Raised Graduation Information Technology Solutions for the Visual Interface

László Gugolya[*]
[*]Óbuda University AMK, Székesfehérvár, Hungary
gugolya.laszlo@amk.uni-obuda.hu

Abstract— **In Hungary, the information technology subject teaching scarce resources to cultivate. The number of contact lessens are low, so teachers should think about a solution by which, home practicing is prioritized. The active programming tasks usually highly motivate students. Responsible teachers must teach students the basics of informatics. This way students will recognize if they are interested in information technology. Currently, the graduation information technology system of general education subjects requires deep comprehensive knowledge of programming. Comprehensive test exercises are being developed in such a way that a wide variety of programming languages can be applied for finding the solution. This principle leads to a situation where the tasks are basically console applications, which in today's world do not represent a strong motivation for the majority of students. According to the facts mentioned before, we examined whether it is worth teaching visual programming in high school graduation for some particular tasks.**

I. Introduction

For several years, I have been participating in the implementation of advanced information technology exams. The participatory element was diverse as well as an organizer, as an examiner, oral teacher involved in the process.

The final graduation system had been introduced in 2005, was an important point in the programming part of the release .Tasks are basically accomplished by applying elementary programming theorems and usually completed by text file handling. Solving the problems you can use multiple programming languages, which you selected. The list of programming languages is constantly changing. Because of the changes and differences in language tasks are developed in such a way that the console interface: they should be solved without any input controls.

Several arguments can be made either against or pro console application. To tell the truth, the console application motivational point of view is not the best choice. The question is therefore raised: what kind of results should be reached in teaching visual interfaces. What are the disadvantages, what are the benefits if the problems are solved by choosing visual programming methods and tools. This topics is being discussed in the followings.

A. Opportunities

The option of choosing programming languages changed from year to year, but not fundamentally. The most significant change is the disappearance of Borland's products from the list. This fact, however, does not affect teaching Pascal due to the appearance of Lazarus. In the case of C ++, however, the only choice is Visual Studio C ++ (Visual programming point of view). In Linux environment, Eclipse is used instead of NetBeans. In the case of C++ programming language we use QtCreator.

The list shows that the visual interface when creating the Lazarus / Pascal, Java (NetBeans / Eclipse), Visual Studio (C / C ++, Basic, C #) selection is possible. The possible systems are shown in the figures below.

FreePascal 2.6.4
FreePascal 2.6.4 + Geany 1.25
Lazarus 1.4.2 (fpc 2.6.4)
Code::Blocks 13.12 MinGW/GCC (4.8.1)
MinGW/TDM-GCC 4.8.1 + Geany 1.25
Python 3.4.3 + IDLE (Python GUI)
MS Visual Studio 2013 Express for Windows Desktop (Visual Basic, Visual C#, Visual C++)
JAVA SE – NetBeans 8.0.2, JDK 8

Figure 1. MS Windows systems, radio programming environments (2016)

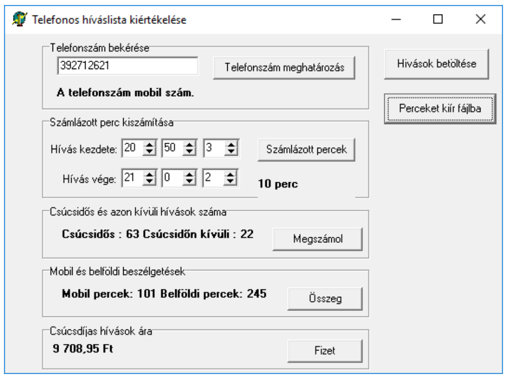
II. Case Study

A. Official solutions with visual tools

The exam took place 17 to 32 have a formal arrangement pascal or derivatives language was issued. There are two solutions for Visual Basic, C # solution for five and eight were between C ++ solution.

It is located between an official of the solutions, which have been among the visual interface. Between the first two series of tasks we find written in Delphi solution, and then after a long pause in 2012 by Lazarus (Delphi clone) was the official sample solution. These solutions can also be seen how could allow the visual interface.

When you open and save dialogs when files are used so that you can avoid common errors in file open. The task


Telefonos híváslista kiértékelése

Telefonszám bekérése: 392712621
A telefonszám mobil szám.

Számlázott perc kiszámítása
Hívás kezdete: 20 50 3
Hívás vége: 21 0 2
10 perc

Csúcsidős és azon kívüli hívások száma
Csúcsidős : 63 Csúcsidőn kívüli : 22

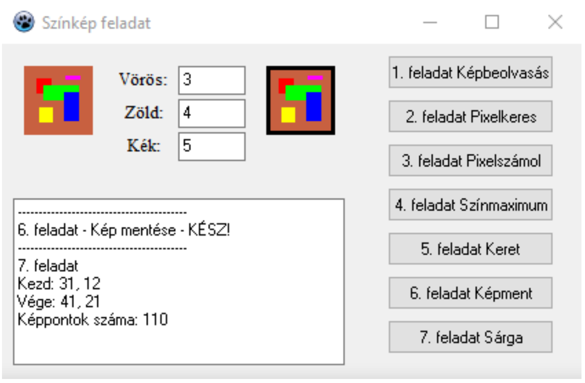
Mobil és belföldi beszélgetések
Mobil percek: 101 Belföldi percek: 245

Csúcsdíjas hívások ára
9 708,95 Ft

Figure 2. Official Solution - February 2006 (Delphi)

management with a list of phone calls show that the subpoenas sought to be made more user-friendly. This SpinEdit were used for the appropriate restrictions. In this case, the DateTimePicker also a good choice, it also allows a controlled input. The other task has seen limited use, because the tasks are executed by pressing buttons, and the results appear in-TLabel.

For color images exercises in autumn 2012 in the screenshot shows the order of the task. It is also seen that here were used to solve particular tasks buttons. The Open, Save dialog windows are also resolved. The results of the multi-line text component (TMemo) appear. Request for information on the limited to simple text input.



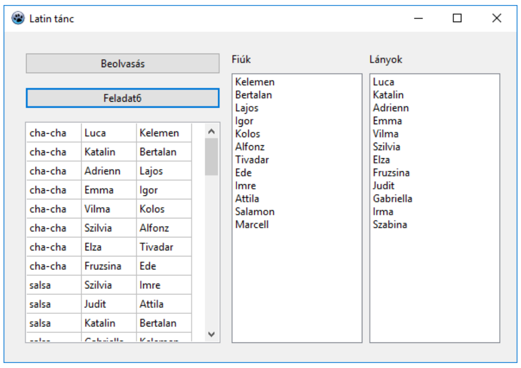
```
------------------------------------
6. feladat - Kép mentése - KÉSZ!
------------------------------------
7. feladat
Kezd: 31, 12
Vége: 41, 21
Képpontok száma: 110
```

Figure 3. Official Solution - October 2012 (Lazarus)

You can see that in the event of a formal solutions are also displayed on the visual interface. It is quite one-sided, because only with the great Hungarian tradition Pascal family of solutions can be found. Also, check out other options as well, and also to consider whether it still lies in whether other options for Pascal/Lazarus.

B. Lazarus/Free Pascal (Delphi)

Unlike other language in the Pascal language difficulties managing a dynamic array[4]. The problem solving this is not necessary, because it would always provide the maximum number of elements, but understandably convenient programming data to the list-like structure. This can be overcome by using the indicators, but experience shows that this understanding of time-consuming for students. If simple. say consisting of text data list, you need to be a TComboBox or TListBox replace elements of a dynamic list. Here, the elements are stored in the Items property of which TStringList[1]. Thus, it is no use visual interface. A good example is the May 2015 maturity and a foreign language (Latin dance)



cha-cha	Luca	Kelemen
cha-cha	Katalin	Bertalan
cha-cha	Adrienn	Lajos
cha-cha	Emma	Igor
cha-cha	Vilma	Kolos
cha-cha	Szilvia	Alfonz
cha-cha	Elza	Tivadar
cha-cha	Fruzsina	Ede
salsa	Szilvia	Imre
salsa	Judit	Attila
salsa	Katalin	Bertalan

Fiúk	Lányok
Kelemen	Luca
Bertalan	Katalin
Lajos	Adrienn
Igor	Emma
Kolos	Vilma
Alfonz	Szilvia
Tivadar	Elza
Ede	Fruzsina
Imre	Judit
Attila	Gabriella
Salamon	Irma
Marcell	Szabina

Figure 4. Latin Dances – TStringGrid and TListBox

exercises. In the sixth task I had to collect the girls and the boys' list. The listbox handling is easy to do.

```pascal
for  i := 0 to Length(Adat)-1 do
begin
  if lbxFiu.Items.IndexOf(Adat[i].fiu)=-1
   then lbxFiu.Items.Add(Adat[i].fiu);
end;
lbxFiu.Items.SaveToFile('.\\szerepek.txt');
```

This example demonstrates that you can easily save file, although it is true that it is not formatted export permits. In such cases, the conventional iterative design portfolio.

In another example, an overview of the stock is being scanned. This should be the exercises in autumn 2007, where you had to process the results of a major football championship track. Needed to store data structure is as follows.

```pascal
type
     TEgyMeccs = record
        Fordulo : Byte;
        hGol, vGol : Byte;
        hfGol, vfGol : Byte;
        Hazai, Vendeg : String[20];
      end;
      TMeccsTomb = array of TEgyMeccs;
var
      Meccs : TMeccsTomb;
```

The length of the array used to store the results of the scan is determined (setLength). Here, too, it used a dynamic list. You can upload a file directly. In this example, the data displayed in a Grid. The filling can be done directly using TStringList[2] per line. The data records disassembling the appropriate fields in the Pascal "new" elements of language (delimiter) is possible.

```pascal
 var TextFile, sor : TStringList;
   i :integer;
 begin
   TextFile := TStringList.Create;
   TextFile.LoadFromFile('.\\meccs.txt');
   Sor := TStringList.Create;
   sgrdAdatok.RowCount := TextFile.Count;
   SetLength(Meccs,TextFile.Count);
   for i := 1 to TextFile.Count-1 do
   begin
     Sor.Clear;
     Sor.StrictDelimiter:=true;
     Sor.Delimiter:=' ';
     Sor.DelimitedText:= TextFile[i];
     sgrdAdatok.Rows[i] := Sor;
     with Meccs[i] do
     begin
       Fordulo:= StrToInt(Sor[0]);
       hGol:= StrToInt(Sor[1]);
       vGol:= StrToInt(Sor[2]);
       hfGol:=StrToInt(Sor[3]);
       vfGol:= StrToInt(Sor[4]);
       Hazai:= Sor[5];
       Vendeg:= Sor[6];
     end;
   end;
 end;
```

TListBox in the display, chasing solution is useful in the task 7, where the number of results occurring had to be determined (indexOf, Add). The handling of files (save, load) can be easily solved by using dialog windows and grid.

```
//TOpenDialog
if Open.Execute then
  begin
grd.LoadFromCSVFile(Open.FileName,' ',false)
    grd.DeleteRow(1);
  end;
//TSaveDialog
if Save.Execute then
grd.SaveToCSVFile(Save.FileName,' ',false);
```

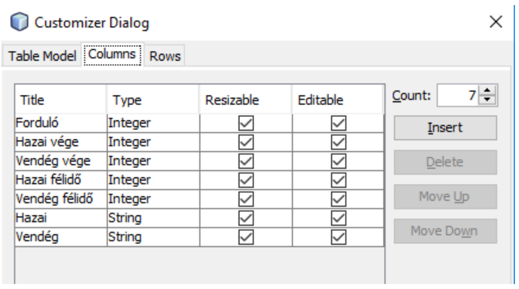
C. Netbeans/Java

Java visual editor interface provides the least help. Under MS Windows optional NetBeans provides more opportunities than WindowBuilder under Eclipse. Let's look at the previous "soccer" task. First, we are prepared to class, which allows you to store can be a dynamic array (ArrayList)[7].

```
public class Meccs {
  private int fordulo;
  private int hazaiVege;
  private int vendegVege;
  private int hazaiFelido;
  private int vendegFelido;
  private String hazaiNeve;
  private String vendegNeve;

    Meccs(String sorMeccs){
    String [] data=sorMeccs.split(" ");
    fordulo=Integer.parseInt(data[0]);
    hazaiVege=Integer.parseInt(data[1]);
    vendegVege=Integer.parseInt(data[2]);
    hazaiFelido=Integer.parseInt(data[3]);
    vendegFelido=Integer.parseInt(data[4]);
    hazaiNeve= data[5];
    vendegNeve= data[6];
    }
    }
```

A few clicks prepared from the get / set methods for the data fields. The data were also done in a table (JTable) imaging help. NetBeans provides an opportunity for entering properties.



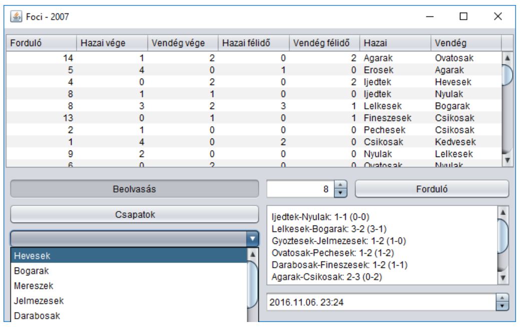
Title	Type	Resizable	Editable
Forduló	Integer	✓	✓
Hazai vége	Integer	✓	✓
Vendég vége	Integer	✓	✓
Hazai félidő	Integer	✓	✓
Vendég félidő	Integer	✓	✓
Hazai	String	✓	✓
Vendég	String	✓	✓

Figure 5. JTable edit - NetBeans

Then the scan result.

```
be = new BufferedReader(
                new FileReader("meccs.txt"));
String line;
line = be.readLine();
while((line = be.readLine()) != null){
   t.addRow(line.split(" "));
   mind.add(new Meccs(line));
}
be.close();
```

Like Lazarus to use subpoenas to the controlled JCombobox or JSpinner. The date is difficult deems visual equipment. The JSpinner be used for this, but this is cumbersome compared to other devices. So far, the images on screen.



Forduló	Hazai vége	Vendég vége	Hazai félidő	Vendég félidő	Hazai	Vendég
14	1	2	0	2	Agarak	Ovatosak
5	4	0	1	0	Erosek	Agarak
4	0	2	0	2	Ijedtek	Hevesek
8	1	1	0	0	Ijedtek	Nyulak
8	3	2	3	1	Lelkesek	Bogarak
13	0	1	0	1	Fineszesek	Csikosak
2	1	0	0	0	Pechesek	Csikosak
1	4	0	2	0	Csikosak	Kedvesek
9	2	0	0	0	Nyulak	Lelkesek
6	0	2	0	0	Ovatosak	Nyulak

```
Ijedtek-Nyulak: 1-1 (0-0)
Lelkesek-Bogarak: 3-2 (3-1)
Gyoztesek-Jelmezesek: 1-2 (1-0)
Ovatosak-Pechesek: 1-2 (1-2)
Darabosak-Fineszesek: 1-2 (1-1)
Agarak-Csikosak: 2-3 (0-2)
```

2016.11.06. 23:24

Figure 6. JTable - NetBeans

D. Visual Studio – C#

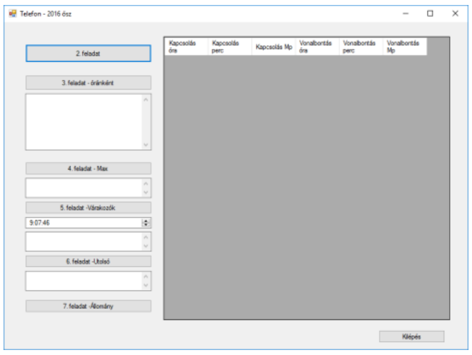
The graduation-use visual interfaces are a great solution for MS Visual Studio. Here you can choose from several languages the candidate. The visual elements of the language in terms of secondary consideration. Currently, the high school in the C # language taught in information technology teachers, so we can consider it. controls used in the other two instruments are available here as well[5]. The Java with the opposite is also possible to specify in structure (struct)[6], so as to avoid the introduction of the class as well. The user-friendly data entry can be easily achieved using the available components. The numbers are easy to use to request the NumericUpDown. The list for treatment, prompted the ListBox, ComboBox. Display the data in a TextBox, Label used. The date, time by asking, for managing the MonthCalendar, DateTimePicker. Data scan and display the DataGrigView, OpenFileDialog, SaveFileDialog use.

The suggests that the previous visual elements are preferred in certain cases. But it is worth to consider whether, overall, all problem what we experience. So let's look at an example of the recent series of problems.

E. Current graduation exercises – C#

The exercises associated with a small company call center[4]. The exact exercises found in the www.oktatas.hu side. Here are just highlight the essential parts. In solving the problems of the keys used, these may be selected from the menu instead of treatment. The solution, the official solution we used with minor changes. The official solution for C ++ have contributed. The solution is based on our screenshot below.

The first problem was to create a function which later

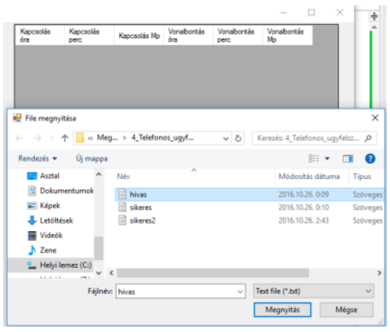


Kapcsolás óra	Kapcsolás perc	Kapcsolás Mp	Vonalbontás óra	Vonalbontás perc	Vonalbontás Mp

Figure 7. October 2016 graduation – MS Visual Studio

had to be used. So it did not require visuals.

The second problem is to read the text file. The scan time to avoid errors in the OpenFileDialogot used, consisting of data structure read the information. The scan data DataGridView visualized. The Grid and OpenFileDialog properties customized to fit the job.

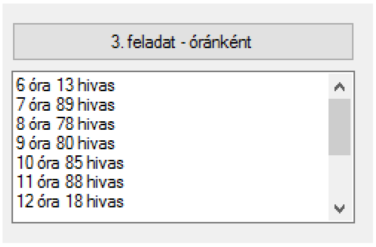


Név	Módosítás dátuma	Típus
hivas	2016.10.26. 0:09	Szöveges
sikeres	2016.10.26. 0:10	Szöveges
sikeres2	2016.10.26. 2:43	Szöveges

Fájlnév: hivas
Text file (*.txt)

Figure 8. 2. problem – MS Visual Studio

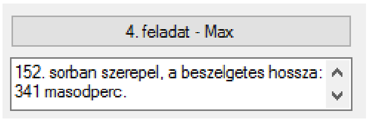
During the third problem to be defined, where the number of outputs per hour calls. Here is a multiline TextBox was used on the dump that we comply with the description.



```
6 óra 13 hivas
7 óra 89 hivas
8 óra 78 hivas
9 óra 80 hivas
10 óra 85 hivas
11 óra 88 hivas
12 óra 18 hivas
```

Figure 9. 3. problem – MS Visual Studio

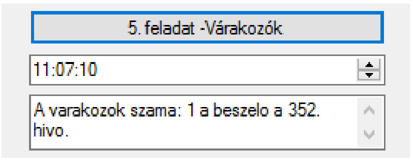
The fourth problem is to determine the longest call has been made. The task of simplifying asked since it was sufficient to show one match.



```
152. sorban szerepel, a beszelgetes hossza:
341 masodperc.
```

Figure 10. 4. problem – MS Visual Studio

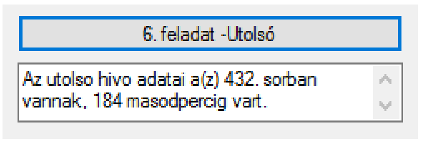
The fifth problem was a time in requesting the task of working hours, and this had to do a search. Here, the visual prompted nice body could help to customize the DateTimePicker. The result is displayed formatted (String.Format) TextBox also-ran solved.



```
11:07:10
A varakozok szama: 1 a beszelo a 352.
hivo.
```

Figure 11. 5. problem – MS Visual Studio

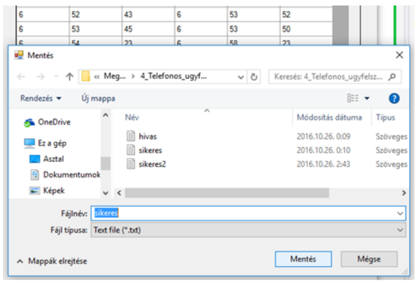
The sixth problem was to define the last details of the conversation. Here again, it was enough to dump a formatted text box.



```
Az utolso hivo adatai a(z) 432. sorban
vannak, 184 masodpercig vart.
```

Figure 12. 6. problem – MS Visual Studio

The seventh problem was to make stock, whatever it was the list of successful calls in that format. It's SaveFileDialog was used to correct for the job property settings.



6	52	43	6	53	52
6	53	45	6	53	50
6	54	23	6	58	23

Név	Módosítás dátuma	Típus
hivas	2016.10.26. 0:09	Szöveges
sikeres	2016.10.26. 0:10	Szöveges
sikeres2	2016.10.26. 2:43	Szöveges

Fájlnév: sikeres
Fájl típusa: Text file (*.txt)

Figure 13. 7. problem – MS Visual Studio

As a conclusion, we found the followings for the case of the comprehensive exams in October 2016. Applying visual interfaces helped students only in one problem for input requests and file handling.

SUMMARY

During preparing this series of problems we got to the conclusion that applying visual programming interface does not support the solution very much. Just a few problems are simplified due to the easier input request, but this control is not part of maturity, so it is also an "excess benefit". Based on our experience in teaching visual interfaces it takes extra time to learn in the current conditions of education. Based on the experience of advanced programming education curriculum it takes two hours a week the school year, when the console interface is taught. Learning visual interfaces need at least three hours a week.

The visual interface teaching is definitely an advantage in the future to help students getting much more motivated towards using a more modern programming interface. They are commonly making more experiments and programming at home. Object-based thinking is being developed, which is also useful when dealing with information technology students in the future.

REFERENCES

[1] http://www.freepascal.org/docs-html/rtl/classes/tstringlist.html.

[2] http://wiki.freepascal.org/TStringList-TStrings_Tutorial.

[3] http://www.oktatas.hu/kozneveles/erettsegi/feladatsorok/emelt_szint_2016osz

[4] Benkő Tiborné - Tóth Bertalan, *Együtt könnyebb a programozás - Free Pascal,* Conputerbooks, 2010.

[5] Benkő Tiborné - Tóth Bertalan, *Együtt könnyebb a programozás – C#,* Conputerbooks, 2011.

[6] Reiter István, *C# programozás lépésről lépésre,* Jedlik Oktatási Studio Kft., 2016

[7] Végh Csaba, Dr, Juhász István, *Java-start!* Logos Kiadó, 2000

,