

# **Budowa aplikacji z graficznym interfejsem użytkownika - GUI (Graphic User Interface)**

- 1. Budowa okienek dialogowych do wyświetlania danych – wyświetlanie produktów**
- 2. Pobranie danych z wybranego wiersza tabeli wyświetlającej produkty**
- 3. Okienko dialogowe do wprowadzania rachunków**
- 4. Okienko dialogowe do wyświetlania rachunków**
- 5. Okienko dialogowe do wprowadzania zakupów**

# **Budowa aplikacji z graficznym interfejsem użytkownika - GUI (Graphic User Interface)**

## **1. Budowa okienek dialogowych do wyświetlania danych**

**Dodanie nowego standardowego okna dialogowego do projektu, nadanie plikom nazwy **wprodukty.h** i **wprodukty.cpp** i wstawienie komponentu **TStringGrid****

The screenshot displays the C++Builder 6 IDE interface. At the top, the menu bar includes File, Edit, Search, View, and Project. The main workspace shows a form titled 'Form1' with a code editor containing the following content:

```
wprodukty.cpp  
RachunekApp.cpp | dprodukt.h | wprodukty.cpp  
//  
//
```

A dialog window titled 'Wyświetlenie produktów' is overlaid on the form. It contains a TStringGrid component with a 4x4 grid structure. The grid has a blue header row and a blue header column. The grid is currently empty. Below the grid are 'OK' and 'Cancel' buttons.

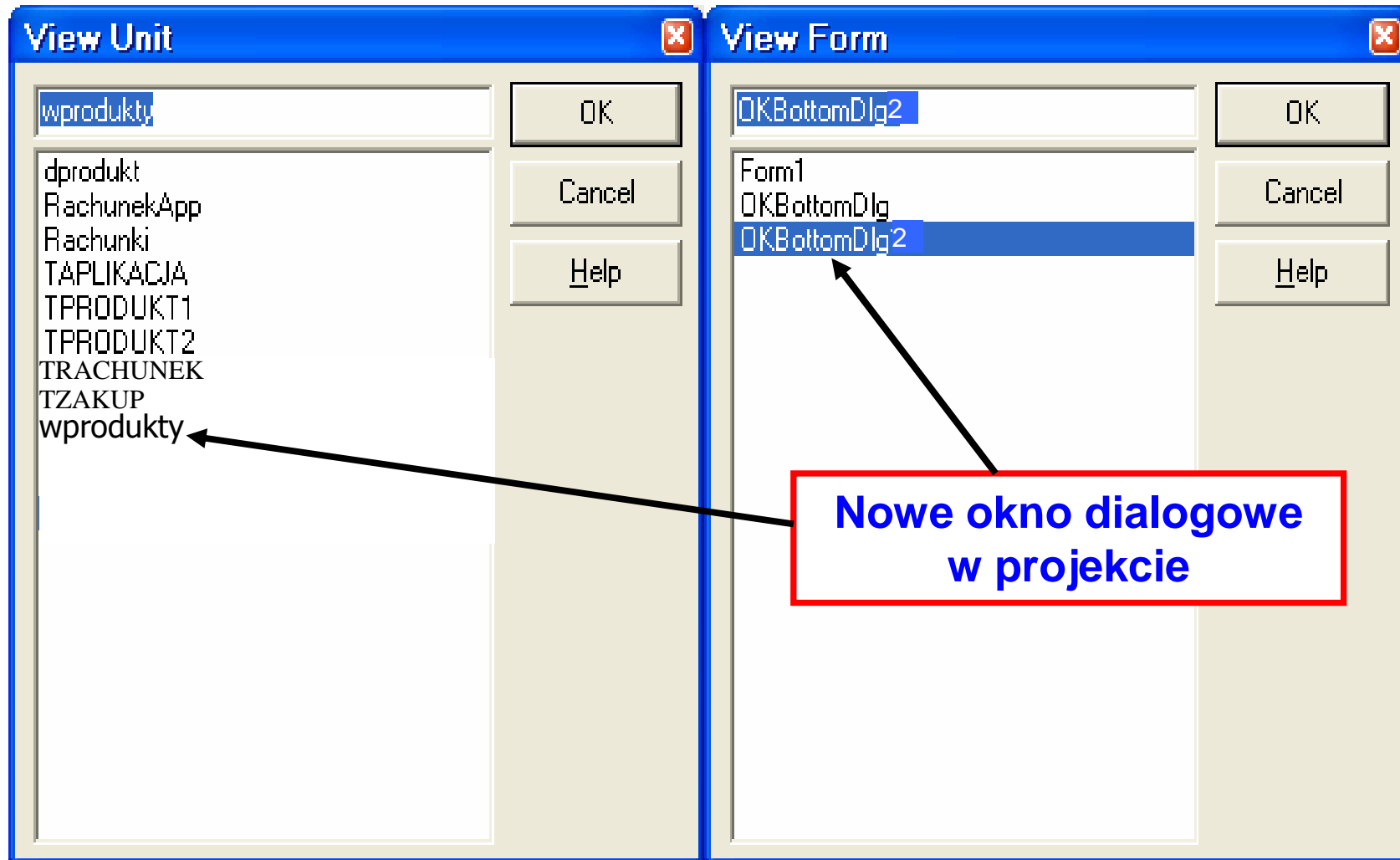
The 'Object Inspector' on the left shows the 'StringGrid1' component selected. Its properties are listed as follows:

- Auto: false
- Auto: false
- BiDi: bdLeftToRight
- Border: [biSystemMenu, biMini]
- Border: bsDialog
- Border: 0
- Caption: swietlenie produktow
- Client: 214
- Client: 389
- Color: clBtnFace
- Constraints: (TSizeConstraints)
- Control: true
- Cursor: crDefault
- Default: dmActiveForm
- Default: false
- Drag: dkDrag
- Drag: dmManual
- Enabled: true
- Font: (TFont)
- Font: fsNormal
- Height: 241
- Help: 0
- Help: 0
- Help: htContext
- Hint: 0
- Horizontal: (TControlScrollBar)
- Icon: (None)
- Key: false

The 'Components' palette on the right shows a list of standard components. 'TStringGrid [Grids]' is selected and highlighted in blue. The 'Add to form' button is visible at the bottom of the palette.

The Windows taskbar at the bottom shows the Start button, C++Builder 6, Windows Commander..., W\_10, and the system tray with the time 17:33.

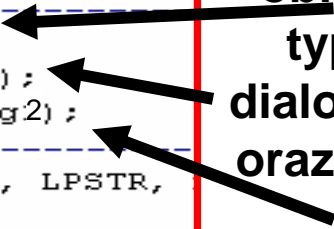
# Dodanie nowego okna dialogowego do projektu



```
Rachunki.cpp
RachunekApp.h  Rachunki.cpp  dprodukt.h  wprodukty.cpp

//-----
#include <vc1.h>
#pragma hdrstop
//-----
USEFORM("RachunekApp.cpp", Form1);
USEFORM("dprodukt.cpp", OKBottomDlg);
USEFORM("wprodukty.cpp", OKBottomDlg2);
//-----
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR,
(
    try
    (
        Application->Initialize();
        Application->CreateForm(__classid(TForm1), &Form1);
        Application->CreateForm(__classid(TOKBottomDlg), &OKBottomDlg);
        Application->CreateForm(__classid(TOKBottomDlg2), &OKBottomDlg2);
        Application->Run();
    )
    catch (Exception &exception)
    (
        Application->ShowException(&exception);
    )
    catch (...)
    (
        try
        (
            throw Exception("");
        )
        catch (Exception &exception)
        (
            Application->ShowException(&exception);
        )
    )
    return 0;
)
//-----
```

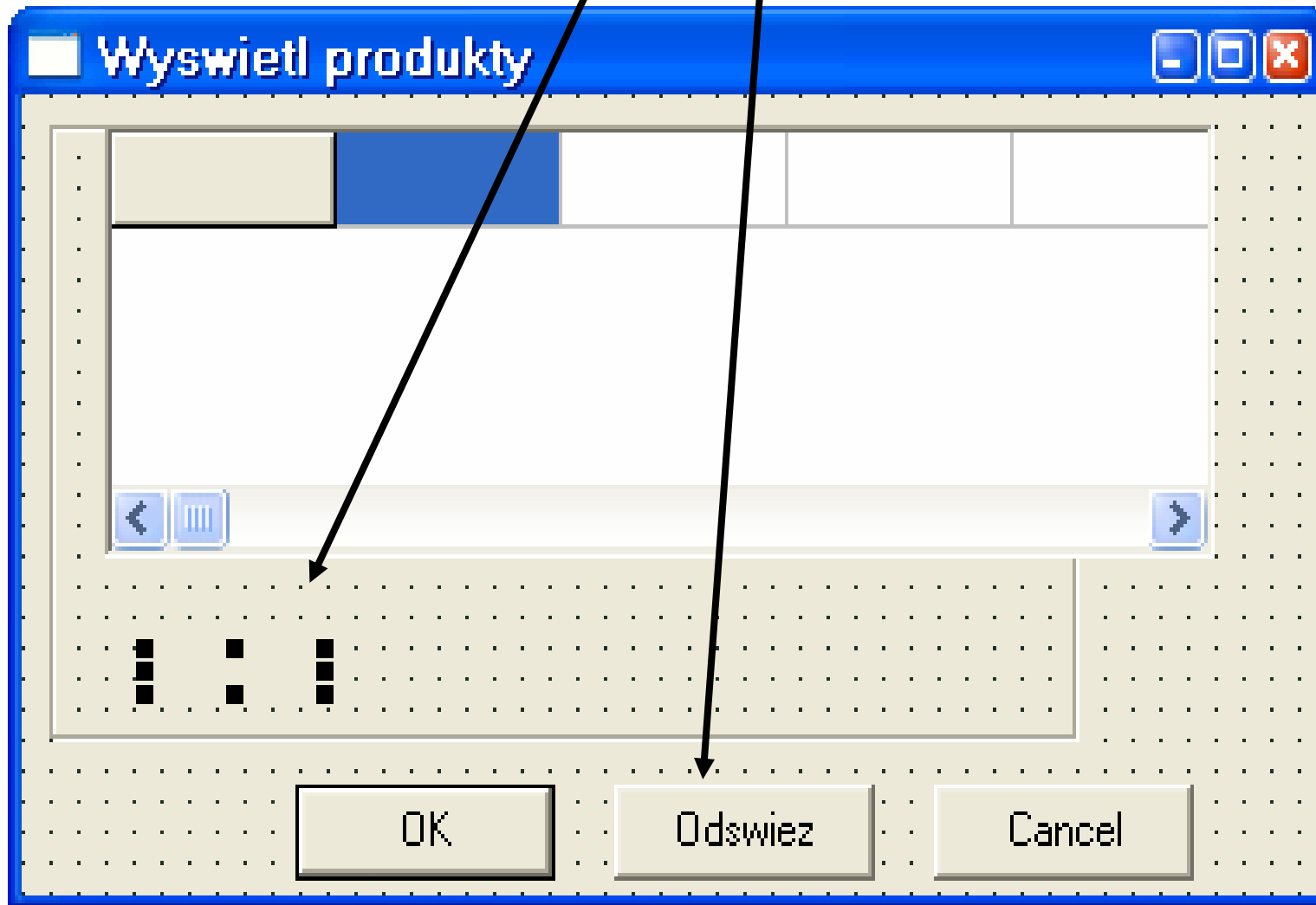
Główny plik GUI z  
automatycznie dołączonym  
obiektem formularza głównego  
typu **TForm1**, obiektem okna  
dialogowego typu **TOKBottomDlg**  
oraz obiektem okna dialogowego  
typu **TOKBottomDlg2**



```
//-----  
#ifndef wproduktyH  
#define wproduktyH  
//-----  
#include <vc1\System.hpp>  
#include <vc1\Windows.hpp>  
#include <vc1\SysUtils.hpp>  
#include <vc1\Classes.hpp>  
#include <vc1\Graphics.hpp>  
#include <vc1\StdCtrls.hpp>  
#include <vc1\Forms.hpp>  
#include <vc1\Controls.hpp>  
#include <vc1\Buttons.hpp>  
#include <vc1\ExtCtrls.hpp>  
#include <Grids.hpp>  
//-----  
class TOKBottomDlg2 : public TForm  
{  
    __published:  
        TButton *OKBtn;  
        TButton *CancelBtn;  
        TBevel *Bevel1;  
        TStringGrid *StringGrid1;  
private:  
public:  
    virtual __fastcall TOKBottomDlg2(TComponent* AOwner);  
};  
//-----  
extern PACKAGE TOKBottomDlg2 *OKBottomDlg2;  
//-----  
#endif
```

Plik nagłówek standardowego okna dialogowego po wstawieniu komponentu TStringGrid

Wstawienie przycisku typu **TButton** do odświeżania zawartości okna oraz pola komunikatów typu **TLabel**





```
wprodukty.cpp
dzakup.cpp | wprodukty.h | wrachunki.cpp | TRACHUNEK.CPP | TZAKUP.cpp | TZAKUP.h | TPRODUKT1.CPP
#include <vcl\Forms.hpp>
#include <vcl\Controls.hpp>
#include <vcl\Buttons.hpp>
#include <vcl\ExtCtrls.hpp>
#include <Grids.hpp>
#include "RachunekApp.h"
//-----
class TOKBottomDlg2 : public TForm
{
    __published:
        TButton *OKBtn;
        TButton *CancelBtn;
        TBevel *Bevel1;
        TStringGrid *StringGrid1;
        TButton *Button1;
        TLabel *Label1;
        void __fastcall StringGrid1Click(TObject *Sender);
        void __fastcall Button1Click(TObject *Sender);
private: //ręcznie = obliczanie ceny netto na podstawie ceny brutto
string __fastcall cena_netto2(string a, string b); //oraz podatku
public:
    bool wybrany; //recznie - zmienna na wybrany rowek oprócz pierwszego
    string tab[5]; //recznie - tablica na atrybuty lancuchowe produktu
    bool __fastcall Execute(); //recznie - obsługa okienka dialogowego
    virtual __fastcall TOKBottomDlg2(TComponent* AOwner);
};
//-----
extern PACKAGE TOKBottomDlg2 *OKBottomDlg2; //deklaracja globalnego wskaźnika
//-----
#endif
```

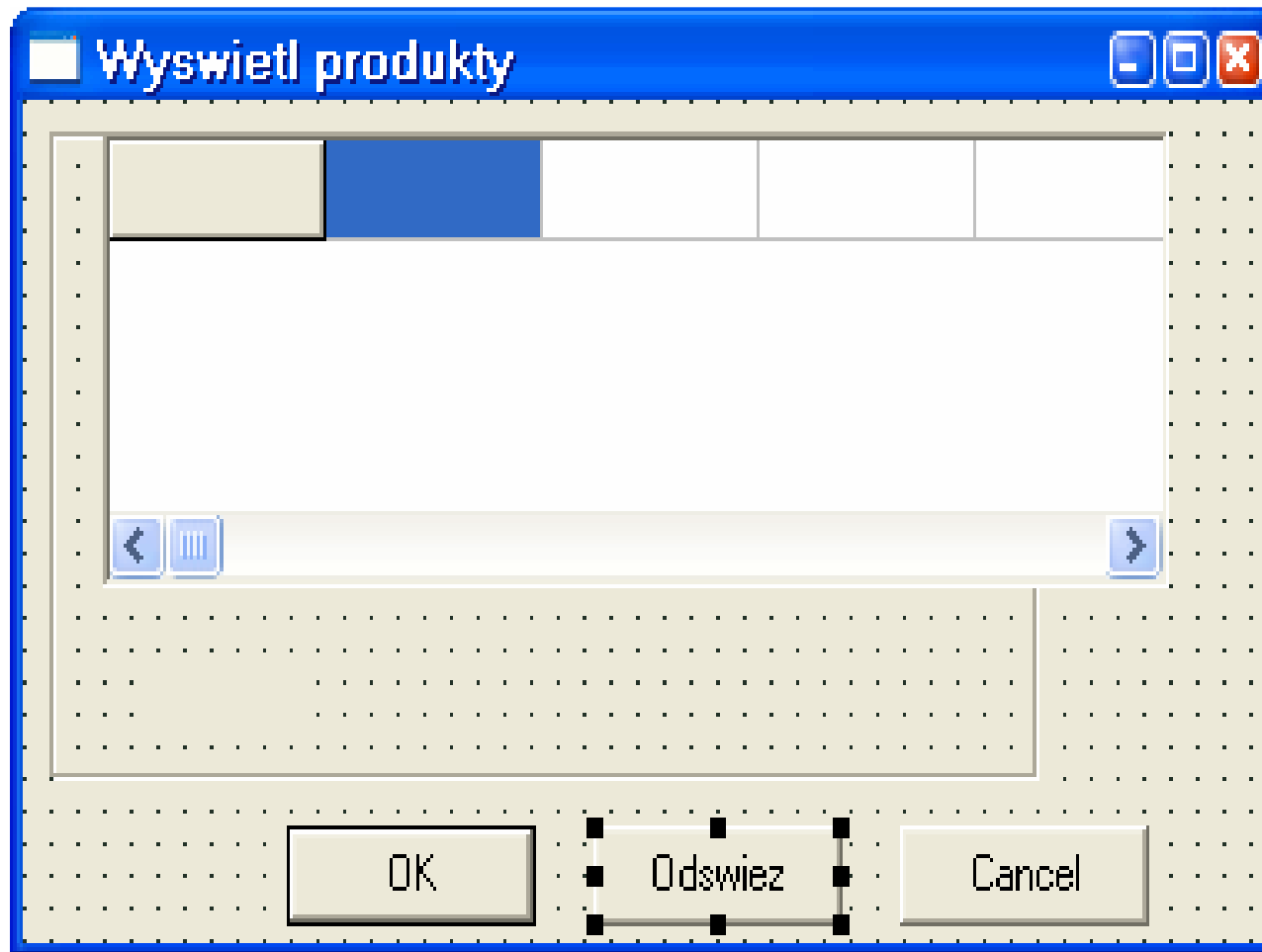
Plik nagłówkowy standardowego okna dialogowego po wstawieniu komponentu **TStringGrid**, przycisku typu **TButton**, pola komunikatów **TLabel** oraz metody **Execute**, atrybutów **wybrany** oraz **tab**

```
bool __fastcall Execute(); //recznie - obsługa okienka dialogowego
virtual __fastcall TOKBottomDlg2(TComponent* AOwner);
```



```
#include <vcl.h>
#pragma hdrstop
#include "wprodukty.h"
//-----
#pragma resource "*.dfm"
TOKBottomDlg2 *OKBottomDlg2; //wskaznik globalny okna dialogowego
//-----
__fastcall TOKBottomDlg2::TOKBottomDlg2(TComponent* AOwner) //automatycznie
: TForm(AOwner)
{
    StringGrid1->Cells[0][0]="L.p."; //utworzenie jednorazowo
    StringGrid1->Cells[1][0]="Nazwa"; //nagłówka tabeli w konstruktorze
    StringGrid1->Cells[2][0]="Cena"; //okna dialogowego
    StringGrid1->Cells[3][0]="Podatek";
    StringGrid1->Cells[4][0]="Promocja";
    StringGrid1->Visible=true; }
//-----
bool __fastcall TOKBottomDlg2::Execute() //ręcznie
{
    //wyswietlanie tabeli
    ShowModal(); //przejscie do trybu modalnego, który konczy sie
    return true; //nacisnieciem klawisza OK lub Cancel;
}
```

Plik modułowy standardowego okna dialogowego – definicja metody **Execute()** wyświetlającej zawartość komponentu typu **TStringGrid** oraz konstruktor wstawiający nagłówki do tabeli typu **TStringGrid**

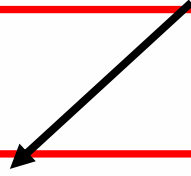


Po kliknięciu na przycisk typu `TButton` z napisem „Odswiez” przechodzi się do pliku źródłowego okienka dialogowego `wprodukty.cpp`, gdzie wygenerowała się pusta metoda do obsługi tego klawisza

```
void __fastcall TOKBottomDlg2::Button1Click(TObject *Sender)
```

```
wprodukty.cpp
RachunekApp.cpp wprodukty.h dprodukt.cpp
#include <vcl\Forms.hpp>
#include <vcl\Controls.hpp>
#include <vcl\Buttons.hpp>
#include <vcl\ExtCtrls.hpp>
#include <Grids.hpp>
#include "RachunekApp.h"
//-----
class TOKBottomDlg2 : public TForm
{
  __published:
    TButton *OKBtn;
    TButton *CancelBtn;
    TBevel *Bevel1;
    TStringGrid *StringGrid1;
    TButton *Button1;
    TLabel *Label1;
    void __fastcall StringGrid1Click(TObject *Sender);
    void __fastcall Button1Click(TObject *Sender);
private: // ręcznie = obliczanie ceny netto na podstawie ceny brutto
string __fastcall cena_netto4(string a, string b, string c); // oraz podatku i promocji
string __fastcall cena_netto3(string a, string b); // oraz promocji
string __fastcall cena_netto2(string a, string b); // oraz podatku
public:
  bool wybrany; // ręcznie - zmienna na wybrany rowek oprócz pierwszego
string tab[5]; // ręcznie - tablica na atrybuty lancuchowe produktu
  bool __fastcall Execute(); // ręcznie - obsługa okienka dialogowego
  virtual __fastcall TOKBottomDlg2(TComponent* AOwner);
};
//-----
extern PACKAGE TOKBottomDlg2 *OKBottomDlg2; // deklaracja globalnego wskaźnika
//-----
#endif
```

Metoda wywoływana podczas działania metody **Execute()**



void \_\_fastcall Button1Click(TObject \*Sender);

bool \_\_fastcall Execute(); // ręcznie - obsługa okienka dialogowego

**Definicja metody wstawiającej dane z kolekcji produkty do komponentu typu TStringGrid**

```
void __fastcall TOKBottomDlg2::Button1Click(TObject *Sender) //automatycznie
{
    //recznie
    TKol2<TProdukt1> produkty=Form1->aplikacja.Podaj_produkty();
    int i=0; TProdukt1* produkt;
    char lan[10];
    produkty.Zeruj(); //odczytywanie kolekcji od początku
    while (produkty.Koniec()==0) //odswiezanie zawartosci komorek tabeli
    { produkt=produkty.Podaj_nast();
      i++; //zmienna do liczenia rowkow tabeli
      TStringGrid1->RowCount=i+1; //dodanie nowego rowka do tabeli
      TStringGrid1->Cells[0][i]=i; //numer pozycji w rachunku
      TStringGrid1->Cells[1][i]=produkt->Podaj_nazwe().c_str();
      TStringGrid1->Cells[2][i]=gcvt(produkt->Podaj_cene(),3,lan);
      float pom=produkt->Podaj_podatek();
      if (pom==-1) //jesli brak podatku wyswietla sie pusty lancuch
          TStringGrid1->Cells[3][i]="";
      else
          TStringGrid1->Cells[3][i]=gcvt(produkt->Podaj_podatek(),3,lan);
      TStringGrid1->Cells[4][i]=""; //pusty lancuch, poniewaz brak promocji
    }
    TStringGrid1->Visible=true;
}
//-----
```

```
#include <Dialogs.hpp>
#include <ExtCtrls.hpp>
#include <Buttons.hpp>
#include "TAplikacja.h"
#include "dprodukt.h"
#include "wprodukty.h"
//-----
class TForm1 : public TForm
{
    __published:      // IDE-managed Components
        TMainMenu *MainMenu1;
        TMenuItem *Plik1;
        TMenuItem *Dane1;
        TMenuItem *Wyswietl1;
        TMenuItem *Oprogramie1;
        TMenuItem *Dodajprodukt1;
        TMenuItem *Dodajrachunek1;
        TMenuItem *Dodajzakup1;
        TMenuItem *Wyswietlprodukty2;
        TMenuItem *Wyswietlrachunek2;
        TMenuItem *N1;
        TMenuItem *Koniec1;
        TMenuItem *oplik1;
        TMenuItem *zplik1;
        void __fastcall Dodajprodukt1Click(TObject *Sender);
        void __fastcall Wyswietlprodukty2Click(TObject *Sender);
private:      // User declarations
public:      // User declarations
        TAplikacja aplikacja; //połączenie z obiektem uzytkownika
        __fastcall TForm1(TComponent* Owner);
};
//-----
extern PACKAGE TForm1 *Form1; //deklaracja globalnego wskaznika
```

Plik nagłówkowy formularza głównego po wstawieniu okna dialogowego do wyświetlania produktów – wstawienie automatycznie metody **Wyswietlprodukty2Click** wywołującej to okno





**RachunekApp.cpp**

```
void __fastcall TForm1::Wyswietlprodukty2Click(TObject *Sender) //automatycznie
{
    OKBottomDlg2->Execute(); //recznie
}
```

40: 38 Modified Insert

**Sporządzanie rachunkow**

Pliki Dane Wyswietl O programie

- Wyswietl produkty
- Wyswietl rachunki

Po kliknięciu na „Wyswietl produkty” można przejść do pisania kodu metody obsługującej wyświetlanie produktów w pliku **RachunekApp.cpp** głównego formularza aplikacji

Slajd 74 z 83 Projekt domyślny Polski

start C++Builder 6 Windows Commander ... W\_10 PL 19:32

# Uruchomienie aplikacji

**Dodawanie produktów**

Nazwa:

Cena:

Podatek:

Promocja:

**Sporządzanie rachunków**

Wyszukiwanie do Pomocy

68%

Pliki Dane **Wyswietl** O programie

- Wyswietl produkty
- Wyswietl rachunki

**Wyswietl produkty**

L.p.	Nazwa	Cena	Podatek	Promocja
------	-------	------	---------	----------

**Wyswietl produkty**

L.p.	Nazwa	Cena	Podatek	Promocja
1	1	1.01	1	



# **Budowa aplikacji z graficznym interfejsem użytkownika - GUI (Graphic User Interface)**

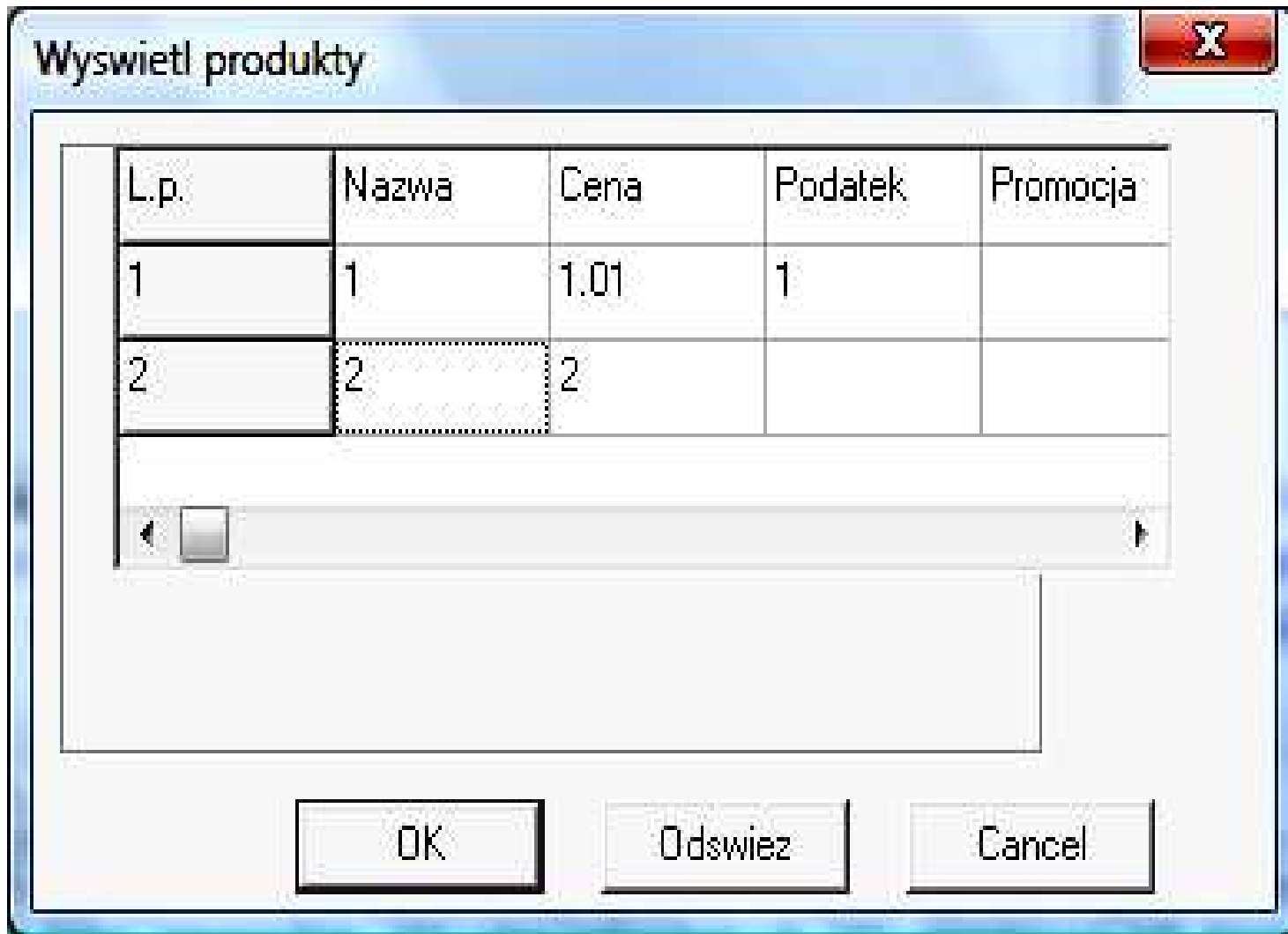
- 1. Budowa okienek dialogowych do wyświetlania danych – wyświetlanie produktów**
- 2. Pobranie danych z wybranego wiersza tabeli wyświetlającej produkty**

```
wprodukty.cpp
drachunek.cpp | dzakup.cpp | wprodukty.h | wrachunki.cpp | TF
#include <vcl\Forms.hpp>
#include <vcl\Controls.hpp>
#include <vcl\Buttons.hpp>
#include <vcl\ExtCtrls.hpp>
#include <Grids.hpp>
#include "RachunekApp.h"
//-----
class TOKBottomDlg2 : public TForm
{
  __published:
    TButton *OKBtn;
    TButton *CancelBtn;
    TBevel *Bevel1;
    TStringGrid *StringGrid1;
    TButton *Button1;
    TLabel *Label1;
    void __fastcall StringGrid1Click(TObject *Sender);
    void __fastcall Button1Click(TObject *Sender);
private: // ręcznie = obliczanie ceny netto na podstawie ceny brutto
string __fastcall cena_netto2(string a, string b); // oraz podatku
public:
  bool wybrany; // ręcznie - zmienna na wybrany rowek oprócz pierwszego
  string tab[51]; // ręcznie - tablica na atrybuty lancuchowe produktu
  bool __fastcall Execute(); // ręcznie - obsługa okienka dialogowego
  virtual __fastcall TOKBottomDlg2(TComponent* AOwner);
};
//-----
extern PACKAGE TOKBottomDlg2 *OKBottomDlg2; // deklaracja globalnego wskaźnika
//-----
#endif
```

Plik nagłówkowy standardowego okna dialogowego do wyświetlania produktów – deklaracja metody **StringGrid1Click** do obsługi wyboru wiersza z danymi w komponencie typu **TStringGrid** za pomocą klikania myszą na wybranym wierszu. Metoda ta jest wywoływana podczas działania metody **Execute()**

void \_\_fastcall StringGrid1Click(TObject \*Sender);

bool \_\_fastcall Execute(); // ręcznie - obsługa okienka dialogowego



```
string __fastcall TOKBottomDlg2::cena_netto2(string a, string b) //ręcznie
{ float x,y; char lan[10];
  x=atof(a.c_str()); y=atof(b.c_str());
  x=(x*100)/(100+y); //cena netto na podstawie ceny brutto x i podatku y
  return gcvt(x,3,lan);
}

void __fastcall TOKBottomDlg2::StringGrid1Click(TObject *Sender) //automatycznie
{ char pom[10]; //ręcznie
  TStringList* lista;
  wybrany=true;
  if (StringGrid1->Row==0) //kliknieto na rowek bez danych
  { wybrany=false; //czyli nie wybrano danych
    return; //koniec wprowadzania danych o produkcie
  }
  lista= StringGrid1->Rows[StringGrid1->Row];
  for (int i=1; i<=4;i++)
    tab[i]=lista->Strings[i].c_str(); //pobranie z rowka atrybutow produktu
  //korekta zawartosci tab[2] reprezentujacego cene brutto na cene netto
  //oraz wyznaczenie typu produktu w tab[0] w zaleznosci, czy wprowadzono podatek
  if(tab[3]!="")
  { tab[0]="2"; //jest podatek = typ TProdukt2
    tab[2]=cena_netto2(tab[2],tab[3]); } //obliczanie ceny netto
  else
    tab[0]="1"; //podatku - jest TProdukt1
}
```

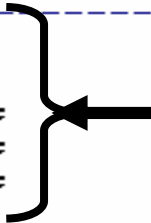
# **Budowa aplikacji z graficznym interfejsem użytkownika - GUI (Graphic User Interface)**

- 1. Budowa okienek dialogowych do wyświetlania danych – wyświetlanie produktów**
- 2. Pobranie danych z wybranego wiersza tabeli wyświetlającej produkty**
- 3. Okienko dialogowe do wprowadzania rachunków**



```
Rachunki.cpp
RachunekApp.h Rachunki.cpp
#include <vc1.h>
#pragma hdrstop
//-----
USEFORM("RachunekApp.cpp", Form1);
USEFORM("dprodukt.cpp", OKBottomDlg);
USEFORM("drachunek.cpp", OKBottomDlg1);
USEFORM("wprodukty.cpp", OKBottomDlg2);
USEFORM("wrachunki.cpp", OKBottomDlg3);
USEFORM("dzakup.cpp", OKBottomDlg4);
//-----
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
{
    try
    {
        Application->Initialize();
        Application->CreateForm(__classid(TForm1), &Form1);
        Application->CreateForm(__classid(TOKBottomDlg), &OKBottomDlg);
        Application->CreateForm(__classid(TOKBottomDlg1), &OKBottomDlg1);
        Application->CreateForm(__classid(TOKBottomDlg2), &OKBottomDlg2);
        Application->CreateForm(__classid(TOKBottomDlg3), &OKBottomDlg3);
        Application->CreateForm(__classid(TOKBottomDlg4), &OKBottomDlg4);
        Application->Run();
    }
    catch (Exception &exception)
    {
        Application->ShowException(&exception);
    }
    catch (...)
    {
        try
        {
            throw Exception("");
        }
        catch (Exception &exception)
        {
            Application->ShowException(&exception);
        }
    }
    return 0;
}
```

Główny plik GUI z  
automatycznie dołączonym  
obiektem formularza głównego  
typu **TForm1** oraz obiektami  
okienek dialogowych

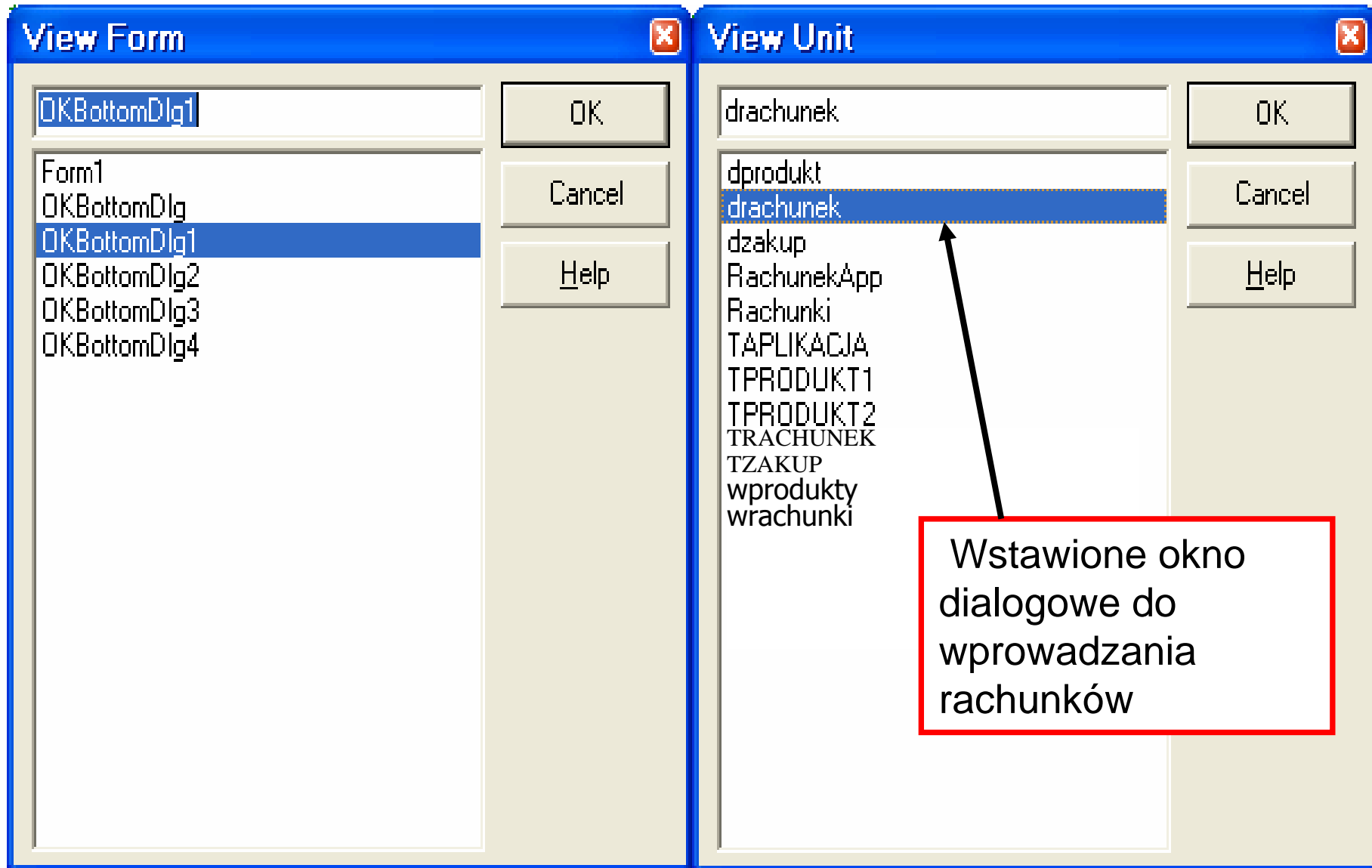


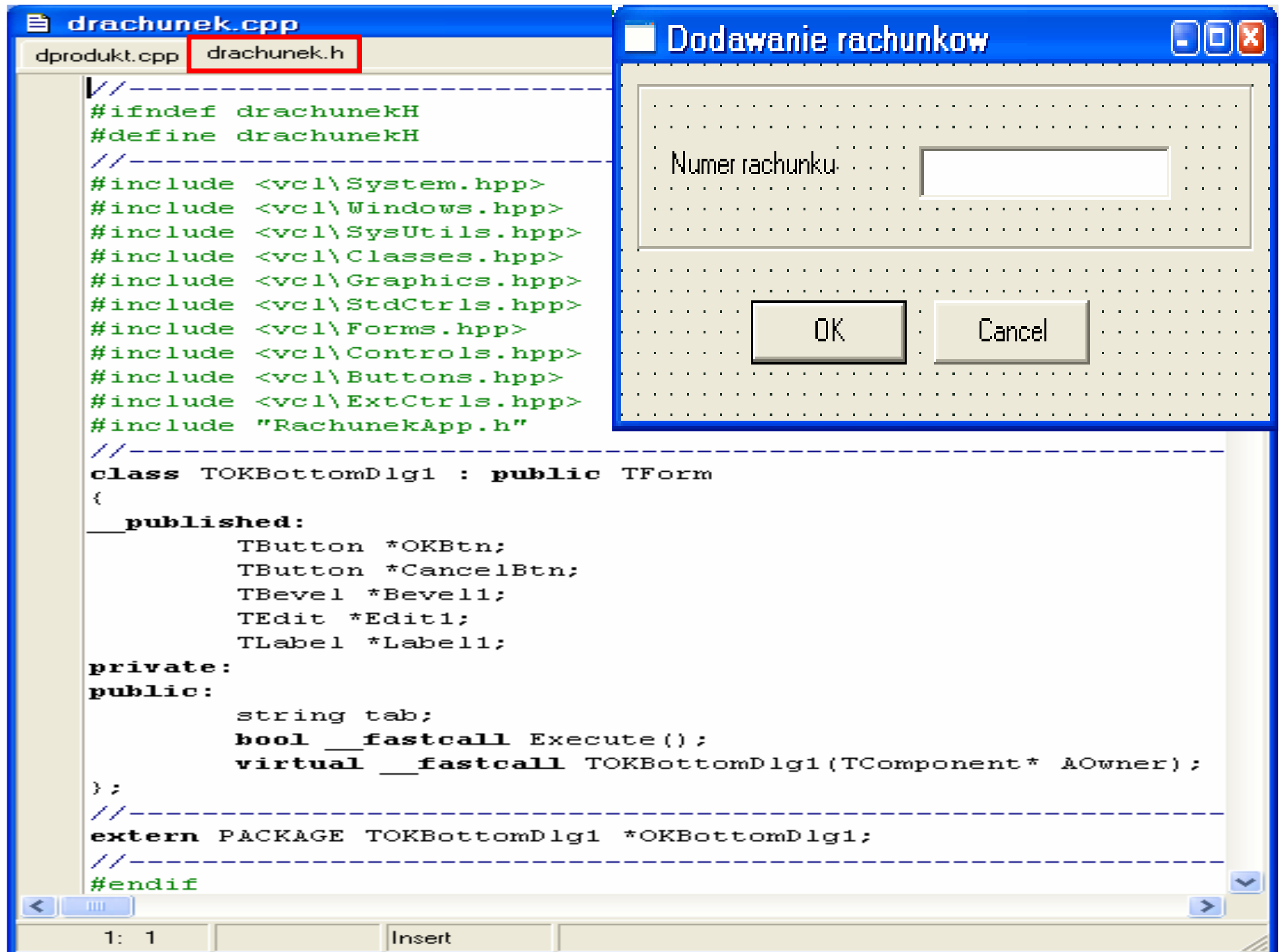
```
//-----  
#ifndef RachunekAppH  
#define RachunekAppH  
//-----  
#include <Classes.hpp>  
#include <Controls.hpp>  
#include <StdCtrls.hpp>  
#include <Forms.hpp>  
#include <Menus.hpp>  
#include <Dialogs.hpp>  
#include <ExtCtrls.hpp>  
#include <Buttons.hpp>  
#include "T Aplikacja.h"  
#include "dprodukt.h"  
#include "drachunek.h"  
#include "dzakup.h";  
#include "wprodukty.h"  
#include "wrachunki.h"
```



```
class TForm1 : public TForm
{
    published:           // IDE-managed Components
        TMainMenu *MainMenu1;
        TMenuItem *Plik1;
        TMenuItem *Dane1;
        TMenuItem *Wyświetl1;
        TMenuItem *Oprogramie1;
        TMenuItem *Dodajprodukt1;
        TMenuItem *Dodajrachunek1;
        TMenuItem *Dodajzakup1;
        TMenuItem *Wyświetlprodukty2;
        TMenuItem *Wyświetlrachunek2;
        TMenuItem *N1;
        TMenuItem *Koniec1;
        TMenuItem *oplik1;
        TMenuItem *zplik1;
        void __fastcall Dodajprodukt1Click(TObject *Sender);
        void __fastcall Wyświetlprodukty2Click(TObject *Sender);
        void __fastcall Dodajrachunek1Click(TObject *Sender);
        void __fastcall Wyświetlrachunek2Click(TObject *Sender);
        void __fastcall Dodajzakup1Click(TObject *Sender);
        void __fastcall Oprogramie1Click(TObject *Sender);
private:           // User declarations
public:           // User declarations
        TApplikacja aplikacja; //połączenie z obiektem uzytkownika
        __fastcall TForm1(TComponent* Owner);
};
//-----
extern PACKAGE TForm1 *Form1; //deklaracja globalnego wskaznika
//-----
#endif
```

Plik nagłówkowy formularza z metodami obsługującymi klikania w pozycje list rozwijanych menu formularza





```
drachunek.cpp
dprodukt.cpp drachunek.cpp

#include <vcl.h>
#pragma hdrstop

#include "drachunek.h"
//-----
#pragma resource "*.dfm"
TOKBottomDlg1 *OKBottomDlg1; //wskaznik globalny okna dialogowego
//-----
__fastcall TOKBottomDlg1::TOKBottomDlg1(TComponent* AOwner)
    : TForm(AOwner)
{
}
//-----
bool __fastcall TOKBottomDlg1::Execute()
{
    //przejscie do trybu modalnego, ktory konczy sie
    if (ShowModal() ==mrOk) //nacisnieciem klawisza OK lub Cancel;
    {
        tab=Edit1->Text.c_str();
        return true;}
    else
        return false;
}
```

Obsługa wprowadzania numeru rachunku w oknie dialogowym za pomocą metody `Execute()`

20: 65 Modified Insert

RachunekApp.cpp

dprodukt.cpp

drachunek.cpp

RachunekApp.cpp

Metoda **Dodajrachunek1Click** w formularzu głównym pobierająca numery nowego rachunku z okna dialogowego za pomocą metody **Execute()**

```
//automatycznie
void __fastcall TForm1::Dodajrachunek1Click(TObject *Sender)
{ int wynik;          //recznie
  if (OKBottomDlg1->Execute ())
  { if (OKBottomDlg1->tab != "")
    { wynik = aplikacja.Wstaw_rachunek (atoi ((OKBottomDlg1->tab).c_str ()));
      Application->MessageBox (napisy[wynik], "");
    }
  }
  else
    Application->MessageBox ("Nie podano numeru rachunku", "");
}
else
  Application->MessageBox ("Anulowano dodanie rachunku", "");
}
```

46: 73

Modified

Insert

Uruchomiona aplikacja – wprowadzanie nowego rachunku

**Sporządzanie rachunków**  
Pliki Dane Wyświetl O programie  
Dodaj produkt  
Dodaj rachunek  
Dodaj zakup

**Dodawanie rachunków**  
Numer rachunku 1  
OK Cancel

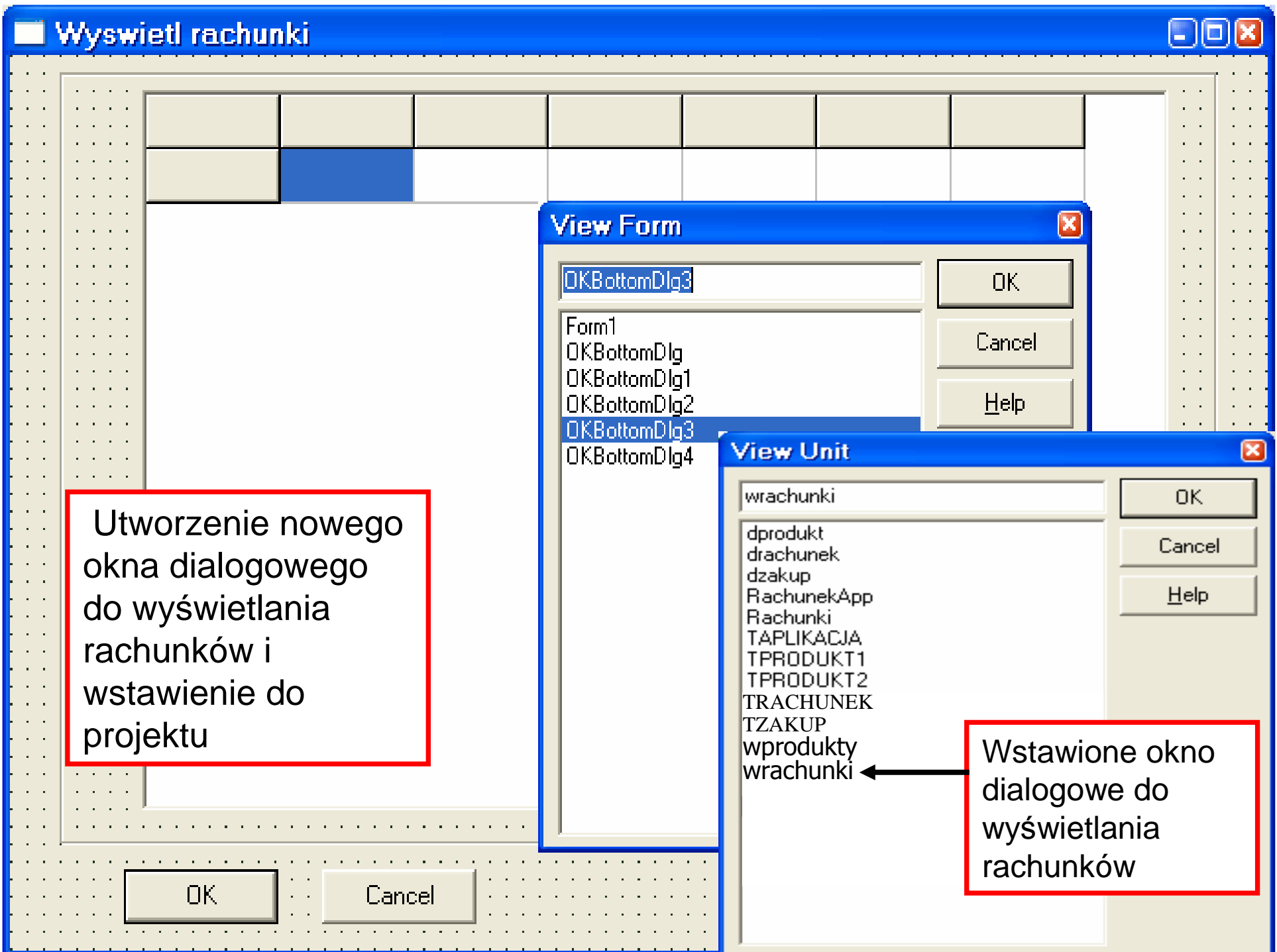
Wstawiono dane  
OK

Microsoft PowerPoint - [W\_10]  
Plik Edycja Widok Wstaw Format Narzędzia Pokaż slajdów Okno Pomoc  
Arial 18 B I U S  
68%  
96  
97  
98  
99  
Zastosuj układ slajdu:  
Układy tekstu  
wartości  
Pokaż przy wstawianiu slajdów  
Polski  
Rachunki  
PL 20:28

# **Budowa aplikacji z graficznym interfejsem użytkownika - GUI (Graphic User Interface)**

- 1. Budowa okienek dialogowych do wyświetlania danych – wyświetlanie produktów**
- 2. Pobranie danych z wybranego wiersza tabeli wyświetlającej produkty**
- 3. Okienko dialogowe do wprowadzania rachunków**
- 4. Okienko dialogowe do wyświetlania rachunków**





```
wrachunki.cpp | drachunek.cpp | RachunekApp.cpp | wrachunki.h
#ifndef wrachunkiH
#define wrachunkiH
//-----
#include <vc1\System.hpp>
#include <vc1\Windows.hpp>
#include <vc1\SysUtils.hpp>
#include <vc1\Classes.hpp>
#include <vc1\Graphics.hpp>
#include <vc1\StdCtrls.hpp>
#include <vc1\Forms.hpp>
#include <vc1\Controls.hpp>
#include <vc1\Buttons.hpp>
#include <vc1\ExtCtrls.hpp>
#include <Grids.hpp>
#include "RachunekApp.h"
//-----
class TOKBottomDlg3 : public TForm
{
    __published:
        TButton *OKBtn;
        TButton *CancelBtn;
        TBevel *Bevel1;
        TStringGrid *StringGrid1;
private: //recznie -wypełnianie pustymi lancuchami nie wykorzystanych komorek
    void __fastcall TOKBottomDlg3::zeruj(int i, int j);
public: bool __fastcall Execute(); //recznie - obsługa okienka dialogowego
    virtual __fastcall TOKBottomDlg3(TComponent* AOwner);
};
```

Plik nagłówkowy  
okna dialogowego  
do wyświetlania  
rachunków

```
bool _fastcall TOKBottomDlg3::Execute() //ręcznie
{TKol2<TRachunek> rachunki=Form1->aplikacja.Podaj_rachunki();
 int i=0,j;
 TRachunek* rachunek;
 TZakup* zakup; TProdukt1* produkt;
 char tab[10];
 rachunki.Zeruj(); //przejscie w kolekcji rachunkow na poczatek
 while (rachunki.Koniec()==0)
 { j=0; //zmienna do numerowania pozycji rachunku
 i++; //zmienna do liczenia rowkow tabeli
 StringGrid1->RowCount=i+1; //dodanie nowego rowka
 rachunek=rachunki.Podaj_nast();
 StringGrid1->Cells[0][i]="Nr rach: ";
 StringGrid1->Cells[1][i]=itoa(rachunek->Podaj_numer(),tab,10);
 zeruj(i,2); //wypełnianie pustymi lancuchami komorek nie wykorzystanych
 i++;
 StringGrid1->RowCount=i+1; //dodanie nowego rowka
 StringGrid1->Cells[0][i]="L.p.";
 StringGrid1->Cells[1][i]="Nazwa";
 StringGrid1->Cells[2][i]="Cena";
 StringGrid1->Cells[3][i]="Podatek";
 StringGrid1->Cells[4][i]="Promocja";
 StringGrid1->Cells[5][i]="Ilosc";
 StringGrid1->Cells[6][i]="Wartosc";
 TKol2<TZakup> zakupy=rachunek->Podaj_zakupy();
 zakupy.Zeruj(); //przejscie w kolekcji zakupow wybranego rachunku na poczatek
```

Definicja metody `Execute()` do wypełniania komponentu typu `TStringGrid` zawartością kolekcji `rachunki`

```
while (zakupy.Koniec()==0)
{
    zakup=zakupy.Podaj_nast();
    produkt=zakup->Podaj_produkt();
    i++;           //numer rowka
    StringGrid1->RowCount=i+1; //dodanie nowego rowka
    j++;           //numer pozycji rachunku
    StringGrid1->Cells[0][i]=j;
    StringGrid1->Cells[1][i]=produkt->Podaj_nazwe().c_str();
    StringGrid1->Cells[2][i]=gcvt(produkt->Podaj_cene(),3,tab);
    float pom=produkt->Podaj_podatek();
    if (pom==-1) //jesli brak podatku wyswietla sie pusty lancuch
        StringGrid1->Cells[3][i]="";
    else
        StringGrid1->Cells[3][i]=gcvt(produkt->Podaj_podatek(),3,tab);

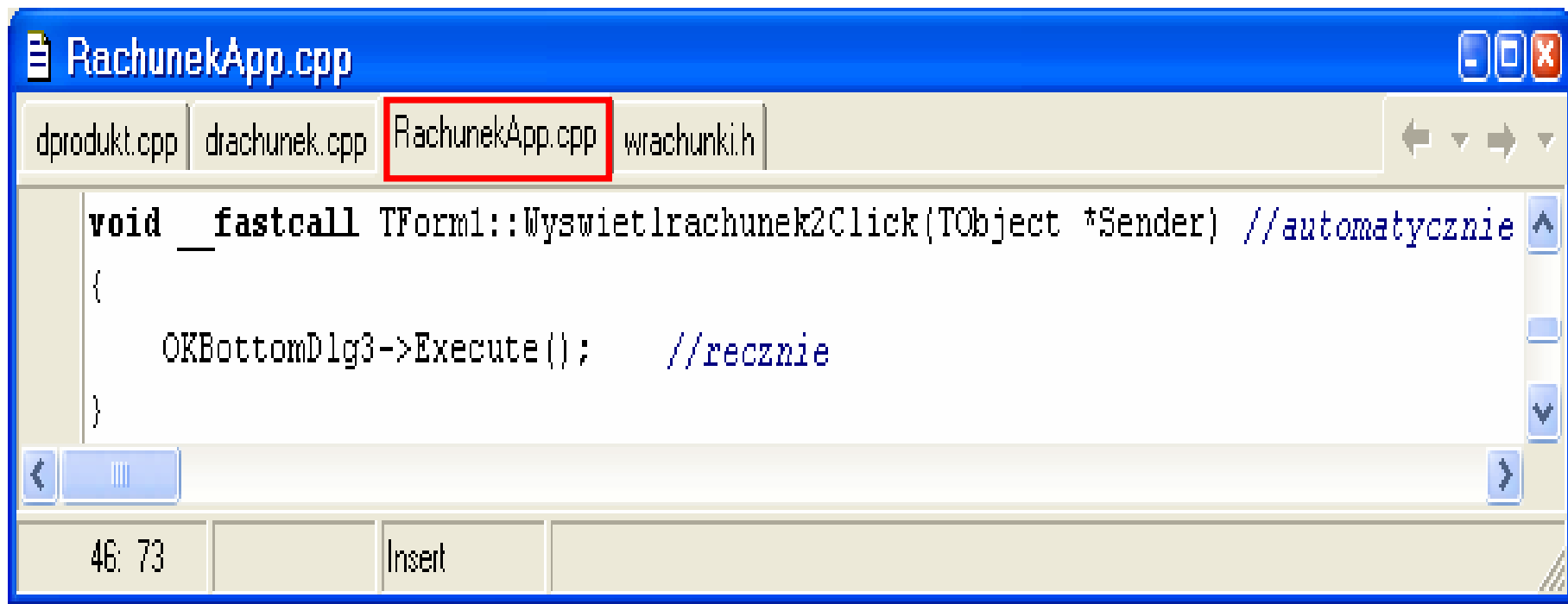
    StringGrid1->Cells[4][i]=""; //brak promocji = wyswietla sie pusty lancuch
    StringGrid1->Cells[5][i]=itoa(zakup->Podaj_ilosc(),tab,10);
    StringGrid1->Cells[6][i]=gcvt(zakup->Podaj_wartosc(),3,tab);
}
```

```
        i++;
        StringGrid1->RowCount=i+1; //dodanie nowego rowka
        StringGrid1->Cells[0][i]="Wartosc: ";
        StringGrid1->Cells[1][i]= gcvt(rachunek->Podaj_wartosc(),4,tab);
        zeruj(i,2); //wypełnianie pustymi lancuchami komorek nie wykorzystanych
        i++;
        StringGrid1->RowCount=i+1; //dodanie nowego rowka
        zeruj(i,0); //i wypelnienie pustymi lancuchami
    }
    if (rachunki.Pusta()==0)
        StringGrid1->Visible=true; //jesli sa dane w tabeli, bedzie wyswietlana
    else
        StringGrid1->Visible=false; // w przeciwnym wypadku jest ukryta
    ShowModal(); //przejscie do trybu modalnego, który konczy sie nacisnieciem
    return true; //klawisza OK lub Cancel;
}
```

```
#include <vcl.h>
#pragma hdrstop
#include "wrachunki.h"
//-----
#pragma resource "*.dfm"
TOKBottomDlg3 *OKBottomDlg3; //wskaznik globalny okna dialogowego
//-----
__fastcall TOKBottomDlg3::TOKBottomDlg3 (TComponent* AOwner)
    : TForm(AOwner)
{
}
//-----
void __fastcall TOKBottomDlg3::zeruj(int i, int j) //ręcznie
{
    // wypelnianie pustymi lancuchami komorek nie wykorzystanych
    for (; j<=6;j++)
        StringGrid1->Cells[j][i] = "";
}

```

## Obsługa zdarzenia klikania na pozycje menu *Wyświetl rachunki*, w pliku *RachunekApp.cpp*

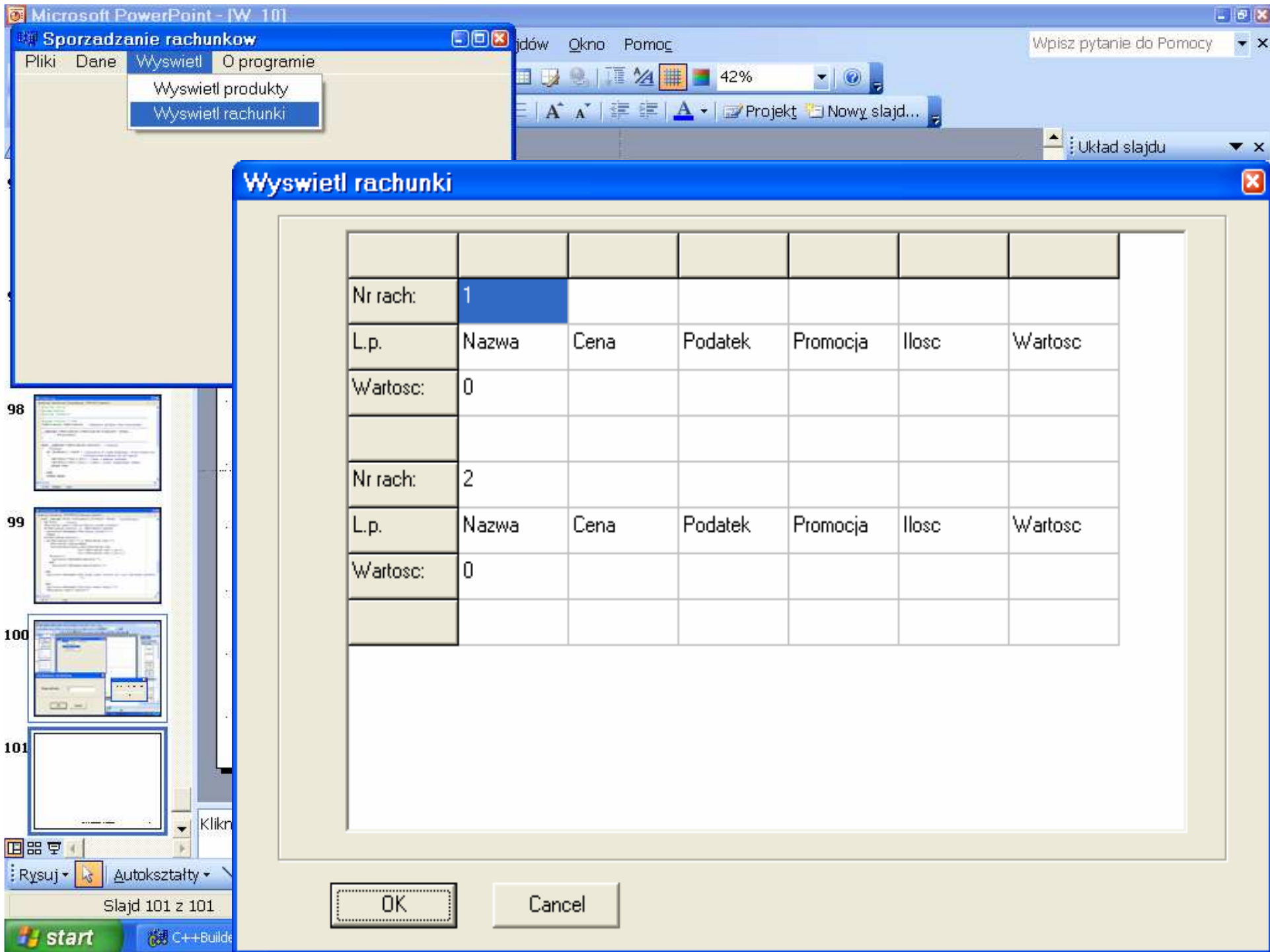


The screenshot shows a code editor window titled "RachunekApp.cpp". The window has a blue title bar and a tab bar with four tabs: "dprodukt.cpp", "drachunek.cpp", "RachunekApp.cpp" (highlighted with a red box), and "wrachunki.h". The main editing area contains the following C++ code:

```
void __fastcall TForm1::WyświetlRachunek2Click(TObject *Sender) //automatycznie
{
    OKBottomDlg3->Execute(); //recznie
}
```

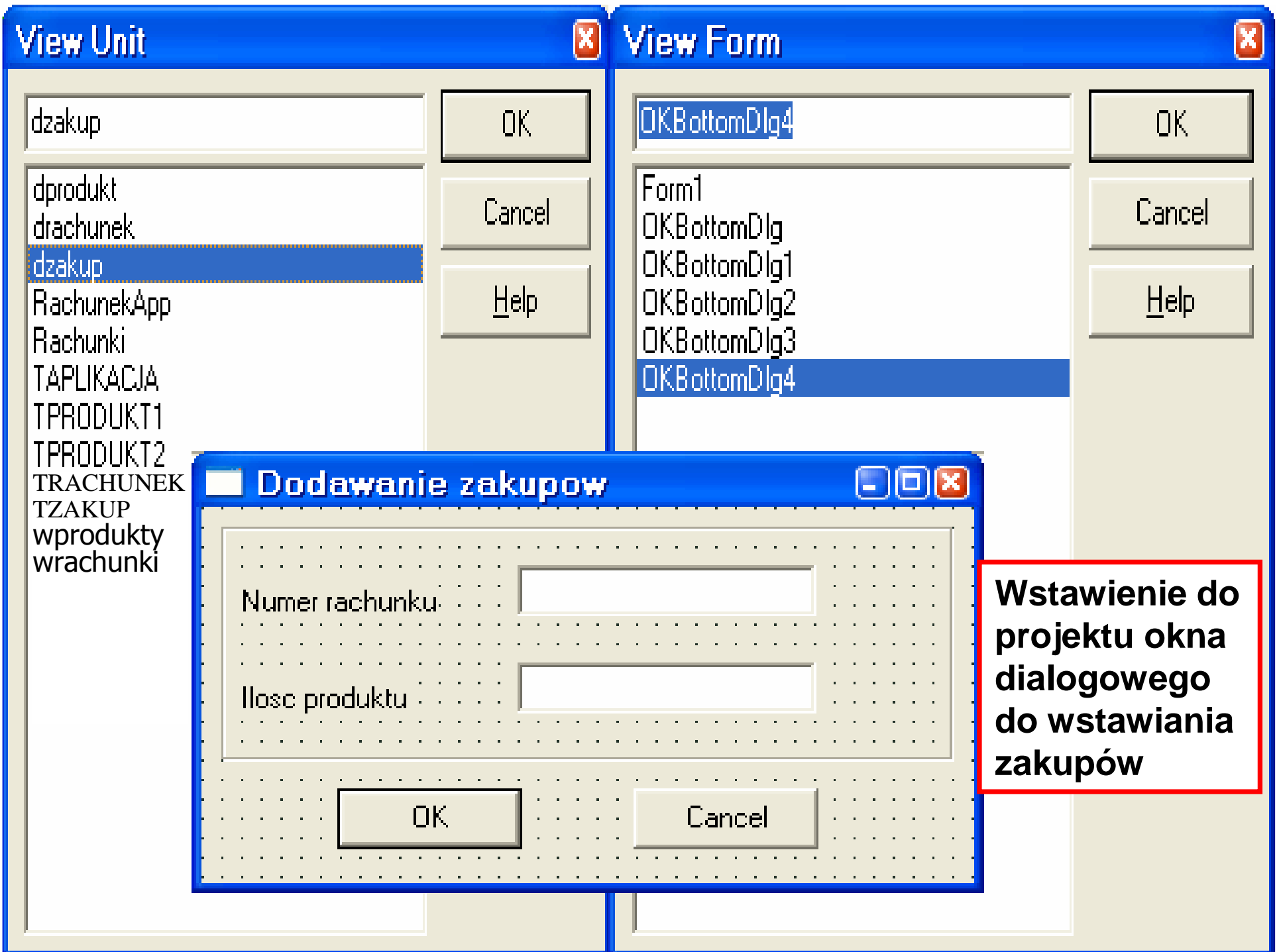
The status bar at the bottom shows the cursor position "46: 73" and the keyboard state "Insert".





# **Budowa aplikacji z graficznym interfejsem użytkownika - GUI (Graphic User Interface)**

- 1. Budowa okienek dialogowych do wyświetlania danych – wyświetlanie produktów**
- 2. Pobranie danych z wybranego wiersza tabeli wyświetlającej produkty**
- 3. Okienko dialogowe do wprowadzania rachunków**
- 4. Okienko dialogowe do wyświetlania rachunków**
- 5. Okienko dialogowe do wprowadzania zakupów**



**Wstawienie do projektu okna dialogowego do wstawiania zakupów**

```
dzakup.cpp
dprodukt.cpp | drachunek.cpp | RachunekApp.cpp | dzakup.h | wrachunki.h |
#define dzakupH
//-----
#include <vcl\System.hpp>
#include <vcl\Windows.hpp>
#include <vcl\SysUtils.hpp>
#include <vcl\Classes.hpp>
#include <vcl\Graphics.hpp>
#include <vcl\StdCtrls.hpp>
#include <vcl\Forms.hpp>
#include <vcl\Controls.hpp>
#include <vcl\Buttons.hpp>
#include <vcl\ExtCtrls.hpp>
#include "RachunekApp.h"
//-----
class TOKBottomDlg4 : public TForm
{
    __published:
        TButton *OKBtn;
        TButton *CancelBtn;
        TBevel *Bevel1;
        TEdit *Edit1;
        TEdit *Edit2;
        TLabel *Label1;
        TLabel *Label2;
    private:
    public: string tab1, tab2; //dane przekazywane do formularza
        //z numerem rachunku i iloscia towaru
        bool __fastcall Execute(); //recznie - obsluga okienka dialogowego
        virtual __fastcall TOKBottomDlg4(TComponent* AOwner);
};
//-----
extern PACKAGE TOKBottomDlg4 *OKBottomDlg4; //deklaracja globalnego wskaznika
//-----
#endif
```

```
#include <vcl.h>
#pragma hdrstop
#include "dzakup.h"
//-----
#pragma resource "*.dfm"
TOKBottomDlg4 *OKBottomDlg4; //wskaznik globalny okna dialogowego
//-----
__fastcall TOKBottomDlg4::TOKBottomDlg4(TComponent* AOwner)
    : TForm(AOwner)
{
}
//-----
bool __fastcall TOKBottomDlg4::Execute() //recznie
{
    //recznie
    if (ShowModal() ==mrOk ) //przejscie do trybu modalnego, który konczy sie
        {
            //nacisnieciem klawisza OK lub Cancel;
            tab1=Edit1->Text.c_str(); //dane o numerze rachunku
            tab2=Edit2->Text.c_str(); //dane o ilosci zakupionego towaru
            return true;
        }
    else
        return false;
}
```

**Definicja metody `Execute()`  
pobierającej numer rachunku oraz  
ilość zakupionego produktu**

```
• void __fastcall TForm1::Dodajzakup1Click(TObject *Sender) //auto
{ int wynik; //recznie
• OKBottomDlg2->Label1->Caption="Wybierz produkt klikajac";
• if (OKBottomDlg2->Execute() && !OKBottomDlg2->wybrany)
• { Application->MessageBox("Nie wybrano produktu","");
• return; }
• if (OKBottomDlg4->Execute())
• { if (OKBottomDlg4->tab1!="" && OKBottomDlg4->tab2!="")
• { OKBottomDlg2->wybrany=false;
• wynik=aplikacja.Wstaw_zakup(OKBottomDlg2->tab,
• atoi((OKBottomDlg4->tab2).c_str()),
• atoi((OKBottomDlg4->tab1).c_str()));
•
• if (wynik==1)
• Application->MessageBox(napisy[6],"");
• else
• Application->MessageBox(napisy[wynik],"");
•
• }
• else
• Application->MessageBox("Nie podano numeru rachunku lub ilosci produktu",
• "");
•
• }
• else
• Application->MessageBox("Anulowano dodanie zakupu","");
• OKBottomDlg2->Label1->Caption="";
•
• }
```

- Definicja metody w pliku RachunekApp.cpp po kliknięciu na „Dodaj zakup”, która pobiera kliknięty wiersz z produktem z okien dialogowych:
- wyświetlający produkty
- wstawiający numer rachunku i ilość produktu

**Sporządzanie rachunków**

Pliki Dane Wyświetl O programie

- Dodaj produkt
- Dodaj rachunek
- Dodaj zakup**

**Wyświetl produkty**

L.p.	Nazwa	Cena	Podatek	Promocja
1	1	1.01	1	
2	2	2		

Wybierz produkt klikając

OK Odśwież Cancel

**Dodawanie zakupów**

Numer rachunku

Ilość produktu

OK Cancel

**Wstawiono dane**

OK

Wyswietl rachunki



Nr rach:	1					
L.p.	Nazwa	Cena	Podatek	Promocja	Ilosc	Wartosc
1	1	1.01	1		2	2.02
Wartosc:	2.02					
Nr rach:	2					
L.p.	Nazwa	Cena	Podatek	Promocja	Ilosc	Wartosc
Wartosc:	0					

OK

Cancel



## Wyswietl rachunki



Lp.	Nazwa	Cena	Podatek	Promocja	Ilosc	Wartosc
Nr rach:	1					
Lp.	Nazwa	Cena	Podatek	Promocja	Ilosc	Wartosc
1	1	1.01	1		3	3.03
2	2	2			8	16
Wartosc:	19.03					
Nr rach:	2					
Lp.	Nazwa	Cena	Podatek	Promocja	Ilosc	Wartosc
1	2	2			5	10
2	1	1.01	1		4	4.04
Wartosc:	14.04					

OK

Cancel