

# Design of Algorithms for Improvement of Delphi's Controls DBGrid and DBNavigator

Fadil Novalic<sup>1</sup>, Bekim Fetaji<sup>2</sup>, Mensura Kudumovic<sup>3</sup>

<sup>1</sup>Department for Computer Sciences, University of Novi Pazar, Serbia

<sup>2</sup>Contemporary Sciences and Technologies, South East European University, Tetovo, Macedonia

<sup>3</sup>Faculty of Medicine, University of Sarajevo, Sarajevo, Bosnia and Herzegovina

**Abstract:** This paper is occupied with the improvement of three main functions of the preview of information from database in a table form: storage of changes or new entry in the table, delete of records from the table and sorting of information by clicking on the heading of column in the table information review.

These improvements were obtained by occurrence coding connected t DBGrid. The accomplishment of these functions is regulated by the way that users of Windows applications accepted it as usual in that businesses, so the regulations of these applications which are programmed in Delphi improved its quality. Very important attribute of this paper is that the coding of algorithms have been done for using its procedures without a change, what creates the modularity of the source code.

**Keywords:** Delphi DBGrid, Delphi DBNavigator, Design Algorithms, Improving Algorithms, Sorting Algorithm

## 1. Introduction

Algorithm is defined as an accurate computer procedure which requires some values, or group of values, as an entrance and accords any value or group of values as an exit [1].

The implementation of algorithms is very expanding. Algorithm is written on the basis of mathematic structure which mostly corresponds to the received problem, and sometimes the existing algorithms are analyzed or repaired for the received problem, which is to be solved by a computer tool. [2].

In order to solve one problem, many variously correct algorithms can be designed. Such algorithms are called equivalent algorithms and for the same entrance dimension give similar exits [2].

The algorithms which are the subject of this work, have been developed because of requirements of some of the basic application functions which show the data from the database, but developed in

Delphi, in order to be modulated for the users' practice. During the development of some information systems and practical tutoring with students, we came to the conclusion that we should improve three basic functions of showing data from database in the table form:

1. storing of changes or new entry in the table
2. deleting of records from the table and
3. sorting of data by pressing the heading of column in the table information review.

These improvements were achieved along with event coding referring to DBGrid: storing of information is done by pressing the key Enter on the keyboard : procedure DBGridKeyPress), delete by click on the key Delete (procedure DBGridKeyUp), and sorting by pressing the mouse on the heading column in the table review (procedure DBGridTitleClick). The process of deleting and sorting information is done by pressing the keyboard instead of pressing with mouse on the buttons that are on Delphi's control of DBNavigator which was made for the process. The modularity of the source code was achieved in the way that Delphi's control names weren't used in this paper, but to proceed them with calling in the argument attributes of the procedure where the code is written. It enables those who want to use these improvements for their Delphi's project to copy the code into the procedure without any changes, and even if he gives other names to controls or set (Unit) Data Set controls for connecting with the base. These procedures were written for ADO DataSet controls.

The simple example of data review was elaborated in this paper. The algorithms were represented descriptively. The way of algorithm progress and source code written for these algorithms has started from elaborating the main command, then

it is completed with the condition analyzing that should be accomplished in order to avoid errors in the course of program activation.

## 2. Related Works

There are many works related to the algorithms implementation in different fields, what shows its rifeeness. For example, Dynamical Visualization Process [4], computer graphic[5] etc. Many scientific works are occupied with the problems of data sorting in the shape of the table which displays Delphi's control DBGrid, as Web portal.[8], [9].

The paper *Improving Delphi TDBGrid* [6] represents some of the functions for improving DBGrid, which makes it more user friendly. There are next improvements: the change of a background color according to the cell type, the integration of textual data searching, arrange of columns, an automathic arrange of column's wideness, the scroll by the wheel on the mouse, word wrapping cell, sound effects and integration like the menu for separate records in the base.

The paper [7] describes the internal realization mechanism of DBGrid control, and introduces how to realize displaying of serial number of record, sorting of column title and saving and restoring of column width and column sequence.

It is explained how to sort DBGrid with the help of THeaderControl component on the Web page . This code is written for Form's OnCreate event.

At the Delphi's portal <http://delphi.about.com>, many useful procedures can be found, and in the article [9] some are explained for shape change of mouse when it is above the article column and for the data sorting in DBGrid by pressing the heading of the column. The procedure for sorting contains the source code which enables the change of font style in the heading of the column where the data were sorted.

The source code for arranging the data in DBGrid, from the web portal example and from the book of Delphi [3], depends on the names of Delphi's controls that are involved in the procedures. It what differs these procedures from mentioned examples is the attribute of the modularity and simplicity for understanding, what manages its wide implementation.

We didn't find the procedures with the property of modularity of source code, which manages the storage and delete of data by pressing the key of keyboard.

## 3. The Description of Delphi's Application for Data Displaying from the Database in Dbgrid

Delphi's control of DBGrid realizes the displaying of data from the particular tables in the database in a table shape. For the need of this paper or research was taken simple example involving students. In the picture 1. is shown Delphi's application with displayed data in DBGrid. The data are sorted in hierarchy in the database.

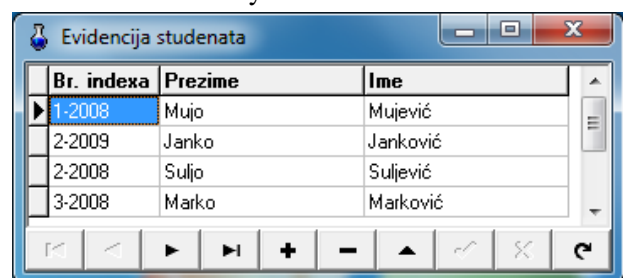


Figure 1. Window of Delphi's application with DBGrid and DBNavigator

To display the data from the base in DBGrid, the link between a component and data has to be appointed. This link is a chain of few components. The first chain is **ADOConnection** control, which serves as a connector to the database. The next chain is a control **AdoTable**. This control is connected to ADOConnection, and its attribute is **TableName** which assesses the connection between the table in the database. ADOTable is a control ,whose attributes would be used in the described procedures in this research. ADOTable is joined as a property of DataSet as next control in the chain of **DataSource**. This control is attached to DBGridu via its property **DataSource**. After mentioned connections , the property **Active** is necessary for ADOTable to be set at **True** and the data will be seen in DBGrid.

At the bottom of the window is controlDBNavigator. Through its property DataSource connected with it and the buttons on it are used to operate with the data from connected table. The control DBGrid and DBNavigator are connected to the same DataSource and so their interacting is realized, that is the possibility to operate with the same data.

#### 4. An Algorithm for Storing Data via DBGrid Pressing the Key ENTER

When the data is shown in Delphi's applications from the database in the control DBGrid, then it can be simply changed with selecting the cells and retyping the new contents. Also the new data entry can be realized by getting into a new line before last in the records in DBGrid. In that way, the new way of data entry is realized. The storing of new changes or new entry can be realized in two ways: some new record should be selected in DBGrid or to click the button of DBNavigator, which is expected for storing changes in the database. This button is named in the picture 2. as Save.



Figure 2. DBNavigator

In this research, the algorithm was designed to realize the task of storing the data changes from the database with one press on the key Enter on the keyboard.

In Delphi, when the connection to database is realized via ADO components for sorting the changes in the database is realized with the method Post in ADODataset control. This method should be used by pressing the key ENTER. Algorithm for storing the data can be described through one step:

##### **Algorithm 1: Storing the changes by pressing the key ENTER**

- Step 1: Was Enter pressed?
  - **Yes:** Save the changes

The source code for this algorithm should be written in the procedure, which is connected to OnKeyPress from DBGrid control:

```
procedure TForm.DBGridKeyPress(Sender: TObject; var Key: Char);
begin
  if Key = #13 then
    TDBGrid(Sender).DataSource.DataSet.Post;
end;
```

To succeed the modularity of the source code of this procedure, to DataSet control from where the method should be realized through the argument property of procedure **Sender**. When this argument is

set as the argument of the type **TDBGrid**, we will get DBGrid from which is realized OnKeyPress. Calling its property **DataSource**, then the property DataSet from DataSource control, we come to DataSetcontrol. Instead of name DataSetcontrol, in the source code we will use the term **TDBGrid(Sender).DataSource.DataSet**. This term will be used in every procedure elaborated in this research. After the completeness of this change data or the new data entry will be sorted. The absence of this algorithm is the display of error in realization, if the key ENTER is pressed in situation when the new data change or data entry is not done, e.g. when DataSet control isn't in Edit or in Insert mode. In the picture 3. Is shown DialogBox with the message about the error: "Dataset not in edit or insert mode".

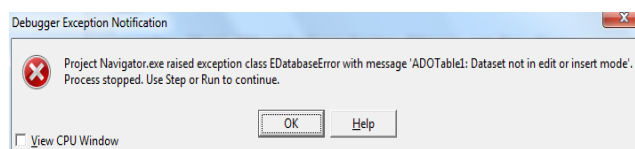


Figure 3. The message about an error occurred during the realization of Post method when the changes were not made

To eliminate the possibility of this error appearance, one more question in an algorithm should be added about the state of DataSet control:

##### **Algorithm 2:** The sorting of the changes by pressing the key ENTER and analyzing the status of DataSet control

- Step 1: Was ENTER pressed?
  - **Yes:**
    - Is it in DataSet or Edit or Insert modue
    - **Yes:** Save the changes.

The source code should be displayed now:

```
procedure TForm.DBGridKeyPress(Sender: TObject;
var Key: Char);
begin
  if Key = #13 then
    if TDBGrid(Sender).DataSource.DataSet.State
      in [dsEdit, dsInsert] then
      TDBGrid(Sender).DataSource.DataSet.Post;
end;
```

#### 4.1. An Algorithm for Cancelling the Changes or New Data Entry via DBGrid by Pressing the Key ESC

To cancel the sorting or the new data entry, it is needed to click DBNavigator button CANCEL. (Picture2.). This procedure should be realized

pressing the key ESC in the keyboard, because it's the regular custom of the user of Windows applications to cancel some important procedure.

What key ESC realizes without additional programming when OnKeyPress is in DBGrid, to cancel the change of an item field in the record which was currently selected. However, the changes within other item fields of that record which have been already realized, will be done moving in other record. It's needed to use the method CANCEL above DataSet control, when it's wanted to cancel the changes in the record. An algorithm for this procedure is similar to algorithm of storing the changes pressing the key ENTER. The first is in ASCII value for pressing the key, because we now examine if the key ESC is pressed, whose value is 27. The second difference is in the method, instead of Post, we now use the method Cancel.

**Algorithm 3:** *The cancel of the changes pressing the key ESC*

---

- o Step 1: has the key ESC been pressed?
  - **Yes:** Cancel the changes.
- 

To realize the method Cancel, it's not needed for Data Set control to be in Edit or Insert mode. The source code for this algorithm is also written in procedure DBGridKeyPress:

```
procedure TForm.DBGridKeyPress(Sender: TObject;
var Key: Char);
begin
  if Key = #27 then
    TDBGrid(Sender).DataSource.DataSet.Cancel;
end;
```

Because the source code for both algorithms (2 i 3) is written in the same procedure, we will show the complete procedure:

```
procedure TForm.DBGridKeyPress(Sender: TObject;
var Key: Char);
begin
  if Key = #13 then
    if TDBGrid(Sender).DataSource.DataSet.State
    in [dsEdit, dsInsert] then
      TDBGrid(Sender).DataSource.DataSet.Post;
  if Key = #27 then
    TDBGrid(Sender).DataSource.DataSet.Cancel;
end;
```

The source code of this procedure will operate in every Delphi's project without any changes in the procedures OnKeyPress at DBGrid.

**4.2. The Comparison of the Two Equivalent Algorithms for Storing the Changes or Entries in the Database**

An algorithm which responds to the whole procedure DBGridKeyPress is the following:

**Algorithm 4:** *Storing or cancelling the changes pressing the key ENTER or ESC on keyboard*

---

- o Step 1: Was the key ENTER pressed?
  - **Yes:**
    - Is DataSet in Edit or Insert mode?
    - **Yes:** Save the changes.
  - o Step 2: Is ESC pressed?
  - **Yes:** Cancelling the changes.
- 

The equivalent algorithm 4 can be written in the next:

**Algorithm 5:** *Storing or cancelling the changes pressing the key ENTER or ESC in the keyboard*

---

- o Step 1: Is DataSet in Edit or in Insert mode ?
  - **Yes:**
    - Is DataSet in Edit or in Insert mode?
    - **Yes:** Save the changes.
  - o Step 2: Is ESC pressed?
  - Yes:** Cancelling the changes
- 

The next results are achieved, by testing the algorithms (4 i 5) :

- Step 2 is the same with both algorithms.
- In the first step, two conditions were examined: is the key ENTER pressed and status of DataSet control.

In algorithm 4, the first question is if the key ENTER was pressed, then the status of DataSet control is examined. So, at the every pressing the key means the examination if the key ENER is pressed, and that the status of DataSet control is examined only when the first condition is realized:

The first condition at algorithm 5 is examination of DataSet control status. At every key pressing is examined if the key ENTER was pressed, and then DataSet control is only examined if the first condition was accomplished. The first condition at algorithm 5 is the examination of DataSet control, and the value of ASCII code of pressed code is searched, when mentioned status is in Edit or Insert.

The results of algorithm testing bring the conclusion that in algorithm 5 more often both conditions of the step 1 are examined, because of reason whenever the keys in the keyboard are pressed, when the procedure DBGridKeyPress is realized, those are the new changes or entries, what means that the condition of DataSet control to be in Edit or Insert mode is accomplished. Accordingly, the algorithm 4 is realized faster: We will entrance the source code written on the basis of algorithm 5.

```

procedure TForm.DBGridKeyPress(Sender: TObject;
var Key: Char);
begin
  if TDBGrid(Sender).DataSource.DataSet.State
  in [dsEdit, dsInsert] then
    if Key = #13 then
      TDBGrid(Sender).DataSource.DataSet.Post;
    if Key = #27 then
      TDBGrid(Sender).DataSource.DataSet.Cancel;
end;

```

### 5. An Algorithm for Deleting the Data via Dbgrid Pressing the Key DELETE

The delete of selected record from shown in DBGrid, are realized by clicking the button DELETE on DBNavigator. (Picture2.). Before the delete of the application creates the possibility of acknowledgment or cancelling the delete, what is shown in picture 4.

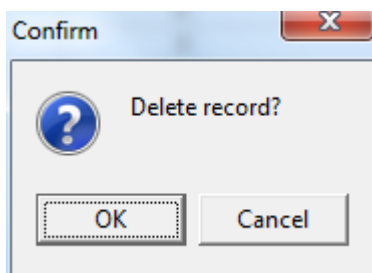


Figure 4. DialogBox for affirmation of delete taken by DBNavigator

The algorithm for deleting the records elaborated in this research enables the construction of questions about acknowledgement for delete, in the way the programmer choose with the corresponding language. After acknowledgement of delete, the method Delete above DataSet control is realized and the record is deleted.

The next succession of designing algorithm for delete in this research is that on the basis of it can be done the procedure DELETE by pressing the key on

the keyboard, what is the usual way for deleting Windows applications.

### Algorithm 6: The delete of records shown in DBGrid pressing DELETE

- Step 1: Was DELETE pressed?
  - Yes:
    - Confirm the delete?
      - Yes: Delete.
      - No: Exit

The procedure which contains the source code written according to this algorithm, should be realized when the key DELETE is pressed. However, when the procedure DBGridKeyPress occurs, Windows doesn't register the press on the key DELETE. This algorithm was coded because of this. Here is the source code:

```

procedure TForm.DBGridKeyUp(Sender: TObject; var
Key: Word; Shift: TShiftState);
begin
  if MessageDlg('Jeste li sigurni da želite
brisanje podataka?', mtConfirmation, [mbYes,
mbNo], 0) = mrYes then
    TDBGrid(Sender).DataSource.DataSet.Delete;
end;

```

The source code for storing the data can be written in procedure DBGridKeyUp, to make one procedure contains the code for storing and deleting the data. However, that's not the good solution because of the reason that OnKeyUp method post is realized after the second pressing the key ENTER, what isn't usual task for saving the data entry. DialogBox with the message acknowledgement for deleting the data like in picture 5.

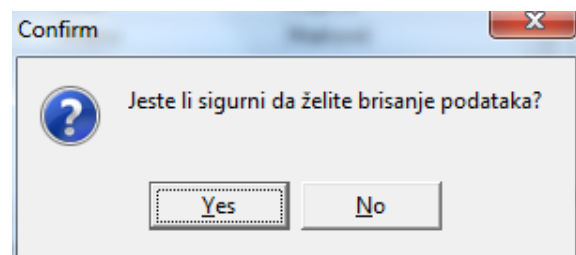


Image 5. DialogBox with message for affirmation of delete

A special attention should be applied to the standard function of key DELETE, which exists during the entry or change of the text. That is used for deleting the typed characters, so in in an algorithm should be examined the status for DataSet. The key DELETE should be used for deleting records from database only when the change or entry

of the data is operated in database, when the status of DataSet control isn't in Edit or Insert.

The algorithm should appear like following:

**Algorithm 7:** *The delete of records shown in DBGrid* pressing the key DELETE when the change or new entry is not realized:

Step 1: Was DELETE pressed?

- 
- **Yes:**
    - Is it out of the status for change or entry?
      - **Yes:**
        - Confirm the delete?
          - **Yes:** Delete.
          - **No:** Exit
- 

The procedure DBGridKeyUp was written on the basis of algorithm 7 and it is:

```
procedure TForm.DBGridKeyUp(Sender: TObject; var
Key: Word; Shift: TShiftState);
begin
  if Key = VK_DELETE then
    if (TDBGrid(Sender).DataSource.DataSet.State
    <> dsEdit) and
      (TDBGrid(Sender).DataSource.DataSet.State
    <> dsInsert) then
      if MessageDlg('Jeste li sigurni da želite
brisanje podataka?', mtConfirmation, [mbYes,
mbNo], 0) = mrYes then
        TDBGrid(Sender).DataSource.DataSet.Delete;
end;
```

The source code of this procedure will operate in every Delphi's project without any change of procedure of OnKeyUp above DBGrid.

## 6. An Algorithm for Sorting the Data in Dbgrid Pressing the Heading of the Column

In Windows applications that give the table display of the data, the value sorting from particular table is realized by clicking on the heading of the column. The sorting is realized in a growing chain with the first click, then the next click sorts the declining chain.

Standardly, Delphi's control of DBGrid doesn't contain this property. Common algorithm, according to the source code for sorting the data in DBGrid can be written, and is based on the questions if the declining chain has already been sorted. If it is, then the sorting in declining chain is realized. An algorithm for sorting the data in DBGrid according to the values in the columns from one chain, which compares and sorts in the terms of values, is in Delphi's method Sort which is realized at DataSet control:

### Algorithm 8: Sorting the data DBGrid

- Step 1: Is sorting different from declining string?
  - **Yes:** Sort in a declining string.
  - **No:** Sort in a growing string.

For this algorithm in the source code is used term:

*TCustomADODataset(TColumn(Column).Grid.DataSource.DataSet)* to come via argument a *Column* to DataSet control.

```
procedure TForm.DBGridTitleClick(Column:
TColumn);
begin
  with
    TCustomADODataset(TColumn(Column).Grid.
    DataSource.DataSet) do
  begin
    if (Pos(Column.FieldName, Sort) = 1)
    and (Pos(' DESC', Sort) = 0) then
      Sort := Column.FieldName + ' DESC'
    else
      Sort := Column.FieldName + ' ASC';
    end;
  end;
```

The command for sorting the data in DBGrid is composed from two strings: the name of the field adjoined to the column field, where the sorting is realized as the code of the sorting, in a growing string (ASC) and declining (DESC). In the source code with the command **Pos** is examined the regulariry of the string by examining if there from the position 1 begins the field name (Pos(Column.FieldName, Sort)=1) and if string i ' DESC' is absent from string of sorting (Pos(' DESC', Sort)= 0).

The windows display will be shown with the sorted data about students with the index number and growing and declining string:

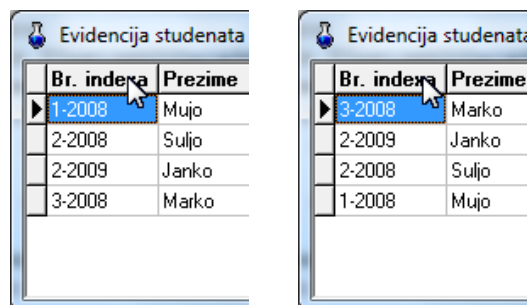


Image 6. Display of data in DBGrid soerted in a growing string (left) and decling string (right)

We can say for this procedure, that it will operate in any Delphi's project without any changes within the task OnTitleClick at DBGridom.

## 7. Conclusion

While realizing many Delphi's projects which represented unique information systems for the need of small companies, as examples for teaching the students, it proved that such projects can be realized easily with the use of ADO controls for connecting to the database and Data control of DBGrid and DBNavigator. The operation of the basic functions of DBNavigator is very often realized with the use of menu or toolbar, and DBNavigator is omitted.

The procedures described in this research were achieved that the basic functions of DBNavigator are realized in the way which is the most acceptable by the users of information systems, and it is by pressing the key on the keyboard and with the use of the mouse click on the heading of DBNavigator control. That makes the information systems more user friendly.

The simplicity and modularity of the source code from this research enables the Delphi's beginners to program and develop their projects related to what is achieved with the Delphi's standard controls for operating the database.

Concerned that the source code of the procedures described in this research was based on the modularity, the future work which can be based on this work is creating Plugin. That can be addition with already installed Delphi's applications, which contain DBGrid controls. This enables the two basic functions described in this work stay in the most usual way.

## Reference

- [1]. Cormen T., Leiserson C., Rivest R., Stein C., *Introduction to Algorithms*, Cambridge, Massachusetts, London, 2002.
- [2]. Sukic C., Sukic E., *Informatika i računarstvo*, Graficolor, Kraljevo, 2010.
- [3]. Cantu M., *Delphi 7*, Kompjuter biblioteka, Čačak, 2003
- [4]. Bekim Fetaji, *Development and Analyses of Dynamical Visualization Process Tool in Run Time and its Usability Evaluation*, TTEM Technics Technologies Education Management, Journal ISSN 1840-1503, paper accepted for Vol. 6, No.2, 2011.
- [5]. Sukić Ć., Saračević M. (2012), *UML and JAVA as effective tools for implementing algorithms in computer graphics*, TEM JOURNAL - Technology, Education, Management, Informatics, Journal ISSN:2217-8309, e-ISSN:2217-8333, Vol. 1, No.2, 2012, pp.111-117.
- [6]. Mahfoozi A., *Improving Delphi TDBGrid*, CPOL, (web link: <http://www.codeproject.com/Articles/>), 2011.
- [7]. Pei-rong W., *The Development of Data Browsing Control Based on DBGrid*, Computer Knowledge and Technology, vol.35, 2008.
- [8]. *Delphi Knowledge Base* (2011), Delphi portal, link:<http://delphi-kb.blogspot.com/2011/04/how-to-sort-columns-in-tdbgrid.html>
- [9]. Gajic Z., *Sorting records in Delphi DBGrid by Clicking on Column Title* (2003), Delphi portal, link: <http://delphi.about.com/>

*Corresponding author:* Bekim Fetaji  
*Institution:* South East European University,  
Contemporary Sciences and Technologies, Tetovo,  
Macedonia  
*E-mail:* [b.fetaji@seeu.edu.mk](mailto:b.fetaji@seeu.edu.mk)