cena ');
  writeln ('-----');
  gotoXY (5,22);
  Textcolor (white);
  Textbackground (6);
  write ('Jus varat apskatit visu sarakstu lietojot bultinas "->" un "<-");
  gotoXY (30,24);
  write ('ESC - iziet izvelnee');
  textbackground (white);textcolor(black);
  window (1,4,80,19);
  curr_pos := 0;

  refresh := true;
  repeat
  if refresh then begin
    seek(textfile, curr_pos);
    i := 1;
    Inproc:=False;
    repeat
      {$I-}
      read (textfile, REC);
      {$I+}
      with REC do begin
        textcolor (black);
        if curr_pos + i <= 9 then
          writeln (' ',curr_pos + i,' ',(Vards):7,' ',Uzvars:7,' ',Numurs:3,' ',stavoklis:10,' ',istabas:14,'
          ',Cena:14:2)
        else writeln (curr_pos + i,' ',(Vards):6,' ',Uzvars:6,' ',Numurs:3,' ',stavoklis:11,' ',istabas:14,'
          ',Cena:14:2);
        inc (i);
      end;
    until (i > 15) or (eof (textfile));
    refresh := false;
  end;
  case ord(readkey) of
    77 : begin
      if curr_pos + 15 < filesize(textfile) then begin
        inc(curr_pos, 15); refresh := true; end;
      end;

```

```

75 : begin
    if curr_pos - 15 >= 0 then begin
        dec(curr_pos, 15); refresh := true; end;
    end;
    27 : begin window(1,1,80,25); Clrscr; Inproc:=True; Exit; end;
end;
until false;
window (1,22,80,23); Clrscr;
close (textfile);
textcolor (black);
textbackground(white);
Window(1, 1, 80, 25);
end;

```

(*-----*)

Procedure addrecords (var textfile: MyFile; filename: string);

```

var
    rec : Outlaw;
    str : string;
    ms : array [1..6] of boolean;
    sum : integer;
    quit: boolean;
TypeofS: boolean;
    i : integer;

```

Procedure View;

```

var
command: integer;
    ch : char;
change : boolean;
y,x,i : integer;

```

procedure clean;

```

var
    y1: integer;
begin
    for y1:=7 to 17 do
        begin
            Textcolor (white); Textbackground(white);
            GotoXY (20,y1); write (' ');
        end;
        textcolor (white);
    end;
end;

```

```

begin
    sum:=0;
    command:=1;
    y:=7;
    x:=25;

```

Textbackground(white);

```

gotoXY(20,4);
TextColor (white);Textbackground(6);
Write (' Datus ir jaievada atbilstosos laukos');
Textcolor (blue);Textbackground(white);
gotoXY(25,7);
write ('Ievadiet vardu: '); textbackground(6); write (' ');Textbackground(white);
gotoXY(25,9);
write ('Ievadiet uzvardu: '); textbackground(6); write (' ');Textbackground(white);
gotoXY(25,11);
write ('Ievadiet numuru: '); textbackground(6); write (' ');Textbackground(white);
gotoXY(25,13);
write ('Ievadiet istabu skaitu numuraa: '); textbackground(6); write (' ');Textbackground(white);
gotoXY(25,15);
write ('Vai numurs ir brivs? Ja/ne? '); textbackground(6); write (' ');Textbackground(white);
gotoXY(25,17);
write ('Ievadiet cenu par diennakti: '); textbackground (6); write (' ');Textbackground(white);
gotoXY (20,19);
Textbackground (6);Textcolor (white);
Write ('Uzpiediet enter pirms un pec lauka ievadisanas');textbackground(white);
gotoXY (30,21);
Textbackground (6);Textcolor (white);
Write ('Esc - iziet izvelnee');textbackground(white);

```

```

While True do
begin
quit:=false;
ch:=readkey;
If (sum=6) then Exit;
If ord(ch) = 27 then begin Quit:=True; Exit; end;
if ord(ch) = 72 then
begin
if (command <> 1) then dec (command)
else command:= 6;end
else if ord(ch) = 80 then
begin
if command <> 6 then inc(command)
else command:=1;
end else if ord(ch) = 13 then
begin
case command of

1 : begin
str:="";
ShowCursor;
gotoXY(41,7);textbackground(6); write (' ');gotoXY(41,7);
TypeofS:=True;InputCheck(str,12,TypeofS);
hidecursor;
Rec.vards:=str;
If InProc then begin Close(textfile); Exit; end;
textbackground(white);
ms[command]:=True;

```



```

end;

2 : begin
  Showcursor;
  gotoXY(43,9);textbackground(6); write (' ');gotoXY(43,9);
  TypeOfS:=True;InputCheck(str,12,TypeofS);
  hidecursor;
  If InProc then begin Close(textfile); Exit; end;
  Rec.uzvars:=str;
  textbackground(white);
  ms[command]:=True;
  end;

3 : begin
  Showcursor;
  TypeOfS:=False; gotoXY(42,11);textbackground(6); write (' '); gotoxy(42,11);
  InputCheck(str,3,TypeOfS);
  hidecursor;
  If InProc then begin Close(textfile); Exit; end;
  Val (str,Rec.Numurs,Code);
  textbackground(white);
  ms[command]:=True;
  end;

4 : begin
  Showcursor;
  gotoXY(57,13);textbackground(6);write (' ');gotoXY(57,13);
  TypeOfS:=False; InputCheck(str,2,TypeOfS);
  hidecursor;
  If InProc then begin Close(textfile); Exit; end;
  Val (str, Rec.istabas,Code);
  textbackground (white);
  ms[command]:=True;
  end;

5 : begin
  Showcursor;
  textbackground(6); gotoXY(53,15); write (' ');gotoXY(53,15);
  TypeOfS:=true;InputCheck(str,2,TypeOfS);hidecursor;
  If InProc then begin Close(textfile); Exit; end;
  If ((str <> 'Ja') and (str <> 'Ne')) then
  begin
    gotoxy(5,23);
    Textbackground (lightgreen); textcolor (31);
    write (' Jums ir jaievada "Ja" vai "Ne", spiediet taustinu un tad spiediet enter');
    Textbackground (white); textcolor (white);
    readkey;
    gotoXY(1,23);ClrEOL;
    end
  else begin
    Rec.stavoklis:=str;
    textbackground(white);

```

```

ms[command]:=True;
end;
end;

6 : begin
  Showcursor;
  gotoXY(54,17);TypeOfS:=False; Textbackground(6);write (' ');gotoXY (54,17);
  TypeOfS:=False;InputCheck(str,3,TypeOfS);
  If InProc then begin Close(textfile); Exit; end;
  textbackground (white);
  Val (str,Rec.cena,Code);
  ms[command]:=True;
  hidecursor;
  end;
end;

sum:=0;
for i:=1 to 6 do begin if ms[i] = True then inc (sum); end;
If sum = 6 then begin
gotoXY (18,23);TextColor (31); textbackground (green);
write ('Lai izietu un saglabatu spiediet jebkuru taustinu');
textbackground (white);end;

case command of

  1: begin clean; gotoXY(X-5,Y); textbackground(green); textcolor(white); write ('---->');
textbackground(white); end;
  2: begin clean; gotoXY(X-5,Y + 2); textbackground (green); textcolor(white); write ('---->');
textbackground(white); end;
  3: begin clean; gotoXY(X-5,Y + 4); textbackground (green); textcolor(white); write ('---->');
textbackground(white); end;
  4: begin clean; gotoXY(X-5,Y + 6); textbackground (green); textcolor(white); write ('---->');
textbackground(white); end;
  5: begin clean; gotoXY(X-5,Y + 8); textbackground (green); textcolor(white); write ('---->');
textbackground(white); end;
  6: begin clean; gotoXY(X-5,Y + 10);textbackground (green); textcolor(white); write ('---->');
textbackground(white); end;

  end;
end;
end;

begin
Inproc:=false;
CLRSCR;
{$I-}
assign (textfile, filename);
reset (textfile);
{$I+}
if (ioresult<>0) then begin FError (1); Exit; end;
seek (textfile, filesize (textfile));

```

```

sum:=0;
for i:=1 to 6 do ms[i]:=False;
View;
If (quit=True) or (sum <>6) then begin Exit;end;
{$I-}
write (textfile, REC);
close (textfile);
{$I+}
end;

```

```

(*-----*)

```

```

Procedure DataView (var textfile: myfile; filename: string);
var i: integer; REC: Outlaw;
    curr_pos : integer;
    refresh : boolean;
begin
  Clrscr;
  {$I-}
  assign (textfile, filename);
  reset (textfile);
  {$I+}
  seek (textfile, 0);
  CLRSCR;
  TextColor (blue);
  Textbackground(White); Clrscr;
  writeln ('-----');
  writeln (' Nr   Vards   Uzvars   Numurs   Brivs numurs   Istabu skaits   cena   ');
  writeln ('-----');
  gotoXY (5,21);
  Textcolor (white);
  Textbackground (6);
  write ('Jus varat apskatit visu sarakstu lietojot bultinas "->" un "<-"');
  textbackground (white);textcolor(black);
  gotoxy(25,23);
  textbackground (6); textcolor (white);
  write ('beigt apskatisanu "space" taustins');
  textbackground (white);
  window(1,4,80,19);

  curr_pos := 0;
  refresh := true;
  repeat
  if refresh then begin
    seek(textfile, curr_pos);
    i := 1;
    Inproc:=False;
    repeat
    {$I-}
    read (textfile, REC);
    {$I+}
    with REC do begin

```

```

    textcolor (black);
    if curr_pos + i <= 9 then
        writeln (' ',curr_pos + i,' ',(Vards):7,' ',Uzvards:7,' ',Numurs:3,' ',stavoklis:10,' ',istabas:14,'
',Cena:14:2)
    else writeln (curr_pos + i,' ',(Vards):6,' ',Uzvards:6,' ',Numurs:3,' ',stavoklis:11,'
',istabas:14,' ',Cena:14:2);
    inc (i);
    end;
    until (i > 15) or (eof (textfile));
    refresh := false;
end;

case ord(readkey) of
77 : begin
    if curr_pos + 15 < filesize(textfile) then begin
        inc(curr_pos, 15); refresh := true; end;
    end;
75 : begin
    if curr_pos - 15 >= 0 then begin
        dec(curr_pos, 15); refresh := true;end;
    end;
32 : break;
27 : begin window(1,1,80,25); Clrscr; Inproc:=True; Exit; end;
end;
until false;
window (1,22,80,23); Clrscr;
textcolor (black);
textbackground(white);
Window(1, 1, 80, 25);
end;

```

(*-----*)

```

procedure discharge (var textfile: myfile; filename: string);
var
    REC : Outlaw;
    n,i : integer;
TypeOfS: boolean;
    str : string;
    code : integer;

```

```

begin
    dataview (textfile, filename);
    if InProc then Exit;
    InProc:=false;
    assign (textfile, filename);
    reset (textfile);
    gotoxy(10,23); textbackground (6); textcolor (white);
    writeln ('Kuru Ierakstu velieties dzest?');
    textbackground (white); showcursor;
    gotoXY (42,23); Textbackground (green); write (' ');
    gotoXY (42,23); textcolor (white);

```

```

TypeOfS:=False;
InputCheck (str,2,TypeOfS);
If InProc then begin Close(textfile); Exit; end;
val (str,n,Code);
Textbackground(White);
hidecursor;
if (ioresult <> 0) or ((n < 1) or ( n > filesize (textfile))) then
begin FError (3); exit; end;

n:=n-1; {-1 jo sakam ar 0}
for i := n+1 to pred(filesize(textfile)) do
begin
seek(textfile, i); read(textfile, Rec);
seek(textfile, i-1); write(textfile, Rec);
end;
seek(textfile, filesize(textfile) - 1);
truncate(textfile);
Close (textfile);
FError (4);
Content (textfile,filename);
If Inproc then Exit;
assign (textfile, filename);
{$I-}
reset (textfile);
{$I+}
for i:=1 to filesize(textfile) - 1 do
begin
seek (textfile, i - 1);
read (textfile,rec);
records[i]^:=rec;
end;

Close (textfile);
readkey;
end;

(*-----*)

procedure correct (var textfile: myfile; filename: string);
var
REC,law: Outlaw;
nos: string;
i : integer;
str : string;
code,size : integer;
TypeOfS : boolean;
ch : string[1];

begin
InProc:=False;
Dataview (textfile, filename);
If InProc then begin Close(textfile); Exit; end;

```

```

assign (textfile, filename);
reset (textfile);
size:=filesize(textfile);
gotoxy(10,23); textbackground (6); textcolor (white);
writeln ('ievadiet labojama ieraksta numuru');
textbackground (white); showcursor;
gotoXY (45,23); Textbackground (green); write ( ' ');
gotoXY (45,23); textcolor (white);
TypeOfS:=False;
InputCheck (str,2,TypeOfS);
If InProc then begin Close(textfile); Exit; end;
val (str,i,Code);
Textbackground(White);
hidecursor;
seek(textfile,i-1);
{$I+}
if (ioresult <> 0) or ((i < 1) or ( i > filesize (textfile))) then
begin FError (5); exit; end;

read (textfile,REC);
seek (textfile,i-1);
with REC do
begin
  Clrscr;
  gotoXY(7,23);textbackground (6); textcolor (white);
  write ("J" vai "j" - piekrist, jebkurs taustins - atcelt, "enter" - turpinat');
  textbackground (white);
  gotoXY(20,11); textbackground (6);
  write ('Jus gribat mainit vardu? ja:"j"');textbackground (white);
  gotoXY(53,11);Textbackground(lightgreen); write ( ' ');showcursor;
  gotoXY(53,11); TypeOfS:=True; InputCheck (str,1,TypeOfS); ch:=Str;
  If (Ch = 'J') or (Ch = 'j') then
    begin
      Textbackground(white);
      gotoXY(20,13); textbackground (6); textcolor (white);
      write ( ' Ievadi vardu: ');textbackground (white); gotoXY(37,13);
      textbackground(lightgreen); write ( ' ');
      gotoXY(37,13); TypeOfS:=True; InputCheck(str,12,TypeOfS); vards:=str;
      If InProc then begin Close(textfile); Exit; end;
      textbackground(white);
      ClrEOL;
    end
  else if inproc then begin close (textfile); exit; end;

  textbackground(white);
  gotoXY(1,11); ClrEOL;
  gotoXY(1,13); ClrEOL;
  gotoXY(20,11);textbackground (6);
  write ('Jus gribat mainit uzvardu? Ja:"j"'); Textbackground(white);
  gotoXY (54,11);Textbackground (lightgreen); write ( ' ');
  gotoXY(54,11); TypeOfS:=True; InputCheck (str,1,TypeOfS); ch:=Str;
  If (Ch = 'J') or (Ch = 'j') then

```

```

begin
  Textbackground(white);gotoXY(20,13); textbackground (6);
  textcolor (white);write (' Ievadi uzvardu: '); textbackground(white);
  gotoXY (40,13);textbackground (lightgreen); write (' ');
  gotoXY(40,13);InputCheck(str,12,TypeOfS); uzvards:=str;
  If InProc then begin Close(textfile); Exit; end;
  textbackground (white);
end
else if inproc then begin close (textfile); exit; end;

textbackground(white);
gotoXY(1,11); ClrEOL;
gotoXY(1,13); ClrEOL;
gotoXY(20,11); textbackground (6);
write ('Jus gribat mainit numuru?Ja:"j"'); textbackground (white);
gotoXY (54,11); textbackground(lightgreen);write (' ');gotoXY (54,11);
TypeOfS:=true; Inputcheck (str,1,TypeOfS); ch:=Str;
If (Ch = 'J') or (Ch = 'j') then
  begin
    Textbackground(white); gotoXY(20,13); textbackground (6);
    textcolor (white); write (' Ievadi numuru: '); textbackground(white);
    gotoXY (40,13); textbackground (lightgreen); write (' ');
    gotoxy(40,13); TypeOfS:=false; InputCheck(str,2,TypeOfS);
    If InProc then begin Close (textfile); Exit; end;
    val (str, numurs, code);
    textbackground(white);
  end
else if inproc then begin close (textfile); exit; end;

textbackground(white);
gotoXY(1,11); ClrEOL;
gotoXY(1,13); ClrEOL;
gotoXY(20,11); textbackground(6);
write ('Jus gribat istabu sk. numuraa? Ja:"j"'); Textbackground(white);
gotoXY(60,11); textbackground (lightgreen); write (' ');
gotoXY(60,11); TypeOfS:=True; InputCheck (str,1,TypeOfS); ch:=Str;
If (Ch = 'J') or (Ch = 'j') then
  begin
    textbackground(white);gotoXY(20,13); textbackground (6);
    write (' Ievadi skaitu: '); textbackground(white);
    gotoXY(40,13); textbackground (lightgreen); write (' ');
    gotoXY(40,13); typeOfS:=false; InputCheck(str,2,TypeOfS);
    If InProc then begin Close(textfile); Exit; end;
    val (str,istabas, code);
    textbackground(white);
  end
else if inproc then begin close (textfile); exit; end;

textbackground(white);
gotoXY(1,11); ClrEOL;
gotoXY(1,13); ClrEOL;
gotoXY(20,11); textbackground (6);

```

```

write ('Jus gribat mainit numura stavokli? Ja:"j""); Textbackground(white);
gotoXY (62,11); textbackground (lightgreen); write ( ' ');
gotoXY(62,11); TypeOfS:=True; InputCheck (str,1,TypeOfS); ch:=Str;
If (Ch = 'J') or (Ch = 'j') then
begin
textbackground(white); gotoXY(20,13); textbackground (6);
write ('Numurs brivs?Ja/Ne?:'); textbackground(white);
gotoXY(42,13); textbackground (lightgreen);write ( ' ');
gotoXY(42,13); TypeOfS:=true; InputCheck(str,2,TypeOfS); stavoklis:=str;
If InProc then begin Close(textfile); exit; end;
Textbackground(white);
end
else if inproc then begin close (textfile); exit; end;

textbackground(white);
gotoXY(1,11); ClrEOL;
gotoXY(1,13); ClrEOL;
gotoXY(20,11); textbackground (6);
write ('Jus gribat mainit cenu? Ja:"j""); Textbackground(white);
gotoXY(54,11);textbackground (lightgreen); write ( ' ');
gotoXY(54,11); TypeOfS:=True; InputCheck (str,1,TypeOfS); ch:=Str;
If (Ch = 'J') or (Ch = 'j') then
begin
textbackground(white);gotoXY(20,13); textbackground(6);
write ( ' Ievadi cenu: '); textbackground(white);
gotoXY(40,13); textbackground (lightgreen); write ( ' ');
gotoXY(40,13); TypeOfS:=false; InputCheck(str,3,TypeOfS);
val(str,cena,code);
If InProc then begin Close(textfile); Exit; end;
textbackground (white);
end
else if inproc then begin close (textfile); exit; end;
textbackground (white);
write (textfile, REC);
writeln;
close (textfile);
hidecursor;
FError (6);
Content (textfile, filename);
if Inproc then exit;
assign (textfile, filename);
{$I-}
reset (textfile);
{$I+}
for i:=1 to filesize(textfile) - 1 do
begin
seek (textfile, i - 1);
read (textfile, rec);
records[i]^:=rec;
end;
Close (textfile);
readkey;

```



```
end;  
end;
```

```
(*-----*)
```

```
procedure Sort (var textfile: myfile; filename: string);
```

```
var
```

```
    save, rec: Outlaw;  
    nos: string;  
    TypeOfS : boolean;  
    str : string;  
    Findtrue : boolean;  
    i,j : integer;
```

```
begin
```

```
    InProc:=false;  
    assign (textfile, filename);  
    reset (textfile);  
    Clrscr;  
    gotoXY(35,17);  
    textbackground (6);  
    Textcolor (white);  
    Write ('ESC - iziet');  
    textbackground (white);  
    Findtrue:=false;  
    repeat  
        gotoXY(17,13); textbackground (6);  
        write ('Kartosim pec istabu skaita vai uzvardiem?I/U?');  
        textbackground (white);gotoXY(65,13); textbackground (lightgreen); write (' ');  
        showcursor; gotoXY(65,13); TypeOfS:=True; InputCheck (str,1,TypeOfS);  
        Textbackground(white); ch:=str[1];  
        if Inproc then Exit;
```

```
    Findtrue:=false;
```

```
    if (ch = 'I') then
```

```
    begin
```

```
        Findtrue:=true;  
        for i:= 0 to (filesize (textfile)-1) do  
            for j:= 0 to (filesize (textfile)-1) do  
                begin  
                    seek (textfile, i);  
                    read (textfile, rec);  
                    save:=rec;  
                    seek (textfile, j);  
                    read (textfile, rec);  
                    if save.istabas < rec.istabas then  
                        begin  
                            seek (textfile, i);  
                            write (textfile, rec);  
                            seek (textfile, j);  
                            write (textfile, save);
```

```

        end;
    end;
end else
    if (ch = 'U') then
begin
    Findtrue:=true;
    for i:= 0 to (filesize (textfile)-1) do
        for j:= 0 to (filesize (textfile)-1) do
begin
            seek (textfile, i);
            read (textfile, rec);
            save:=rec;
            seek (textfile, j);
            read (textfile, rec);
            if save.uzvards < rec.uzvards then
begin
                seek (textfile, i);
                write (textfile, rec);
                seek (textfile, j);
                write (textfile, save);
            end;
        end;
    end else If ord(ch) = 27 then Exit
    else If ((ch <> 'U') and (ch <> 'A')) then
begin
        gotoxy(WhereX - 1, WhereY);
        TextColor (green); Write ( ' ');
        gotoxy(WhereX - 1, WhereY);
        Textcolor(white);
    end;
    Until Findtrue;

    Close (textfile);
    hidecursor;
    FError (6);
    Content (textfile,filename);
    If Inproc then Exit;

    assign (textfile, filename);
    {$I-}
    reset (textfile);
    {$I+}
    for i:=1 to filesize(textfile) - 1 do
begin
        seek (textfile, i - 1);
        read (textfile,rec);
        records[i]^:=rec;
    end;
    Close (textfile);
    readkey;
end;

```

(*-----*)

Procedure expensive (var textfile : myfile; filename : string);

Const

 holdname = 'holdtext.txt';

var

save,rec : Outlaw;

 str : string;

 typeofS: boolean;

 hold : myfile;

sum,k,i : integer;

 count : real;

begin

InProc:=false;

assign (textfile, filename);

reset (textfile);

assign (hold,holdname);

rewrite (hold);

count:=0;

sum:=0;

for i:= 0 to (filesize (textfile)-1) do

begin

 seek (textfile, i);

 read (textfile, rec);

 if rec.cena > count then

 count:=rec.cena;

end;

for k:= 0 to (filesize (textfile)-1) do

begin

 seek (textfile,k);

 read (textfile,rec);

 with REC do If ((REC.cena) = count) then begin

 inc (sum); write (hold,REC); end;

end;

Close (textfile);

Close (hold);

If sum > 0 then

begin

 gotoXY(10,22); textbackground (6); Textcolor (31);

 write ('Kopigi sameklets(ti) ',sum,' numurs(ri) , spiediet jebkuru taustinu');

 textbackground (white);textcolor(white);

 readkey;

 Content (hold,holdname);

end;

```

end;

(*-----*)

procedure search (var textfile : myfile; filename : string);
Const
    roomname = 'roomdata.txt';
var
    rec : Outlaw;
    yeah : string;
    room : MyFile;
    k,i,sum : integer;

begin
    InProc:=false;
    assign (textfile, filename);
    reset (textfile);
    assign (room,roomname);
    rewrite (room);

    Clrscr;
    k:=0;
    sum:=0;
    yeah:='Ja';

    for i:= 0 to (filesize (textfile)-1) do
    begin
        seek (textfile, i);
        read (textfile, rec);
        with REC do If ((rec.stavoklis) = yeah) then begin
            sum:= sum + rec.istabas; write (room,rec); end;
        end;

    Close (textfile);
    Close (room);

    If sum > 0 then
    begin
        gotoXY(10,22); textbackground (6); Textcolor (31);
        write ('Kopigi samekletas ',sum,' istabas , spiediet jebkuru taustinu');
        textbackground (white);textcolor(white);
        readkey;
        Content (room, roomname);
        end;

    end;

    (*-----*)

procedure MakeMenu (optText: array of string; MaxItems: integer);
var
    i, _X: byte;

```

```

begin
  Y := row;
  _X := X;
  for i := 0 to MaxItems-1 do
    begin
      GoToXY (_X, Y);
      if i = selected then
        TextColor (optSelected)
      else
        TextColor (optNormal);
      write (optText[i]);
      inc (Y,1);
    end;
  end;
end;

(*-----*)

function MenuOption (optText: array of string; MaxItems: integer): byte;
var
  ch: char;
begin
  selected := 0;
  X := (80 - width) div 2;
  row := (25 - MaxItems) div 2;

  repeat
    MakeMenu (optText, MaxItems);

    ch := readkey;
    if ch = #0 then
      ch := readkey;

    case ch of
      #80, #77: {Down/Right}
      begin
        inc (Selected);
        if Selected = MaxItems then
          Selected := 0;
        MakeMenu (optText, MaxItems);
      end;

      #72, #75: {Up/Left}
      begin
        dec (Selected);
        if Selected < 0 then
          Selected := MaxItems-1;
        MakeMenu (optText, MaxItems);
      end;
    end;
  until ch = #13; {Enter}

  MenuOption := Selected + 1;

```

```

    TextColor (optNormal);
    If _style = Vertical Then
        clrscr;
end;

(*-----*)

procedure menu;
var
    Option, i: byte;

begin
    TextBackground (white);
    clrscr;
    hidecursor;
    Textbackground(6);
    Textcolor(white);
    gotoxy(14,5);write (' Jusu riciba ir viesnicas datu bazes informacija ');
    Textbackground(white);
    TextColor (1);
    Gotoxy(15,7);
    for i:= 1 to 48 do write ('H');
    Gotoxy(15,16);
    for i:= 1 to 48 do write ('H');
    Gotoxy(14,7); write ('Ī');
    Gotoxy(14,16); write ('Ī');
    Gotoxy(63,7);write ('»');
    Gotoxy(63,8);write ('e');
    Gotoxy(63,9);write ('e');
    Gotoxy(14,8);write ('e');
    Gotoxy(14,9);write ('e');
    Gotoxy(14,10);write ('e');
    Gotoxy(14,11);write ('e');
    Gotoxy(14,12);write ('e');
    Gotoxy(14,13);write ('e');
    Gotoxy(14,14);write ('e');
    Gotoxy(14,15);write ('e');
    Gotoxy(63,10);write ('e');
    Gotoxy(63,11);write ('e');
    Gotoxy(63,12);write ('e');
    Gotoxy(63,13);write ('e');
    Gotoxy(63,14);write ('e');
    Gotoxy(63,15);write ('e');
    Gotoxy(63,16);write ('j');
    Gotoxy(9,18);Textbackground(6);
    Textcolor(white);
    write('Parvietoties pa izvelni var ar taustinu "->" un "<-" palidzibu');
    Textbackground(white);

    _style := Vertical;
    Option := MenuOption (optText1, Items1);

```

case option of

```
1: begin;
    Textbackground (white);
    Clrscr;
    position (Textfile);
    if (chapter = 1) then begin FError (1); menu; end else
        if (chapter = 2) then begin FError (2); menu; end;
    if (chapter = 0) then begin
        Content (textfile, filename); menu; end;
    end;

2: begin
    Textbackground (White);
    Clrscr;
    position (Textfile);
    if (chapter = 0) or (chapter = 2) then begin
        addrecords (Textfile, filename); menu; end
    else if (chapter = 1) then begin FError (1); menu; end;
end;

3: begin
    Textbackground (white);
    Clrscr;
    position (textfile);
    if (chapter = 1) then begin FError (1); menu; end else
        if (chapter = 2) then begin FError (2); menu; end;
    if (chapter = 0) then begin
        discharge (textfile, filename); menu; end;
    end;

4: begin
    Textbackground (white);
    Clrscr;
    position (textfile);
    if (chapter = 1) then begin FError (1); menu; end else
        if (chapter = 2) then begin FError (2); menu; end;
    if (chapter = 0) then begin
        correct (textfile, filename); menu; end;
    end;

5: begin
    Textbackground (white);
    Clrscr;
    position (textfile);
    if (chapter = 1) then begin FError (1); menu; end else
        if (chapter = 2) then begin FError (2); menu; end;
    if (chapter = 0) then begin
        sort (textfile, filename); menu; end;
    end;

6: begin
    Textbackground (white);
    Clrscr;
```

```

    position (textfile);
    if (chapter = 1) then begin FError (1); menu; end else
        if (chapter = 2) then begin FError (2); menu; end;
    if (chapter = 0) then begin
        search (textfile, filename); menu; end;
    end;
7: begin
    textbackground (white);
    Clrscr;
    position (textfile);
    if (chapter = 1) then begin FError (1); menu; end else
        if (chapter = 2) then begin FError (2); menu; end;
    if (chapter = 0) then begin
        expensive (textfile, filename); menu; end;
    end;
8: begin
    exit;
end;

    end;
end;

end.

```

Moduļa st02 pimteksts

```
unit st02;
```

```
INTERFACE
```

```
uses crt, dos;
```

```
var
```

```
    InProc : Boolean;
```

```
procedure FError (number : integer);
```

```
procedure InputCheck(var buf : string;max_len : integer; param:boolean);
```

```
procedure ShowCursor;
```

```
procedure HideCursor;
```

```
implementation
```

```
procedure FError (number : integer);
```

```
var
```

```
    str : string;
```

```
begin
```

```
case number of
```

```
1: begin
```

```
    Clrscr;
```

```
    gotoXY(10,21); Textbackground (6); TextColor(31);
```

```
    write ('Kluda, fails netika atverts, spiediet jebkuru pogu');
```

```
    Textbackground (white); Textcolor(black);
```

```
    readkey;
```



```

    exit;
end;

2: begin
    Clrscr;
    gotoXY (10,21); Textbackground (6); Textcolor(31);
    write ('Datu fails nesatur neviena ieraksta, spiediet jebkuru pogu');
    Textbackground (White); Textcolor(black);
    readkey;
    exit;
end;

3: begin
    Clrscr;
    gotoXY(15,22); TextColor(31); Textbackground (6);
    writeln ('Sads ieraksts neeksiste. Spiediet jebkuru taustinu. ');
    Textbackground (white); Textcolor(black);
    readkey;
    Exit;
end;

4: begin
    Clrscr;
    gotoXY(15,22); TextColor(31); Textbackground (6);
    writeln ('Ieraksts tika izdzest, nospiediet taustinu');
    Textbackground (white); Textcolor(black);
    readkey;
    Exit;
end;

5: begin
    Clrscr;
    gotoXY(15,22); TextColor(31); Textbackground (6);
    writeln ('Jus noradijat nepieciezamo informaciju, nospiediet taustinu');
    Textbackground (white); Textcolor(black);
    readkey;
    Exit;
end;

6: begin
    Clrscr;
    gotoXY(15,22); TextColor(31); Textbackground (6);
    writeln ('Operacija veiksmigi izpildita, nospiediet taustinu');
    Textbackground (white); Textcolor(black);
    readkey;
    exit;
end;
end;
end;

procedure HideCursor;
var

```

```

    regs : registers;
begin
    regs.ax := $0100;
    regs.cx := $2607;
    intr($10, regs);
end;

procedure ShowCursor;
var
    regs : registers;
begin
    regs.ax := $0100;
    regs.cx := $0506;
    intr($10, regs);
end;

procedure InputCheck(var buf : string; max_len: integer; param: boolean);
var
    Active : Integer;
    str : string;
    ch : char;
    start_pos,i : integer;
type
    tset = set of char;
const
    char_set: array[boolean] of tset = (
        ['0' .. '9', ':', '.'],
        ['a' .. 'z', 'A' .. 'Z', '_', #32]
    );
begin
    str := "";
    start_pos:= WhereX;
    Active := 1;
    repeat
        gotoxy(start_pos, WhereY);
        for i:=(start_pos) to (start_pos + max_len - 1) do
            begin
                write (' ');
            end;
        gotoxy(start_pos, WhereY);
        write(str);
        gotoxy(start_pos + Active - 1, WhereY);

        ch := readkey;
        case ch of
            #0:
                case readkey of
                    #75:
                        if Active > 1 then begin { LEFT }
                            dec(Active);
                            gotoXY(start_pos + Active - 1, WhereY);

```

```

end;

#77:
if WhereX <= (start_pos + Length(str) - 1) then begin { RIGHT }
  inc(Active);
  gotoXY(start_pos + Active - 1, Wherey);
end;

#83:
if Active <= Length(str) then
  Delete(str, Active, 1); { DELETE }

end;

#8:
if length(str) <> 0 then begin { BS }
  dec(Active);
  gotoXY(start_pos + Active - 1, Wherey);
  Delete(str, Active, length(str));
end;

#13: If Length(str) > 0 then break else continue;

#27: Begin Inproc:=True; Exit; end;

else begin

  if (length(str) < max_len) and (ch in char_set[param]) then begin
    If (Active = 1) and (param) then ch:=Upcase(ch)
    else if (param) and (ch In ['A'..'Z']) then ch:=chr(ord(ch) + 32);

    if Active <= Length(str) then Insert(ch, str, active)
    else str := str + ch;
    inc(Active);

  end;

end;

end;

until false;

if (Length(str) < Max_Len) and param then for i:= (Length(str) + 1) to (Max_Len) do str:= str
+' ';
  buf := str;

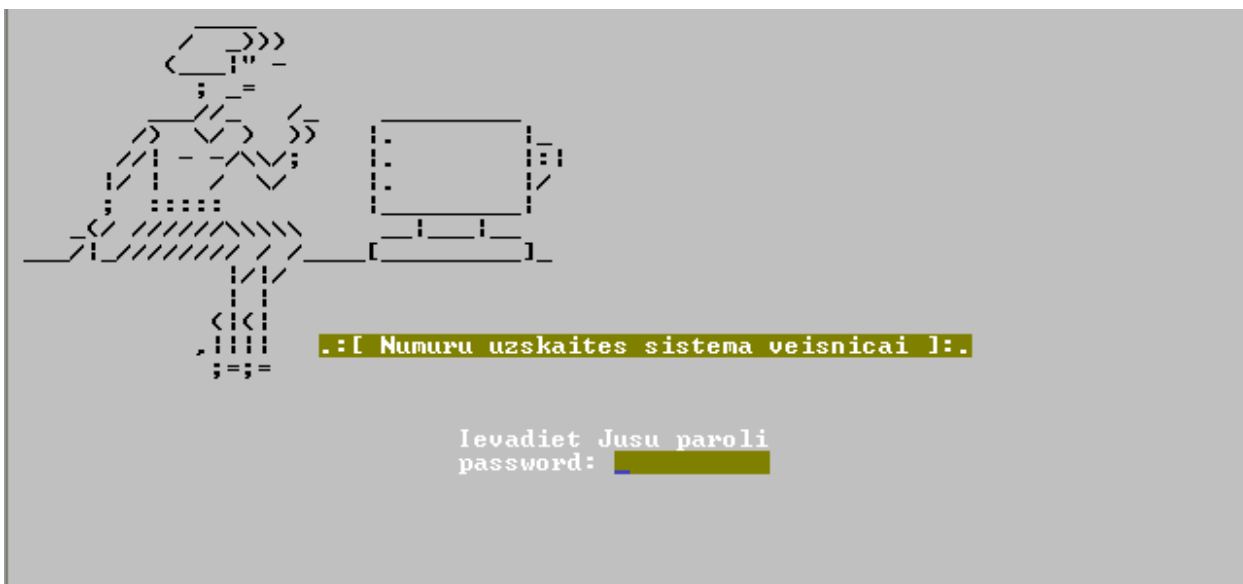
end;

end.

```

7 Instrukcija lietotājam

Pēc programmas palaišanas seko lietotāja autentifikācijas logs, kura lietotājam ir jāpiereģistrējas sistēmā, ja parole tika ievadīta nepareizi, programma lūdz ievadi paroli atkārtoti. [skat. att. 7.1].



7.1 att. Lietotāja autentifikācijas logs.

Pēc autentifikācijas loga seko galvenā programma, kura satur izvēlni ar pieejamām pamatoperācijām, lietotājs var brīvi pārvietoties pa izvēlni, lietojot šos taustiņus [skat. Att. 7.2]. Šī taustiņu kombinācija tiek pielietota pamatoperācijās, kuras ir darbības ar faila apskatīšanu (lietojot šos taustiņus lietotājs var aplūkot pilno sarakstu).

Parvietoties pa izvēlni var ar taustiņu ">" un "<" palīdzību

7.2 att. Taustiņu kombinācija.

Turpmākas programmas pielietošana ir viegla, jo pildot kādu no operācijām, lietotājam vienmēr tiek izskaidrots, ka rīkoties kāda no noteiktajām situācijām [skat. att. 7.2].

"J" vai "j" – piekrist, jebkurs taustīns – atcelt, "enter" – turpināt

7.3 att. Lietotāja instrukcija.

Jā lietotājs veic kādu no nepareizām darbībām, vai arī programma veic kādu no operācijām, programma izveda paziņojumus lietotājam.

8 Secinājumi

Veidojot šo studiju darbu nostiprināju savas zināšanas algoritmizācija un programmēšana, kā arī apguvu iemaņas programmatūras pielāgošana.

Darba gaitā es izpildīju visu to, ko gribēju panākt. Ieguvu pieredzi neliela projekta veidošanā, ka arī ieguvu vērtīgo pieredzi programmas atklādošana un testēšanas metodikā.