



# Ferramenta conversora de interfaces gráficas DELPHI2JAVA-II

Fabricio Fonseca

Mauro M. Mattos



# Introdução

- Utilizada para converter formulários Delphi para código Java;
- Inicialmente sugerida para ser utilizada nas aulas introdutórias de programação OO;
- Fundamentada na questão da migração de softwares.



# Objetivos

- Gerar código Java a partir de formulários desenvolvidos em Delphi;
- Implementar a conversão de um subconjunto de componentes de interface;
- Disponibilizar no código Java alguns dos eventos mais utilizados nas aplicações.



# Fundamentação

- Os softwares envelhecem diante das novas plataformas de desenvolvimento;
- Microsoft .Net e J2EE apresentam-se como opções para evitar a defasagem;
- Neste contexto tem-se o dilema:  
**"Manter o sistema? Reescrever? Migrar?"**



# Manter o sistema?

- Os legados não conversam com novos sistemas;
- Não aproveitam a infra-estrutura de comunicação web;
- Alguns foram desenvolvidos em linguagens de programação já em desuso.



# Reescrever?

- Exige todo o investimento original, o qual muitas vezes "foge" dos orçamentos das empresas;
- Consome muito tempo, sendo que a vida útil do sistema é cada vez menor.



# Migrar?

- Exige cerca de 25% do tempo de desenvolvimento original;
- Custos representam entre 15% a 20% em relação ao custo original;
- A migração pode converter também as regras de negócios embutidas no software original.



# Delphi2Java

- Versão trial localizada durante o desenvolvimento do projeto;
- Ferramenta com propósito semelhante ao descrito neste trabalho;
- Constatada a não continuidade da ferramenta.





# Requisitos

- Arquivos convertidos devem preservar as características originais;
- Duas saída de dados - visual e física;
- Gerar duas classes - interface e eventos;
- Seleção de vários arquivos para conversão;
- Fácil operacionalidade.



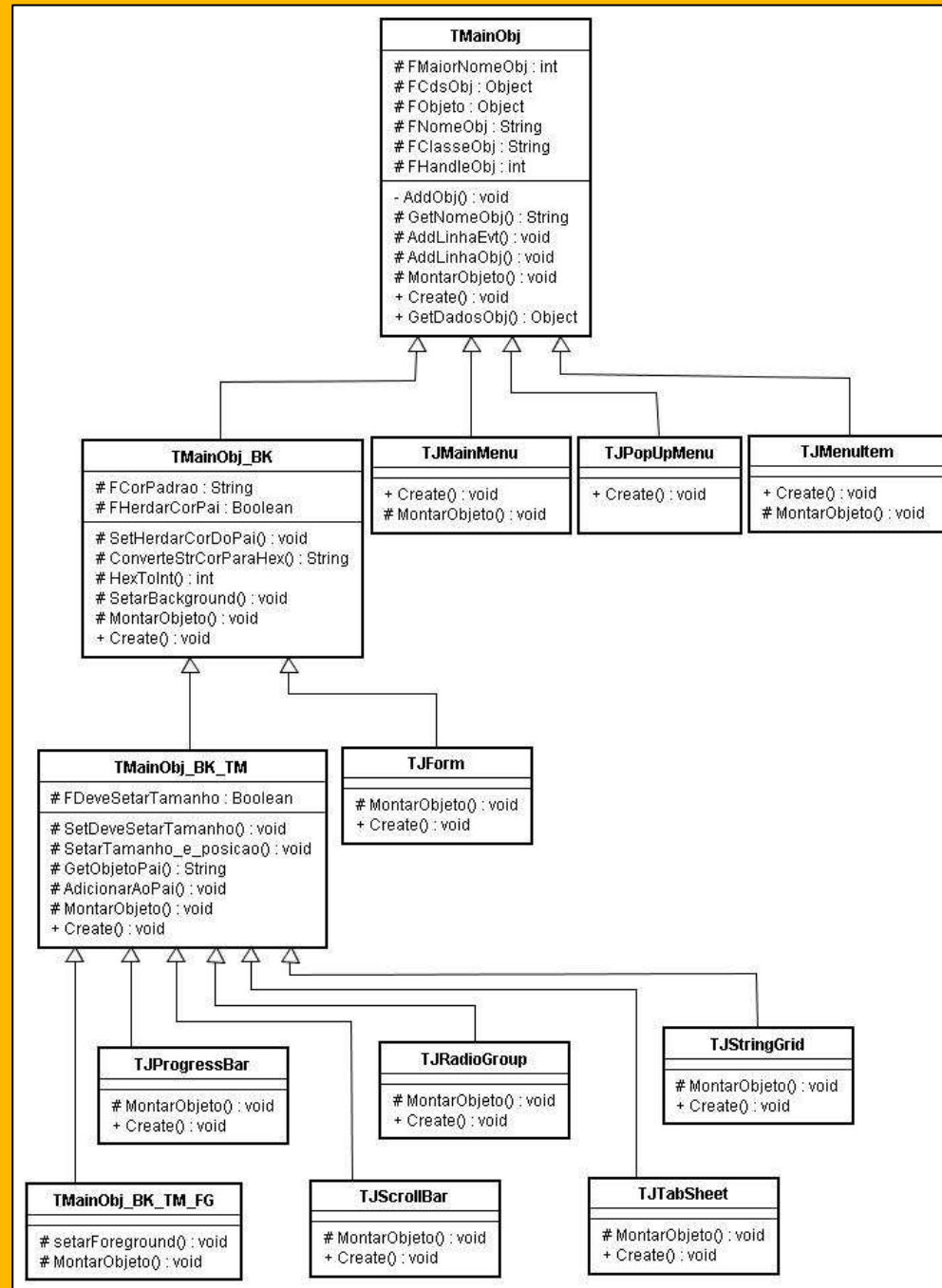
# Especificação

O sistema possui cinco classes principais:

- TMainObj;
- TMainObj\_BK;
- TMainObj\_BK\_TM;
- TMainObj\_BK\_TM\_FG;
- TMainObj\_BK\_TM\_FG\_CPT;

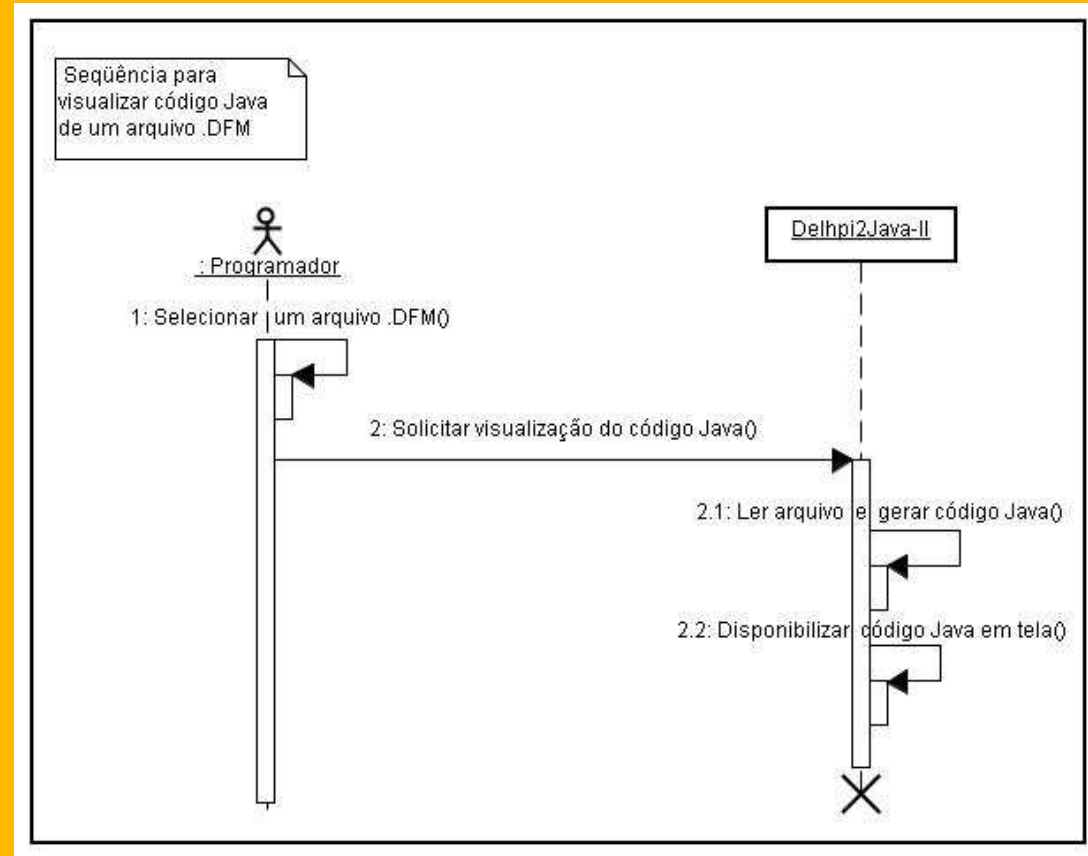


# Diagrama de classes



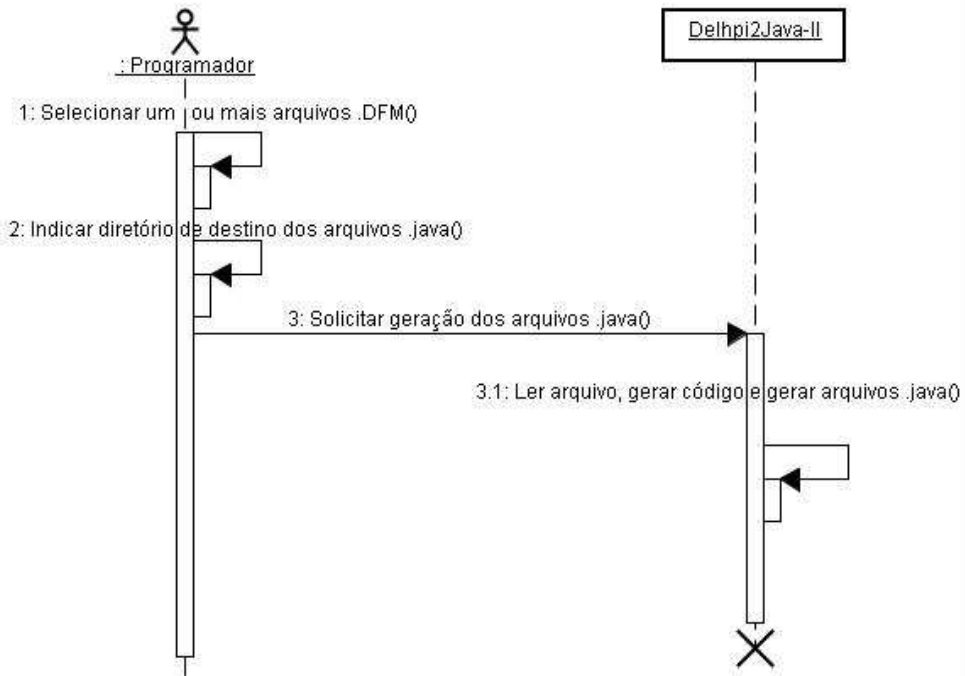


# Diagrama de seqüência



# Diagrama de seqüência

Seqüência para gerar arquivos .java, selecionando-se um ou mais arquivos .DFM



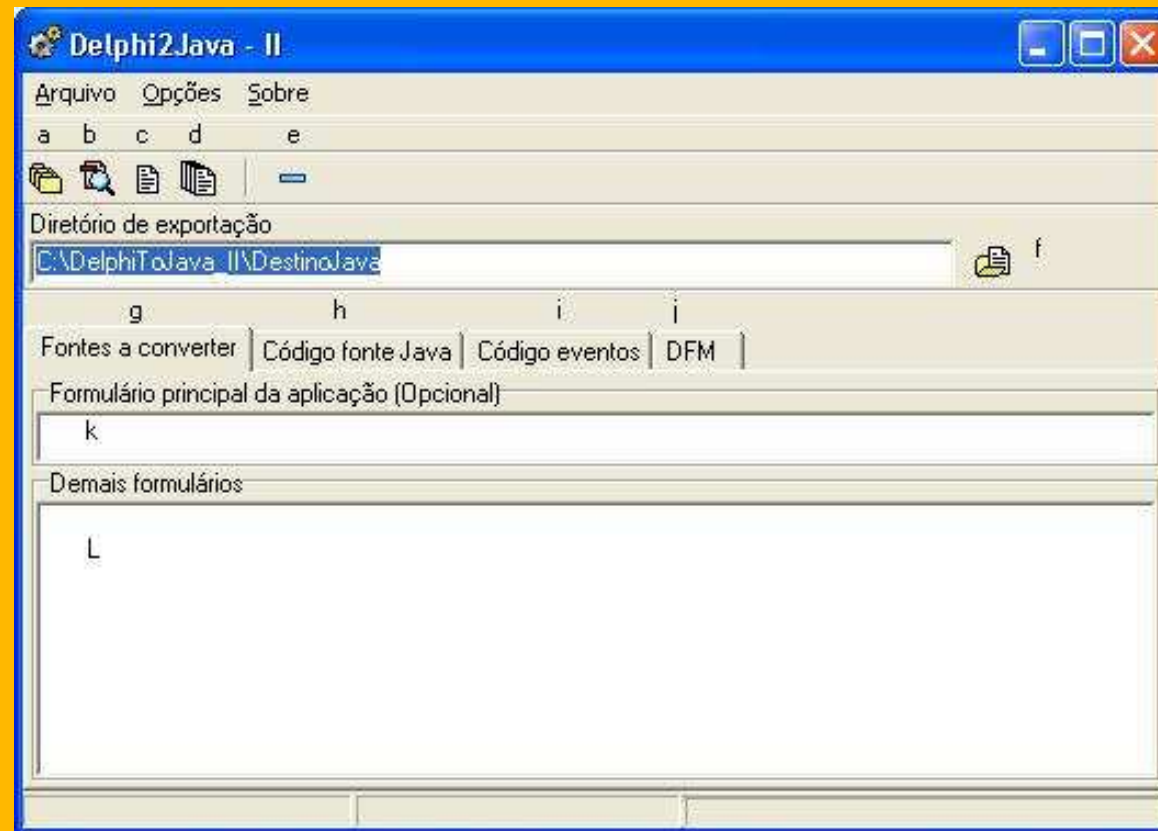
# Implementação

## Técnicas/Ferramentas

- Desenvolvida OO;
- Implementada no ambiente Borland Delphi 7;
- Testes realizados na ferramenta JCreator LE 3.1;

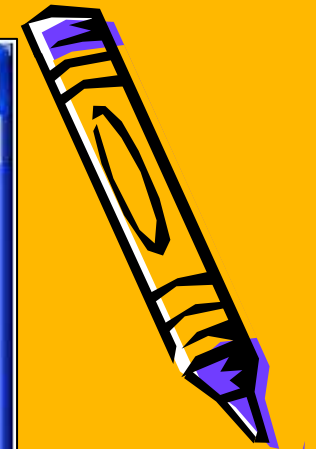
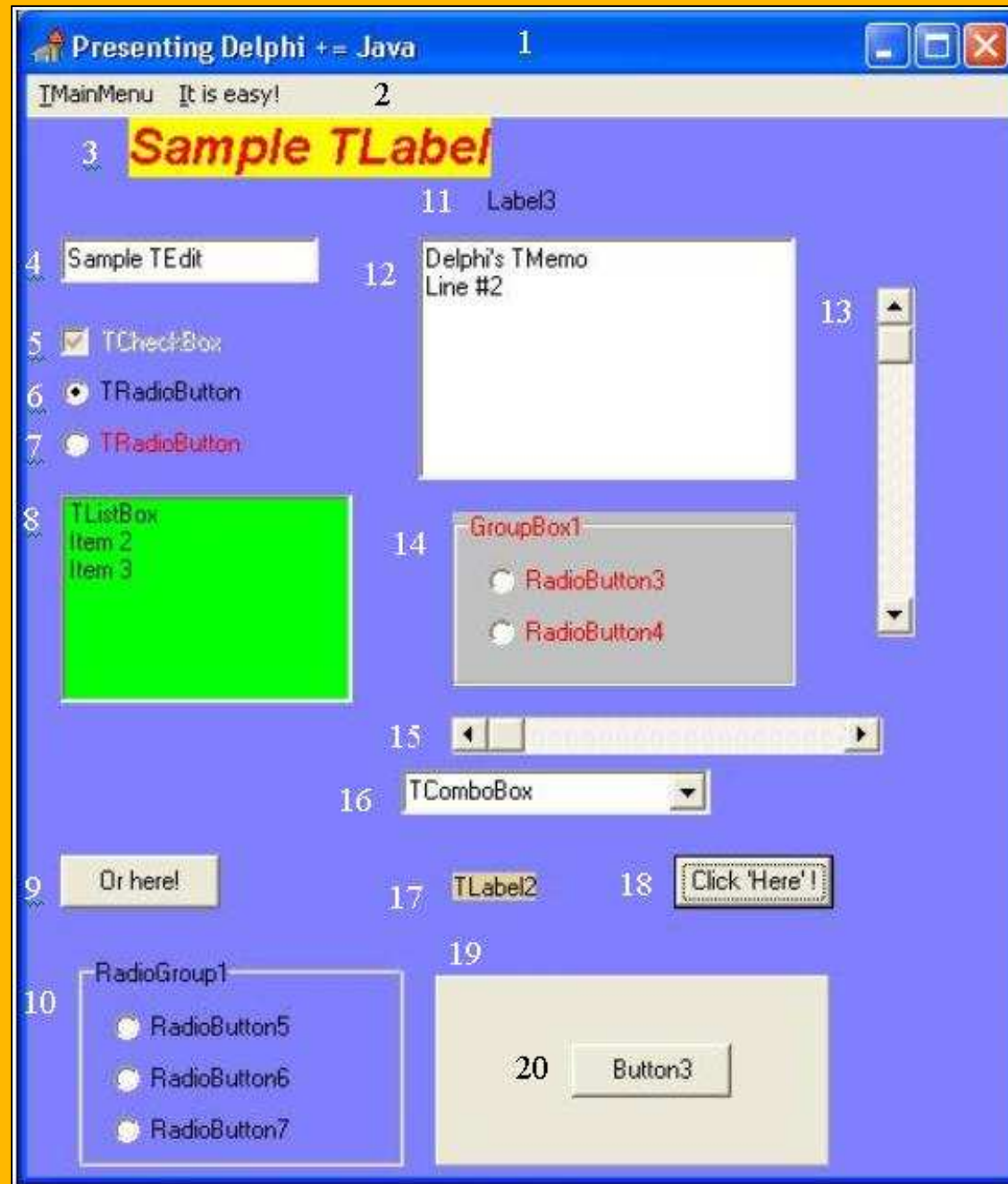


# Implementação Operacionalidade

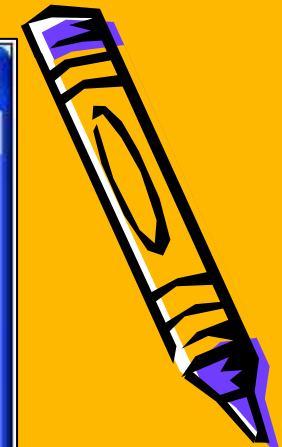
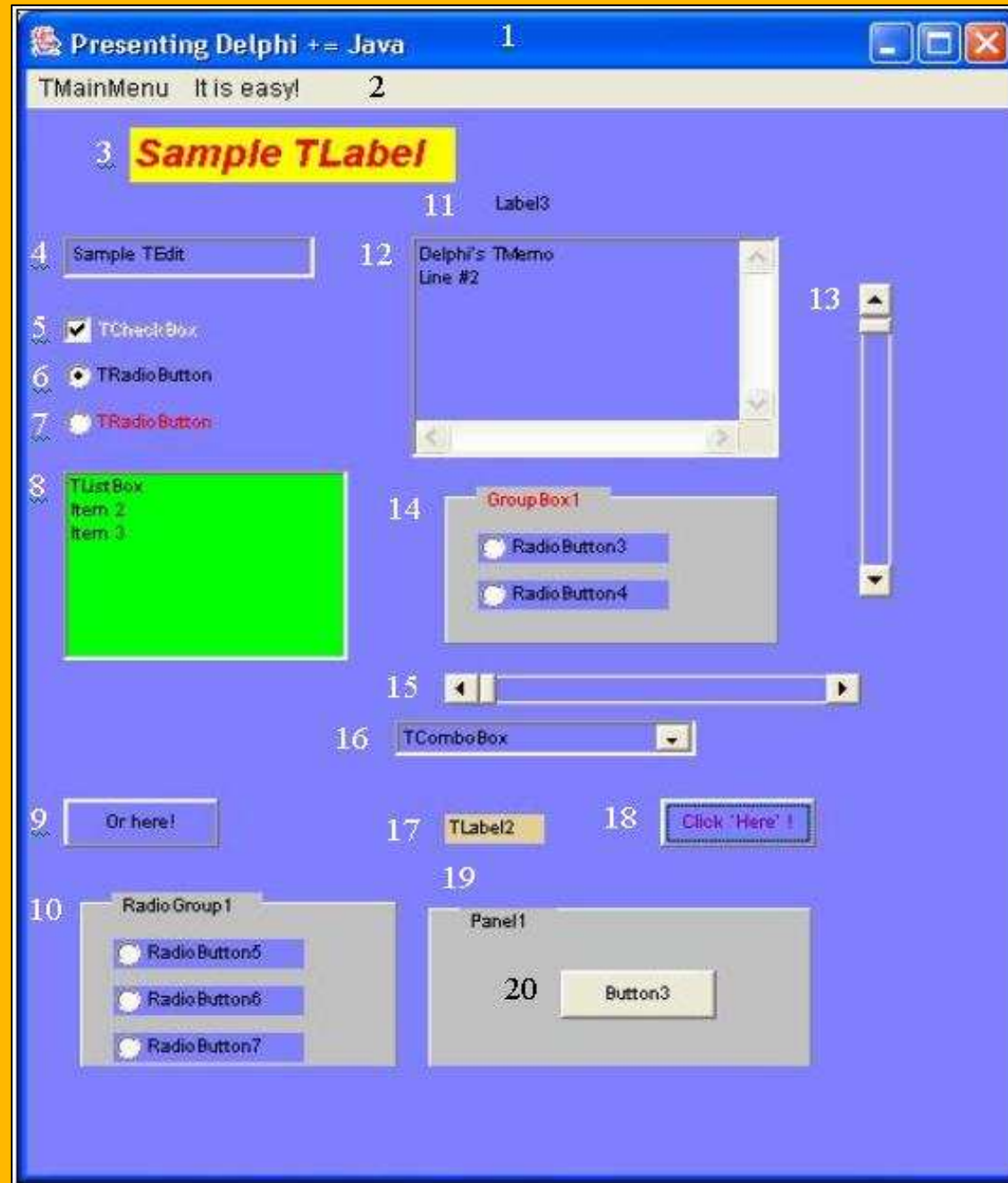




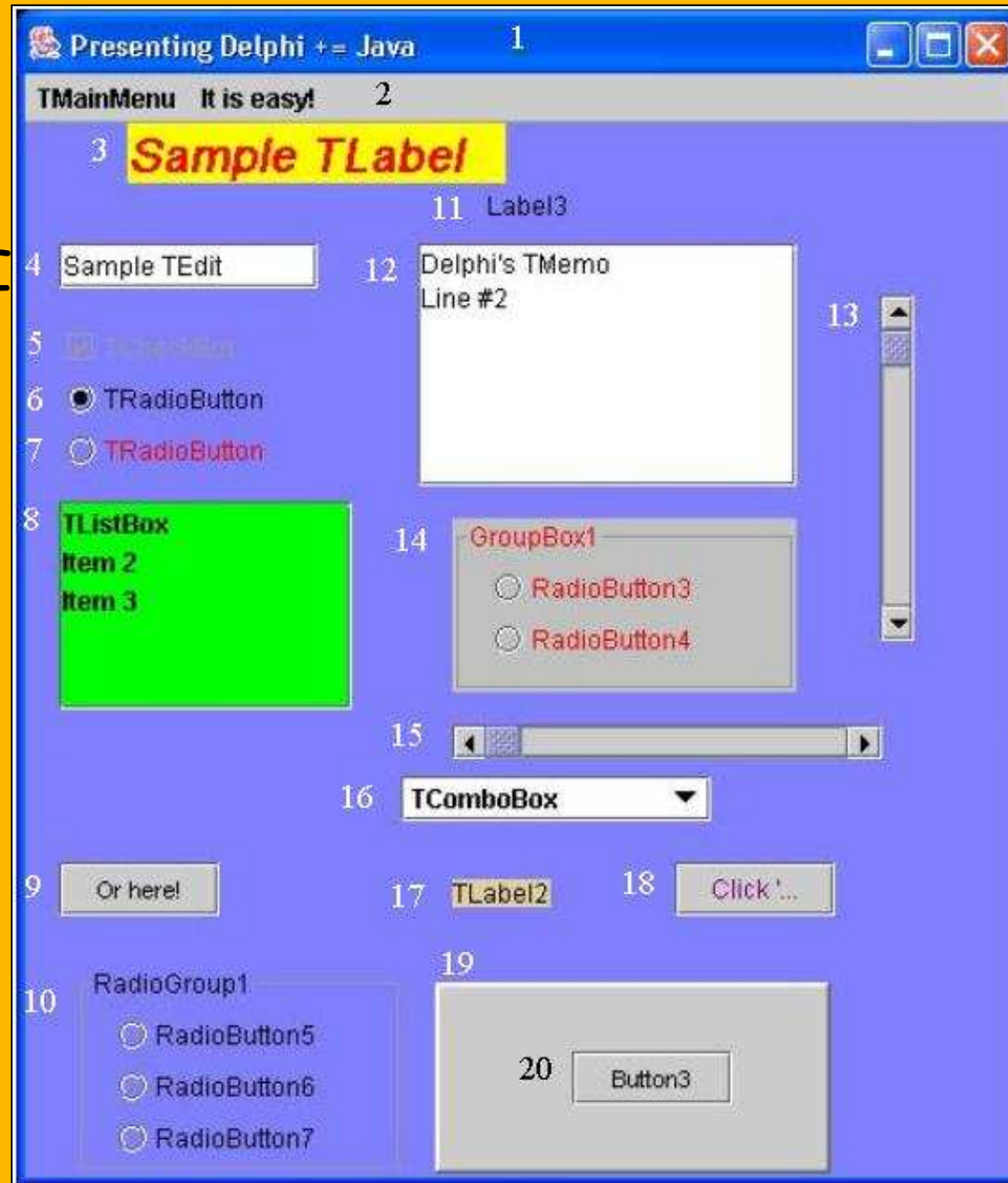
# Formulário Delphi



Convertido  
por  
Delphi2Java



Convertido  
por  
Delphi2Java-II



# Código Java

## Delphi2Java

```
1
2 import java.awt.*;
3 import java.io.*;
4 import java.lang.*;
5 import java.util.*;
6 import d2j.dlabel;
7 import d2j.dprop;
8 import d2j.showmsg;
9 import d2j.showmsg2;
10 import d2j.dutil;
11 import d2j.bitbtn;
12 import d2j.dstr;
13 import d2j.dstrlist;
14 import d2j.dcanvas;
15 import d2j.dvars;
16
17
18 class tform1 extends Frame implements Runnable
19 {
20     Insets finsets;
21     int finsets2_top = 0;
22     int finsets2_left = 0;
23     boolean FLayout = false;
24     boolean ConsDone = false;
25     boolean InPaint = false;
26     dcanvas canvas = new dcanvas(this);
27     dutil DUTIL = new dutil();
28     public Label label1;
29     public Button button1;
30     public TextField edit1;
31     public TextArea memol;
32     public Checkbox checkbox1;
33     public Checkbox radiobutton1;
```



# Código Java

## Delphi2Java-II



```
1 public class TForm1FRM extends javax.swing.JFrame {
2
3     public TForm1FRM() {
4         initComponents();
5         setBounds(341, 218, 471, 548 ); //define o tamanho do formulário
6     }
7
8     private void initComponents() {
9
10        //
11        // setar o título da janela
12        setTitle("Presenting Delphi += Java");
13    };
14    getAccessibleContext().setAccessibleName("TForm1FRM Frame");
15
16    Form1 = new javax.swing.JDesktopPane();
17    Label1 = new javax.swing.JLabel();
18    Label2 = new javax.swing.JLabel();
19    Label3 = new javax.swing.JLabel();
20    Button1 = new javax.swing.JButton();
21    Edit1 = new javax.swing.JFormattedTextField();
22    Mem1 = new javax.swing.JTextArea();
23    ScrPnMem1 = new javax.swing.JScrollPane();
24    CheckBox1 = new javax.swing.JCheckBox();
25    GRPForm1 = new javax.swing.ButtonGroup();
26
```



# Código Java eventos

## Delphi2Java-II



```
1 public class TForm1 extends TForm1FRM {
2
3     public TForm1() {
4         CriarEventos();
5
6     }
7     private void CriarEventos() {
8     } //fim do método CriarEventos
9
10
11     ///////////////////////////////////////////////////////////////////
12     //Área para inserção de código //
13     ///////////////////////////////////////////////////////////////////
14
15
16
17     //método principal (cria e abre formulário)
18     public static void main(String args[]) {
19         new TForm1().show();
20     }
21
22
23 }
24
25
```



# Resultados e discussão



## DELPHI2JAVA X DELPHI2JAVA-II

	Conversão de interfaces gráficas	Conversão de código e tratadores de eventos	Utilização de classe proprietária	Utilização AWT	Utilização <i>Swing</i>
Delphi2Java	Sim	Sim	Sim	Sim	Não
Delphi2Java-II	Sim	Não	Não	Sim	Sim



# Conclusão

Além de agregar conhecimento em relação a programação Java, o desenvolvimento deste projeto serviu para conhecer melhor uma área do mercado pouco explorada no meio acadêmico, a migração de softwares, que surge como uma solução paliativa no momento de decidir se deve reescrever toda a aplicação em uma nova plataforma ou manter o que se tem e torná-la obsoleta.





# Extensões

- converter o código fonte dos tratadores de eventos e demais métodos existentes na aplicação;
- implementar a conversão de componentes do Delphi com acesso a banco de dados utilizando a ferramenta Java DataBase Connectors (JDBC).

