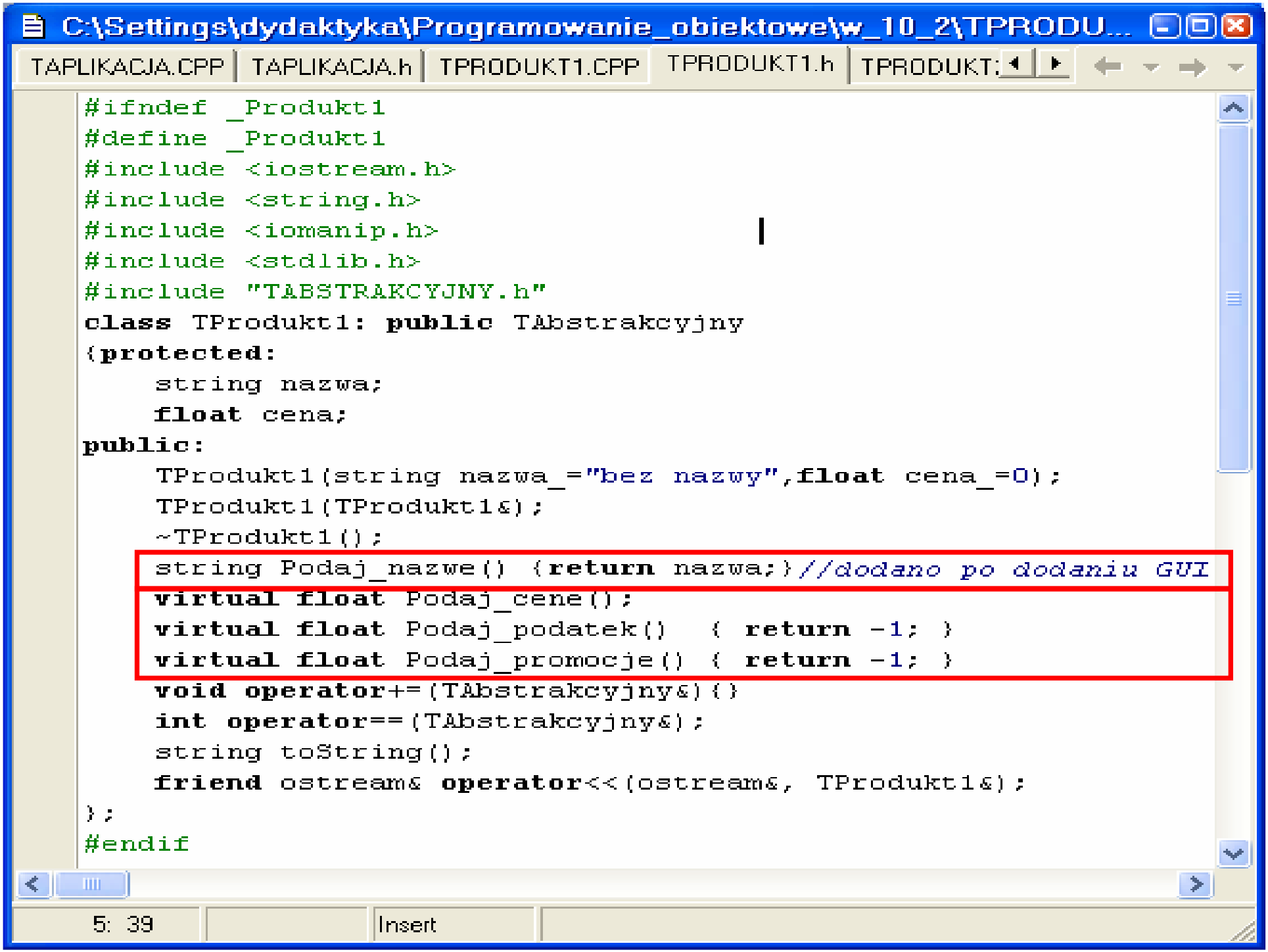
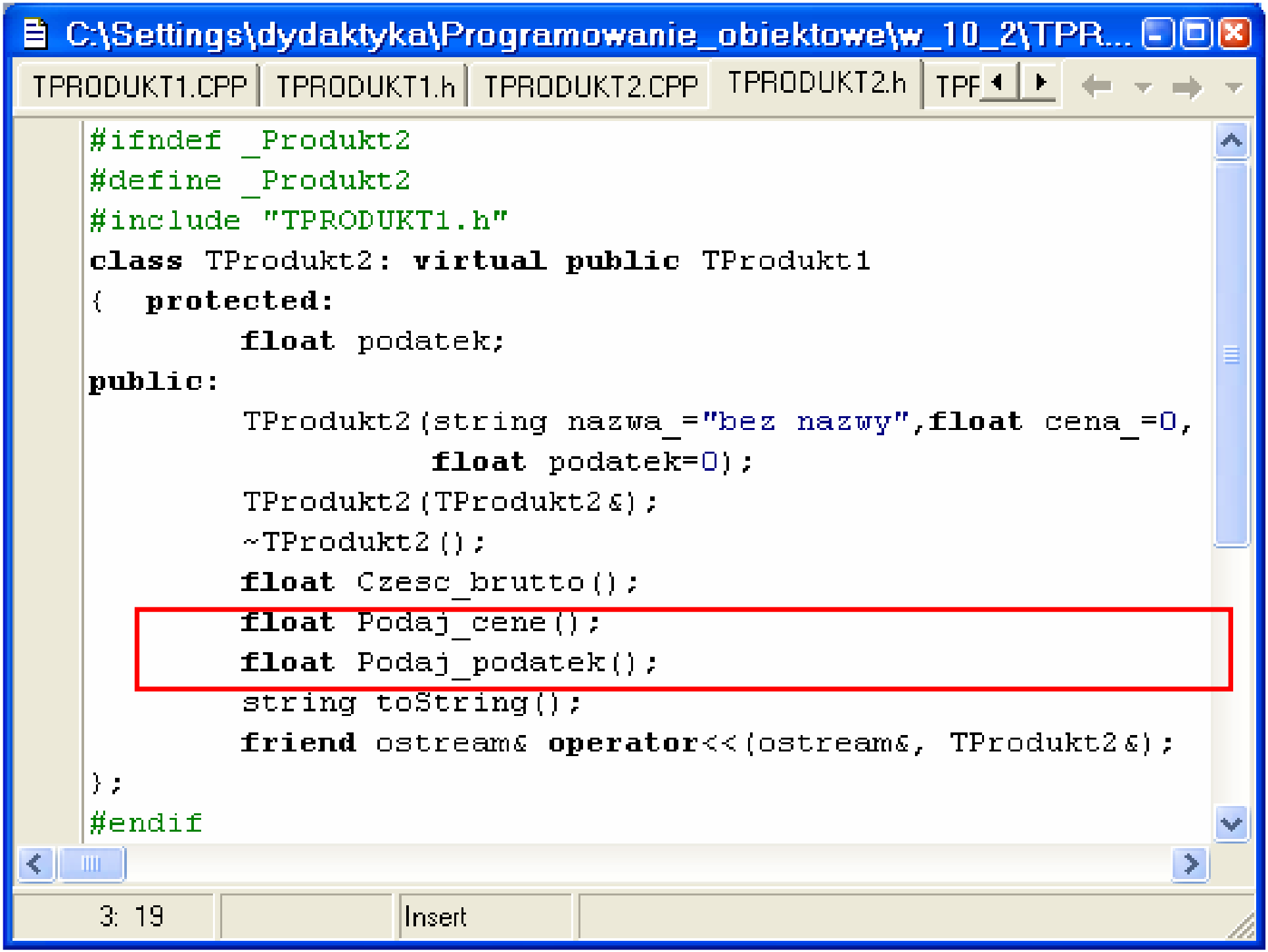
Budowa aplikacji z graficznym interfejsem użytkownika - GUI (Graphic User Interface)

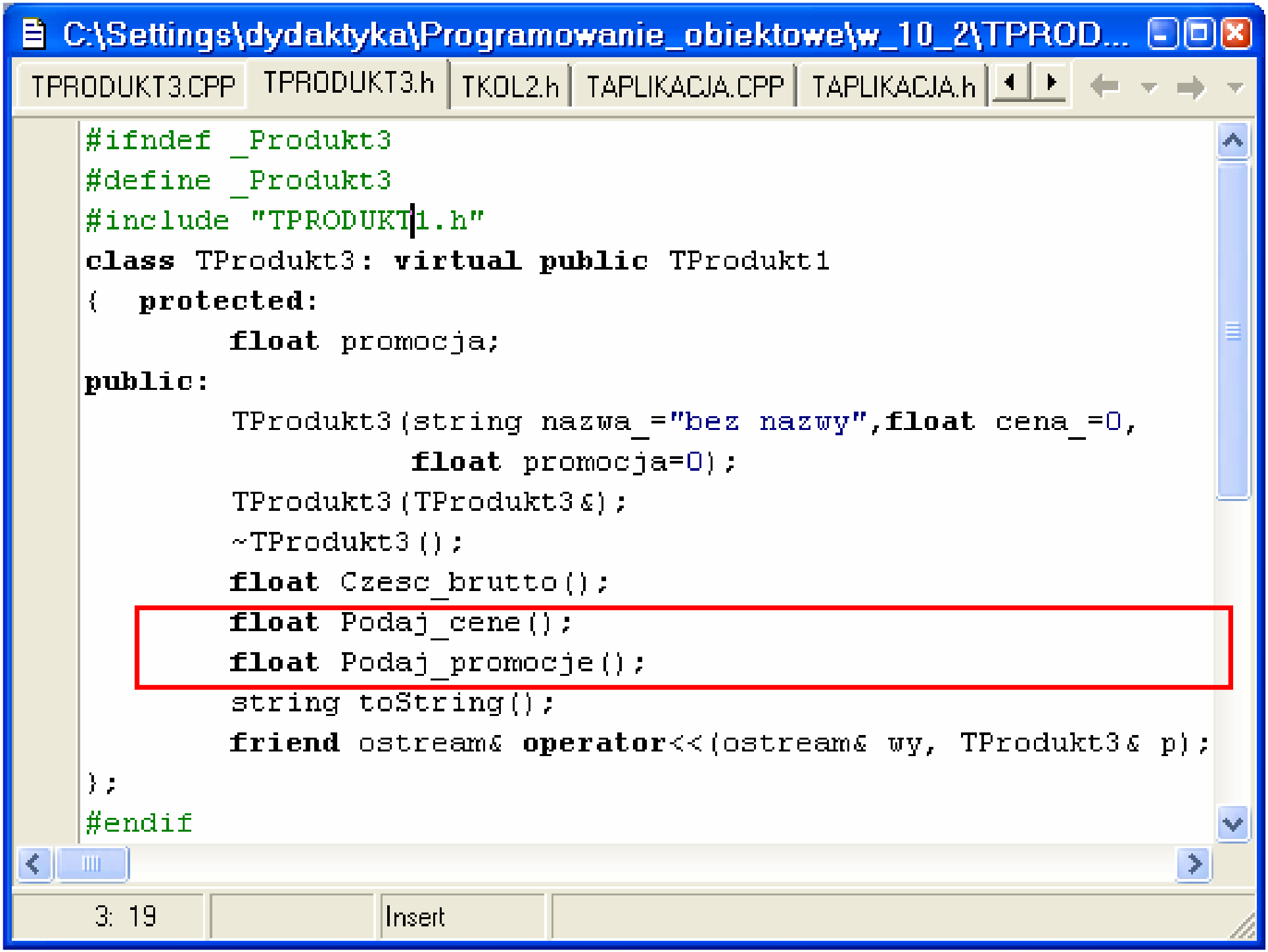
1. **Udostępnianie wszystkich prywatnych atrybutów do prezentacji, wprowadzenie standardu nazewnictwa plików – nazwy plików aplikacji poprzedzone literą T**
2. **Budowa głównego formularza GUI**
3. **Budowa okienek dialogowych do wprowadzania danych**
4. **Budowa okienek dialogowych do wyświetlania danych**
5. **Aplikacja do sporządzania rachunków - uzupełnienie**

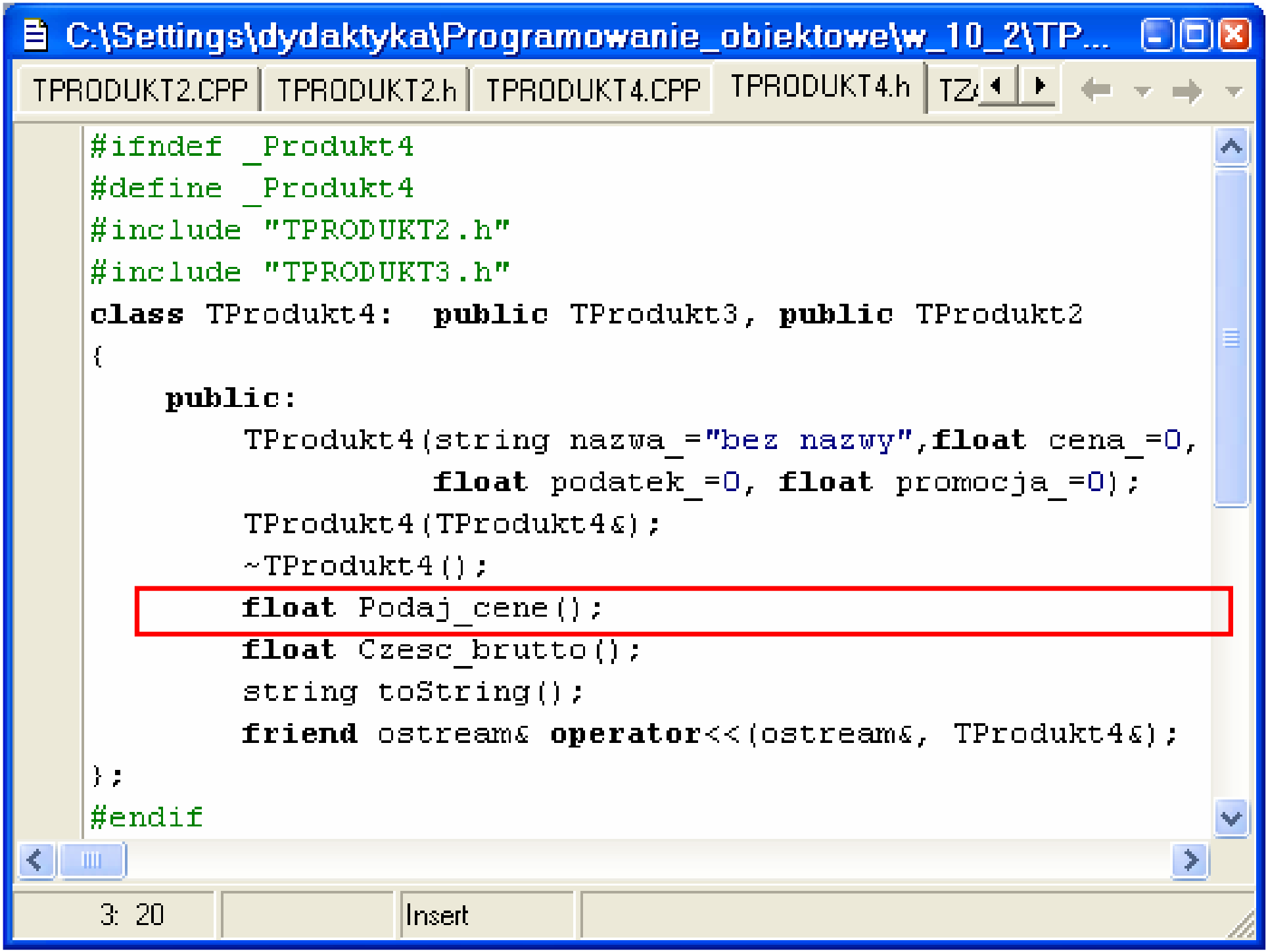
Budowa aplikacji z graficznym interfejsem użytkownika - GUI (Graphic User Interface)

1. **Udostępnianie wszystkich prywatnych atrybutów do prezentacji, wprowadzenie standardu nazewnictwa plików – nazwy plików aplikacji poprzedzone literą T**


C:\Settings\dydaktyka\Programowanie_obiektowe\w_10_2\TPRODU...
TAPLIKACJA.CPP
TAPLIKACJA.h
TPRODUKT1.CPP
TPRODUKT1.h
TPRODUKT:
#ifndef _Produkt1
#define _Produkt1
#include <iostream.h>
#include <string.h>
#include <iomanip.h>
#include <stdlib.h>
#include "TABSTRAKCYJNY.h"
class TProdukt1: public TAbstrakcyjny
{protected:
string nazwa;
float cena;
public:
TProdukt1(string nazwa_="bez nazwy",float cena_=0);
TProdukt1(TProdukt1&);
~TProdukt1();
string Podaj_nazwe() {return nazwa;}//dodano po dodaniu GUI
virtual float Podaj_cene();
virtual float Podaj_podatek() { return -1; }
virtual float Podaj_promocje() { return -1; }
void operator+=(TAbstrakcyjny&){}
int operator==(TAbstrakcyjny&);
string toString();
friend ostream& operator<<(ostream&, TProdukt1&);
};
#endif
5: 39
Insert


C:\Settings\dydaktyka\Programowanie_obiektowe\w_10_2\TPR...
TPRODUKT1.CPP
TPRODUKT1.h
TPRODUKT2.CPP
TPRODUKT2.h
TPF
#ifndef _Produkt2
#define _Produkt2
#include "TPRODUKT1.h"
class TProdukt2: virtual public TProdukt1
{ protected:
float podatek;
public:
TProdukt2(string nazwa_="bez nazwy",float cena_=0,
float podatek=0);
TProdukt2(TProdukt2&);
~TProdukt2();
float Czesc_brutto();
float Podaj_cene();
float Podaj_podatek();
string toString();
friend ostream& operator<<(ostream&, TProdukt2&);
};
#endif
3: 19
Insert


C:\Settings\dydaktyka\Programowanie_obiektowe\w_10_2\TPROD...
TPRODUKT3.CPP
TPRODUKT3.h
TKOL2.h
TAPLIKACJA.CPP
TAPLIKACJA.h
#ifndef _Produkt3
#define _Produkt3
#include "TPRODUKT1.h"
class TProdukt3: virtual public TProdukt1
{ protected:
float promocja;
public:
TProdukt3(string nazwa_="bez nazwy",float cena_=0,
float promocja=0);
TProdukt3(TProdukt3&);
~TProdukt3();
float Czesc_brutto();
float Podaj_cene();
float Podaj_promocje();
string toString();
friend ostream& operator<<(ostream& wy, TProdukt3& p);
};
#endif
3: 19
Insert


C:\Settings\dydaktyka\Programowanie_obiektowe\w_10_2\TP...
TPRODUKT2.CPP
TPRODUKT2.h
TPRODUKT4.CPP
TPRODUKT4.h
#ifndef _Produkt4
#define _Produkt4
#include "TPRODUKT2.h"
#include "TPRODUKT3.h"
class TProdukt4: public TProdukt3, public TProdukt2
{
public:
TProdukt4(string nazwa_="bez nazwy",float cena_=0,
float podatek_=0, float promocja_=0);
TProdukt4(TProdukt4&);
~TProdukt4();
float Podaj_cene();
float Czesc_brutto();
string toString();
friend ostream& operator<<(ostream&, TProdukt4&);
};
#endif
3: 20
Insert

C:\Settings\dydaktyka\Programowanie_obiektowe\w_10_2\TZAKUP.h

TPRODUKT4.CPP | TPRODUKT4.h | TZAKUP.CPP | TZAKUP.h | TRACHUNEK.CPP | TRACHUNEK.I

```
#ifndef _Zakup
#define _Zakup
#include "TPRODUKT1.h"
class TZakup: public TAbstrakcyjny
 {
   protected:
   TProdukt1* produkt;
   float ilosc;
public:
   TZakup(TProdukt1*produkt_=NULL,float ilosc_=0);
   TZakup(TZakup&);
   ~TZakup();
   float Podaj_wartosc();
   float Podaj_ilosc();
   TProdukt1* Podaj_produkt() { return produkt;} //dodano po dodaniu GUI
   void operator+=(TAbstrakcyjny&);
   int operator==(TAbstrakcyjny&);
   string toString();
   friend ostream& operator<<(ostream&, TZakup&);
};
#endif
```

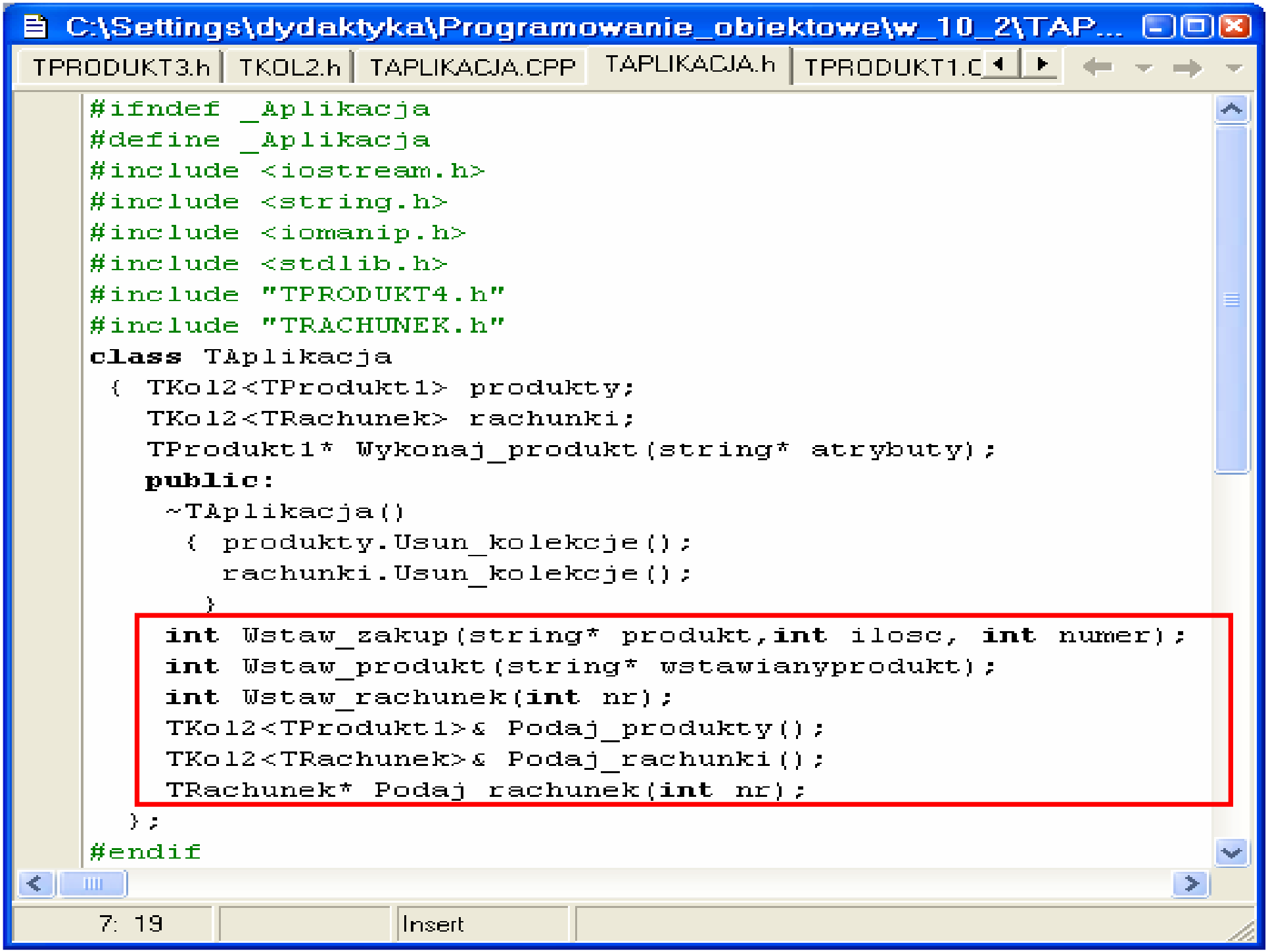
3: 20 | Insert

C:\Settings\dydaktyka\Programowanie_obiektowe\w_10_2\TRACHUNEK.h

TPRODUKT4.h | TZAKUP.CPP | TZAKUP.h | TRACHUNEK.CPP | TRACHUNEK.h

```
#ifndef _Rachunek
#define _Rachunek
#include "TZAKUP.h"
#include "TKOL2.h"
class TRachunek:public TAbstrakcyjny
{protected:
   int numer;
   TKol2<TZakup> zakupy;
   void kopia(TRachunek& r);
public:
   TRachunek(int=0);
   TRachunek(TRachunek& r);
   ~TRachunek();
   TKol2<TZakup>& Podaj_zakupy() { return zakupy;} //dodano po dodaniu GUI
   int Podaj_numer()             { return numer; } //dodano po dodaniu GUI
   void operator=(TRachunek&);
   float Podaj_wartosc();
   int Dodaj_zakup(TZakup*);
   void operator+=(TAbstrakcyjny&) {}
   int operator==(TAbstrakcyjny&);
   string toString();
   friend ostream& operator<<(ostream&, TRachunek&);
};
#endif
```

3: 17 | Insert

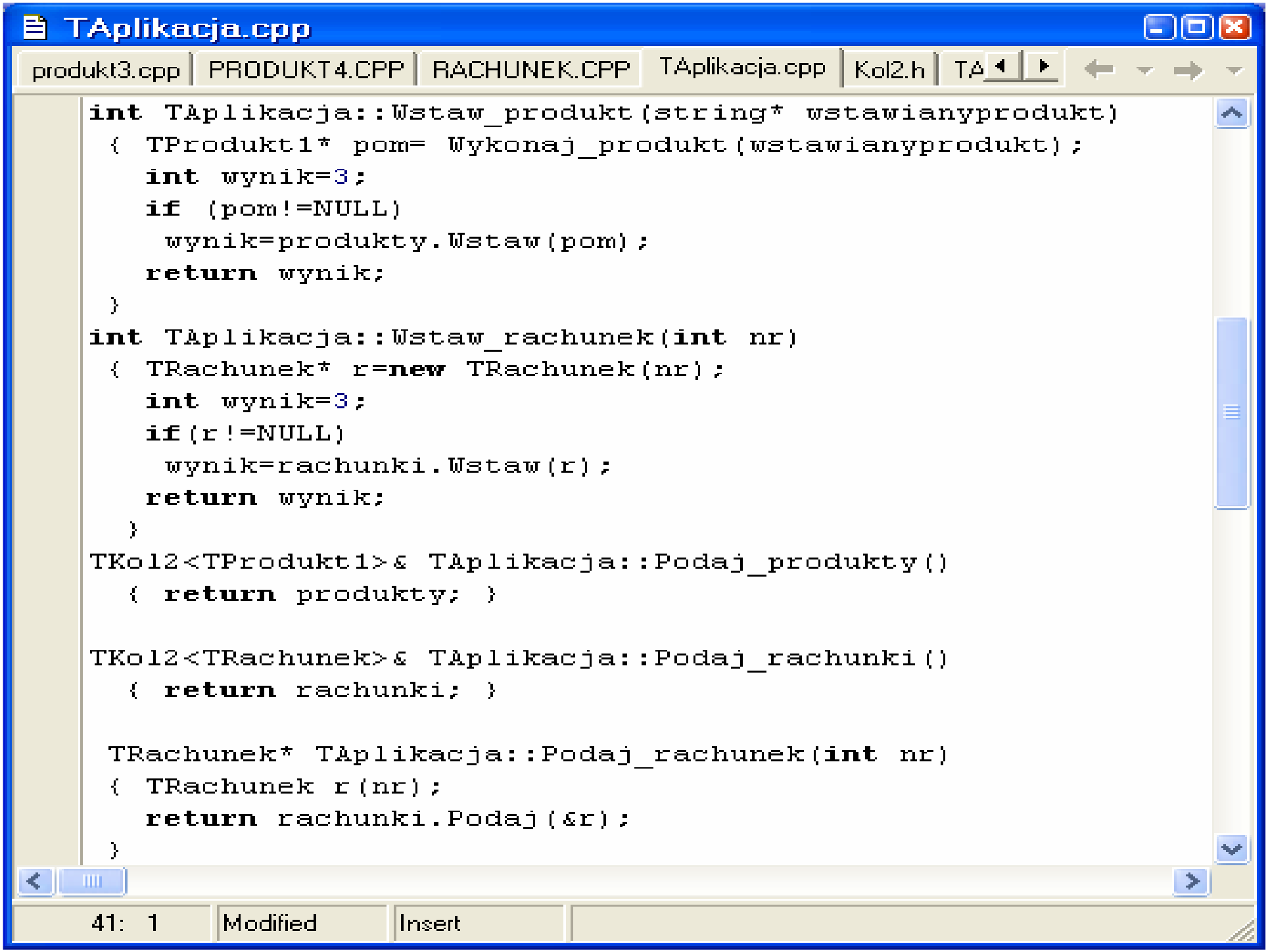

C:\Settings\dydaktyka\Programowanie_obiektowe\w_10_2\TAP...
TPRODUKT3.h
TKOL2.h
TAPLIKACJA.CPP
TAPLIKACJA.h
TPRODUKT1.C
#ifndef _Aplikacja
#define _Aplikacja
#include <iostream.h>
#include <string.h>
#include <iomanip.h>
#include <stdlib.h>
#include "TPRODUKT4.h"
#include "TRACHUNEK.h"
class TAplikacja
{ TKol2<TProdukt1> produkty;
TKol2<TRachunek> rachunki;
TProdukt1* Wykonaj_produkt(string* atrybuty);
public:
~TAplikacja()
{ produkty.Usun_kolekcje();
rachunki.Usun_kolekcje();
}
int Wstaw_zakup(string* produkt,int ilosc, int numer);
int Wstaw_produkt(string* wstawianyprodukt);
int Wstaw_rachunek(int nr);
TKol2<TProdukt1>& Podaj_produkty();
TKol2<TRachunek>& Podaj_rachunki();
TRachunek* Podaj_rachunek(int nr);
};
#endif
7: 19
Insert

TAplikacja.cpp

produkt3.cpp | PRODUKT4.CPP | RACHUNEK.CPP | TAplikacja.cpp | Kol2.h | TAplikacja.h | ZAKUP.CPP

```
#include "TAplikacja.h"
int TAplikacja::Wstaw_zakup(string* produkt, int ilosc, int numer)
{   int wynik;
    TRachunek* poszukiwanyrachunek=new TRachunek(numer);
    TProdukt1* poszukiwanyprodukt=Wykonaj_produkt(produkt);
    if (poszukiwanyrachunek!=NULL && poszukiwanyprodukt!=NULL)
     { TRachunek* znalezionyrachunek;
       znalezionyrachunek=rachunki.Podaj(poszukiwanyrachunek);
       if (znalezionyrachunek!=NULL)
        { TProdukt1* znalezionyprodukt=
              produkty.Podaj(poszukiwanyprodukt);
          if(znalezionyprodukt!=NULL)
            wynik=
             znalezionyrachunek->Dodaj_zakup(new TZakup(znalezionyprodukt,ilosc));
          else
           wynik=5;
        }
       else
        wynik=4;
     }
    else
       wynik=3;
    delete poszukiwanyrachunek;
    delete poszukiwanyprodukt;
    return wynik;
 }
```

22: 15 | Modified | Insert


TAplikacja.cpp
produkt3.cpp
PRODUKT4.CPP
RACHUNEK.CPP
TAplikacja.cpp
Kol2.h
TA
int TAplikacja::Wstaw_produkt(string* wstawianyprodukt)
{ TProdukt1* pom= Wykonaj_produkt(wstawianyprodukt);
int wynik=3;
if (pom!=NULL)
wynik=produkty.Wstaw(pom);
return wynik;
}
int TAplikacja::Wstaw_rachunek(int nr)
{ TRachunek* r=new TRachunek(nr);
int wynik=3;
if(r!=NULL)
wynik=rachunki.Wstaw(r);
return wynik;
}
TKol2<TProdukt1>& TAplikacja::Podaj_produkty()
{ return produkty; }
TKol2<TRachunek>& TAplikacja::Podaj_rachunki()
{ return rachunki; }
TRachunek* TAplikacja::Podaj_rachunek(int nr)
{ TRachunek r(nr);
return rachunki.Podaj(&r);
}
41: 1
Modified
Insert

TAplikacja.cpp

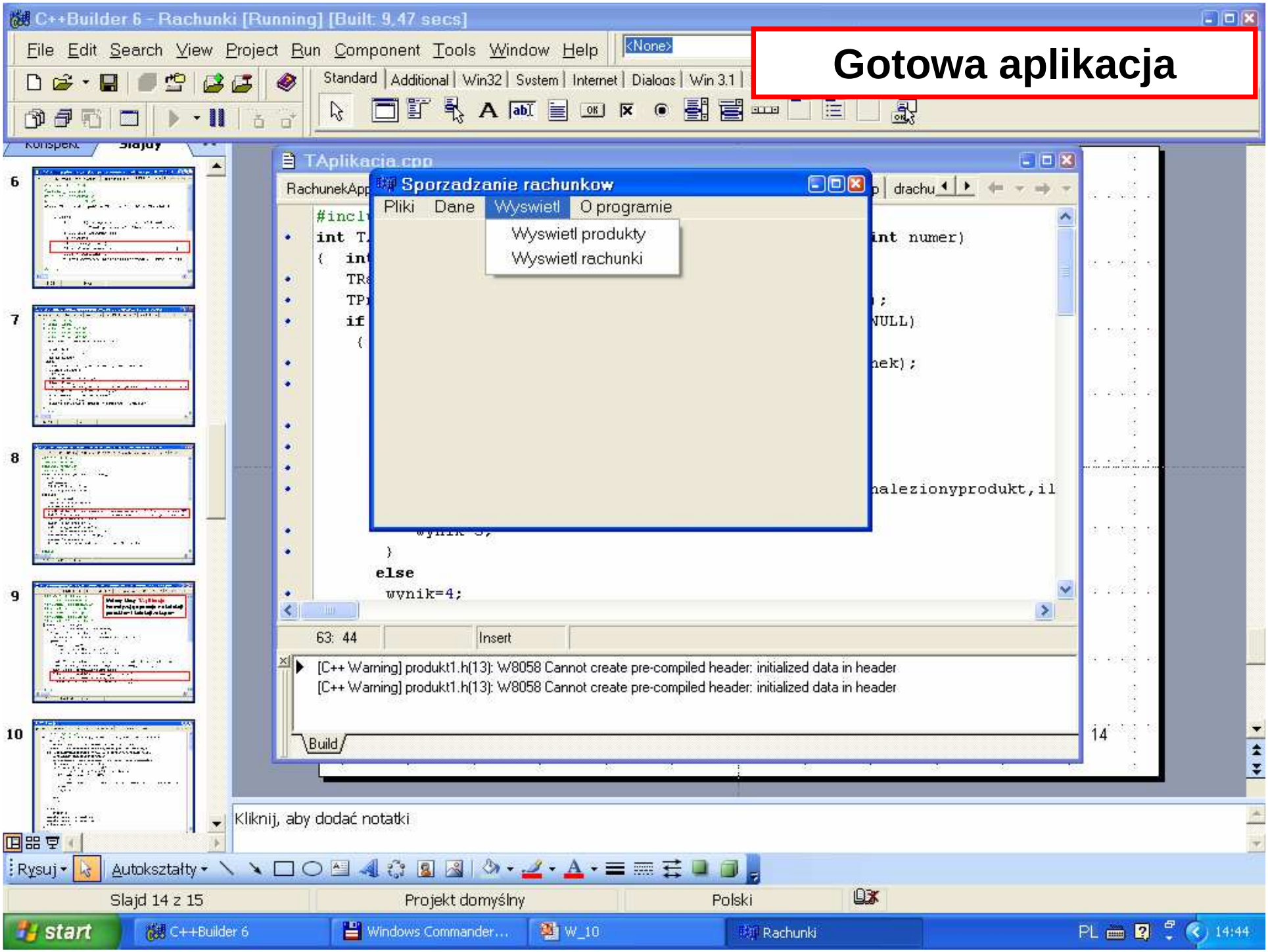
produkt3.cpp | PRODUKT4.CPP | RACHUNEK.CPP | TAplikacja.cpp | TAplikacja.h | ZAKUP.CPP

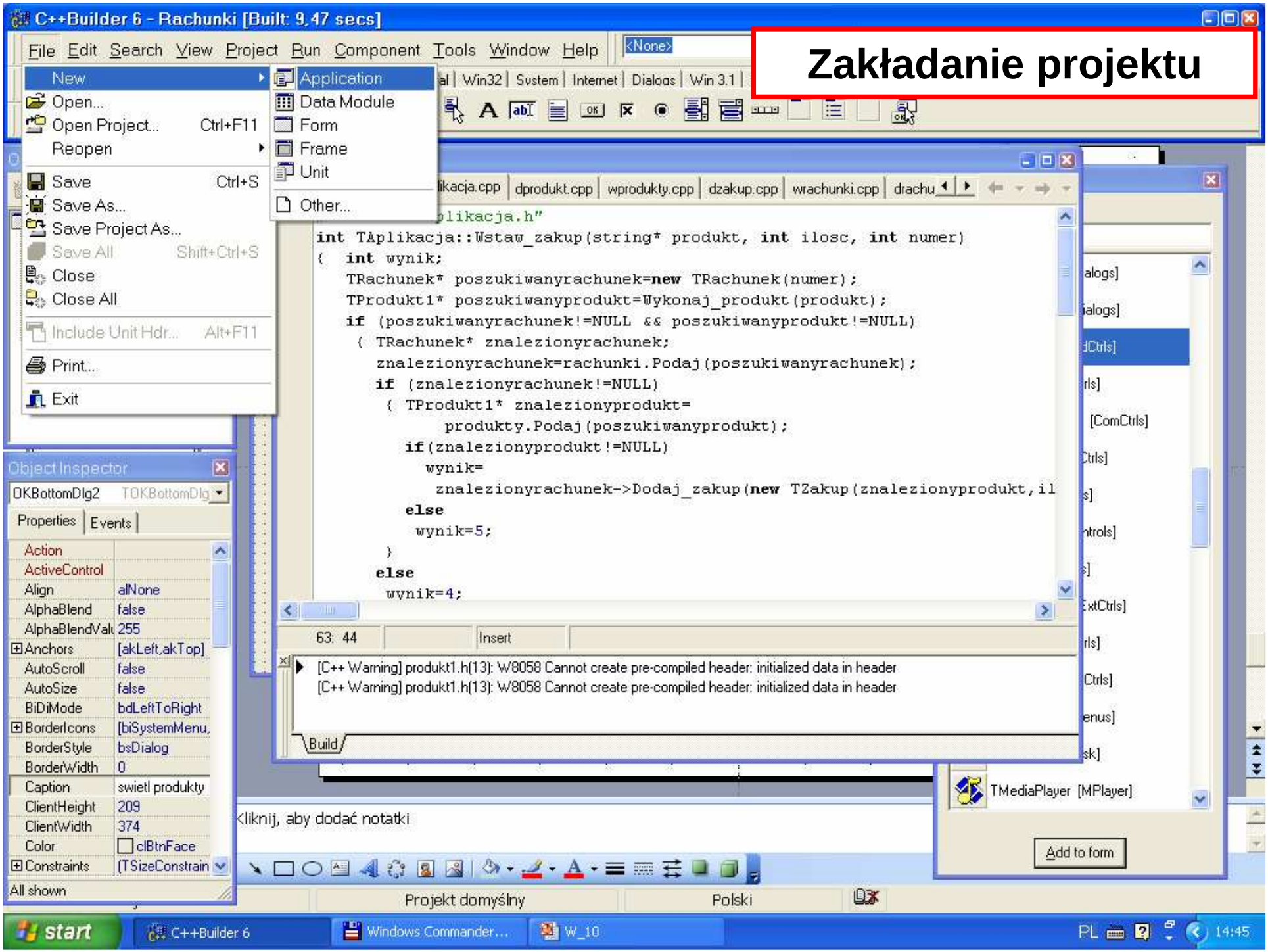
```
TProdukt1* TAplikacja::Wykonaj_produkt(string* atrybuty)
 { TProdukt1* produkt=NULL;
   if (atrybuty[0]=="1")
    produkt=new TProdukt1(atrybuty[1], atof(atrybuty[2].c_str()));
   else
    if (atrybuty[0]=="2")
     produkt=new TProdukt2(atrybuty[1], atof(atrybuty[2].c_str()),
                           atof(atrybuty[3].c_str()));
    else
     if (atrybuty[0]=="3")
      produkt=new TProdukt3(atrybuty[1], atof(atrybuty[2].c_str()),
                            atof(atrybuty[3].c_str()));
     else
       if (atrybuty[0]=="4")
         produkt=new TProdukt4(atrybuty[1], atof(atrybuty[2].c_str()),
                               atof(atrybuty[3].c_str()),
                               atof(atrybuty[4].c_str()));
     return produkt;
 }
```

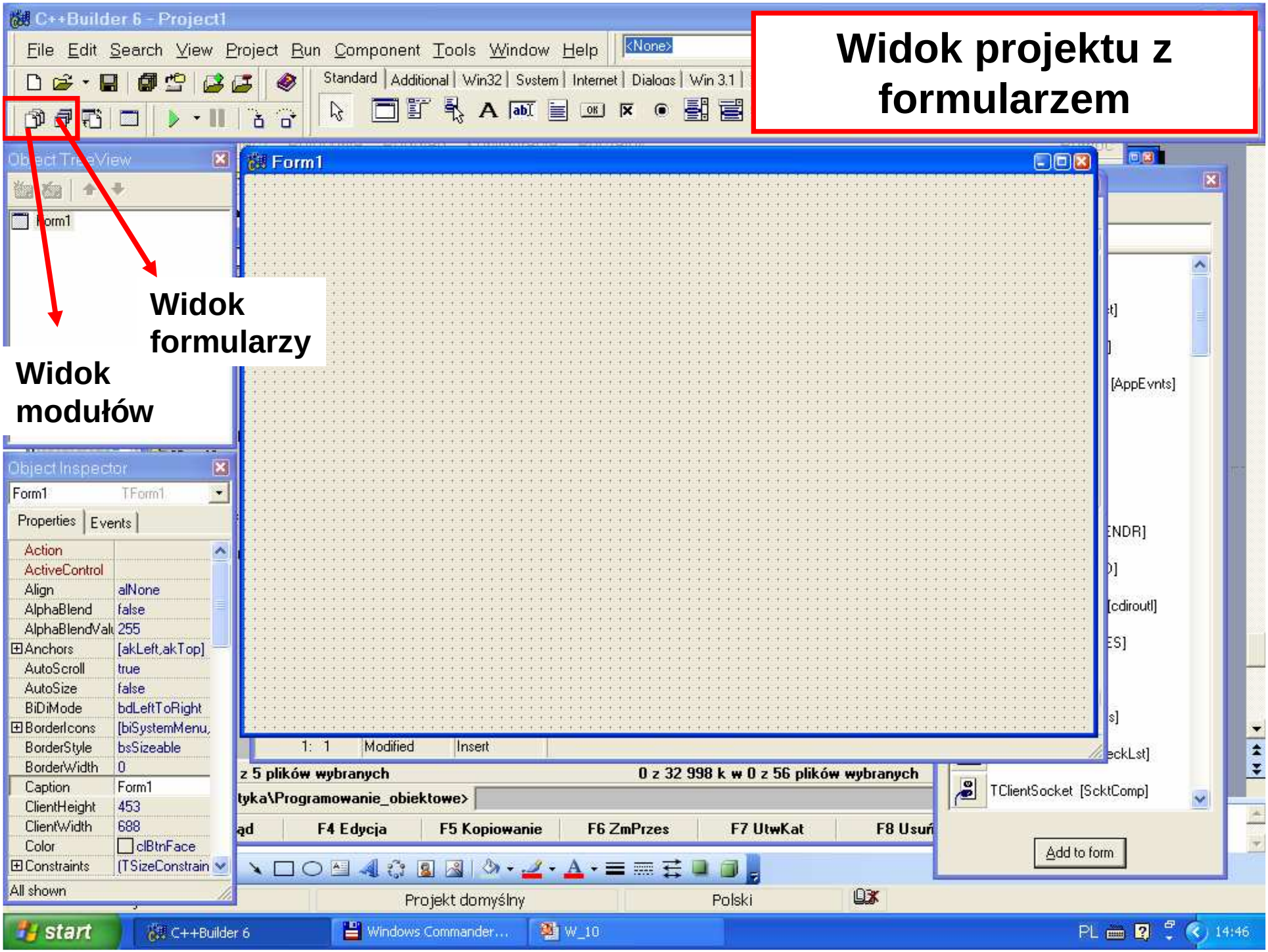
60: 65 | Modified | Insert

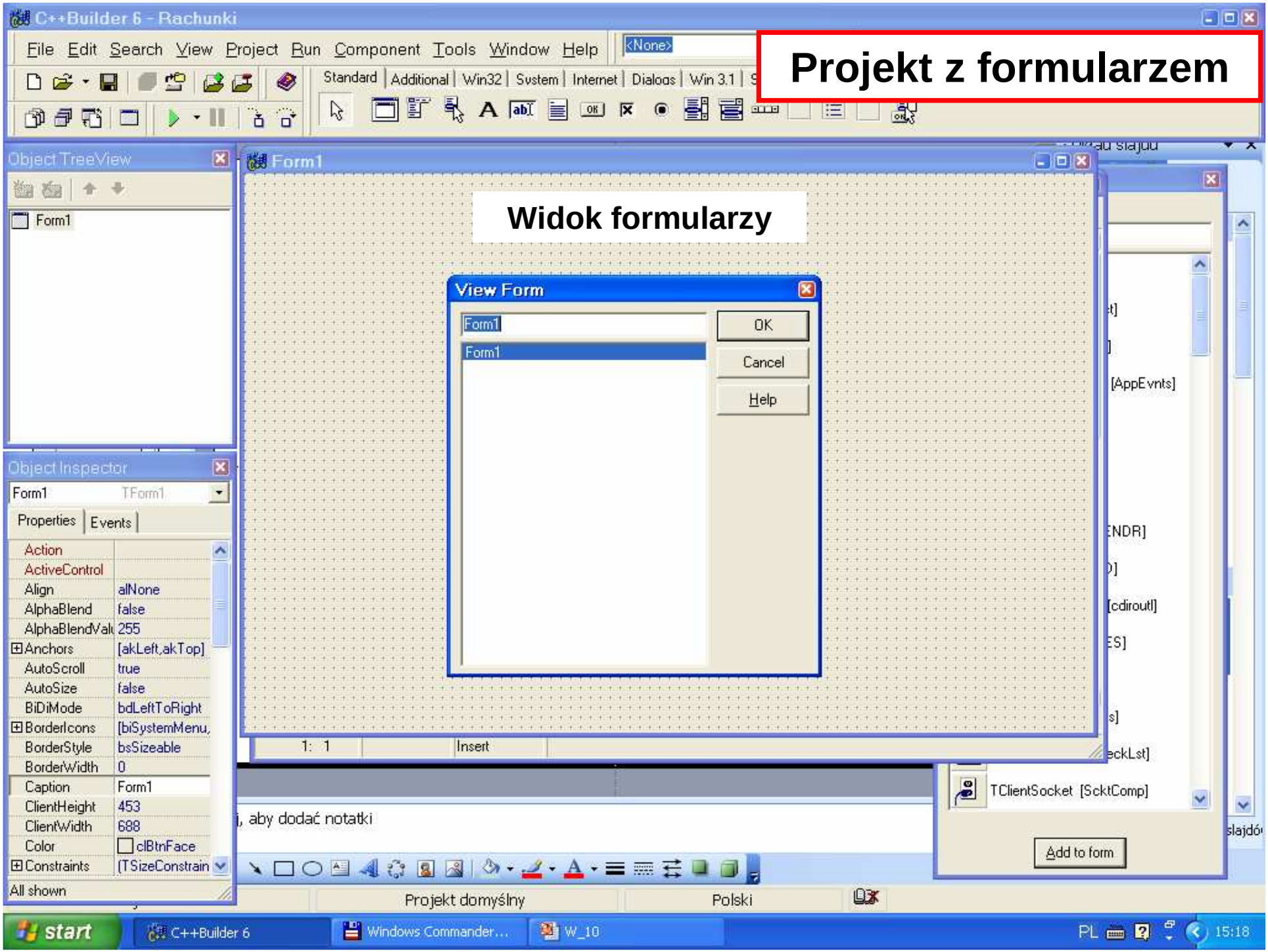
Budowa aplikacji z graficznym interfejsem użytkownika - GUI (Graphic User Interface)

1. **Udostępnianie wszystkich prywatnych atrybutów do prezentacji, wprowadzenie standardu nazewnictwa plików – nazwy plików aplikacji poprzedzone literą T**
2. **Budowa głównego formularza GUI**


Gotowa aplikacja
C++Builder 6 - Rachunki [Running] [Built: 9,47 secs]
File Edit Search View Project Run Component Tools Window Help
Standard Additional Win32 System Internet Dialogs Win 3.1
TAplikacja.cpp
Sporzadzanie rachunkow
Pliki Dane Wyswietl O programie
Wyswietl produkty
Wyswietl rachunki
int numer)
NULL)
nalezionyprodukt,il
else
wynik=4;
63: 44
Insert
[C++ Warning] produkt1.h(13): W8058 Cannot create pre-compiled header: initialized data in header
[C++ Warning] produkt1.h(13): W8058 Cannot create pre-compiled header: initialized data in header
Build
Kliknij, aby dodać notatki
Rysuj Autokształty
Slajd 14 z 15
Projekt domyślny
Polski
start
C++Builder 6
Windows Commander...
W_10
Rachunki
PL
14:44


C++Builder 6 - Rachunki [Built: 9,47 secs]
File Edit Search View Project Run Component Tools Window Help
New
Open...
Open Project... Ctrl+F11
Reopen
Save Ctrl+S
Save As...
Save Project As...
Save All Shift+Ctrl+S
Close
Close All
Include Unit Hdr... Alt+F11
Print...
Exit
Application
Data Module
Form
Frame
Unit
Other...
Zakładanie projektu
dprodukt.cpp
wprodukty.cpp
dzakup.cpp
wrachunki.cpp
int TAplikacja::Wstaw_zakup(string* produkt, int ilosc, int numer)
{ int wynik;
TRachunek* poszukiwanyrachunek=new TRachunek(numer);
TProdukt1* poszukiwanyprodukt=Wykonaj_produkt(produkt);
if (poszukiwanyrachunek!=NULL && poszukiwanyprodukt!=NULL)
{ TRachunek* znalezionyrachunek;
znalezionyrachunek=rachunki.Podaj(poszukiwanyrachunek);
if (znalezionyrachunek!=NULL)
{ TProdukt1* znalezionyprodukt=
produkty.Podaj(poszukiwanyprodukt);
if(znalezionyprodukt!=NULL)
wynik=
znalezionyrachunek->Dodaj_zakup(new TZakup(znalezionyprodukt,il
else
wynik=5;
}
else
wynik=4;
63: 44
Insert
[C++ Warning] produkt1.h(13): W8058 Cannot create pre-compiled header: initialized data in header
[C++ Warning] produkt1.h(13): W8058 Cannot create pre-compiled header: initialized data in header
Build
Object Inspector
OKBottomDlg2
Properties
Events
Action
ActiveControl
Align alNone
AlphaBlend false
Anchors [akLeft,akTop]
AutoScroll false
AutoSize false
BiDiMode bdLeftToRight
BorderIcons [biSystemMenu,
BorderStyle bsDialog
BorderWidth 0
Caption swietl produkty
ClientHeight 209
ClientWidth 374
Color clBtnFace
Constraints (TSizeConstrain
All shown
Kliknij, aby dodać notatki
TMediaPlayer [MPlayer]
Add to form
Projekt domyślny
Polski
start
C++Builder 6
Windows Commander...
W_10
PL
14:45

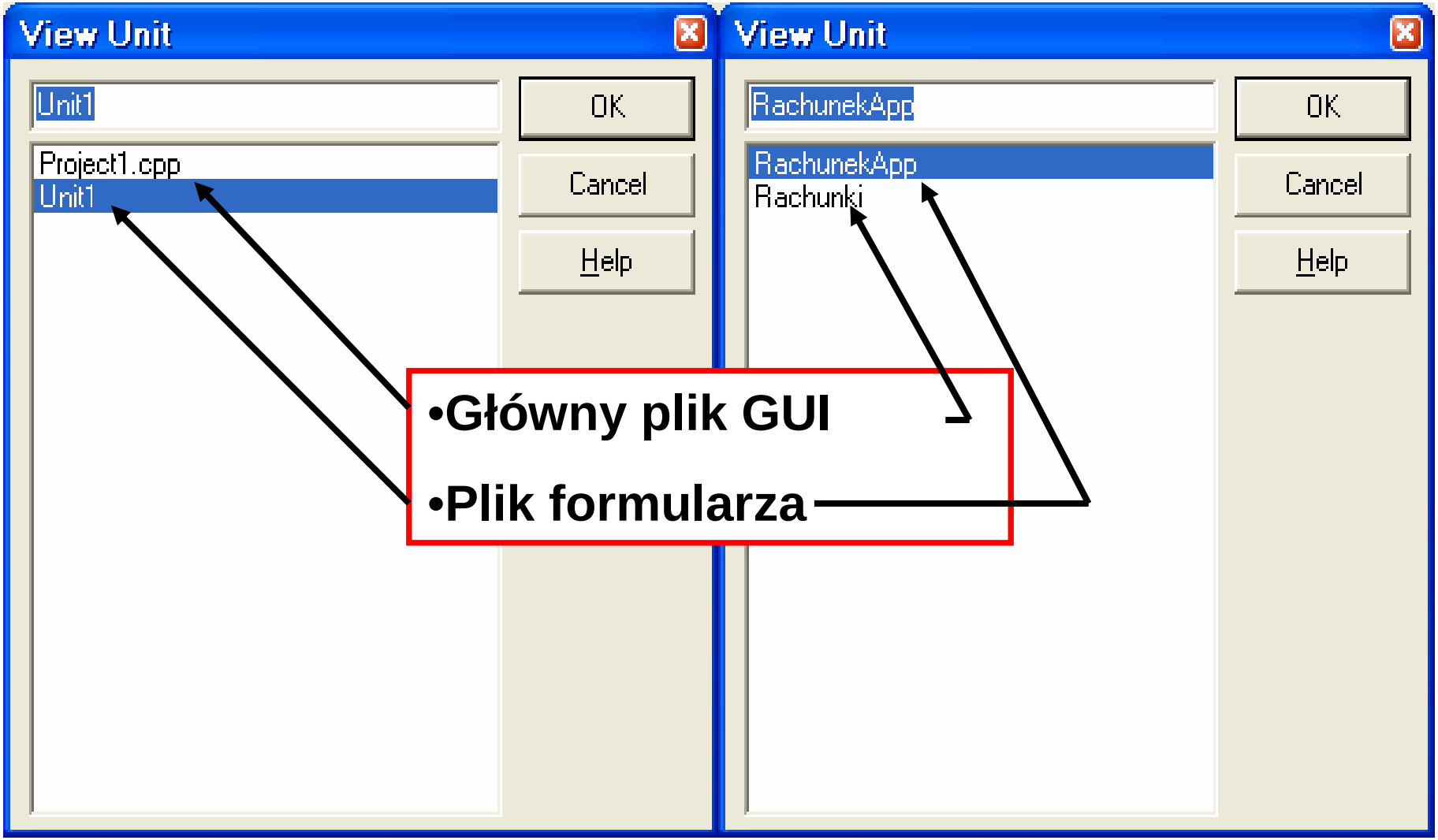

C++Builder 6 - Project1
File Edit Search View Project Run Component Tools Window Help
Standard Additional Win32 System Internet Dialogs Win 3.1
Widok projektu z formularzem
Form1
Widok formularzy
Widok modułów
Object Inspector
Form1 TForm1
Properties Events
Action
ActiveControl
Align alNone
AlphaBlend false
AlphaBlendVal 255
Anchors [akLeft,akTop]
AutoScroll true
AutoSize false
BiDiMode bdLeftToRight
BorderIcons [biSystemMenu,
BorderStyle bsSizeable
BorderWidth 0
Caption Form1
ClientHeight 453
ClientWidth 688
Color clBtnFace
Constraints (TSizeConstrain
All shown
1: 1 Modified Insert
z 5 plików wybranych
0 z 32 998 k w 0 z 56 plików wybranych
F4 Edycja
F5 Kopiowanie
F6 ZmPrzes
F7 UtwKat
TClientSocket [ScktComp]
Add to form
Projekt domyślny
Polski
start
C++Builder 6
Windows Commander...
W_10
PL
14:46

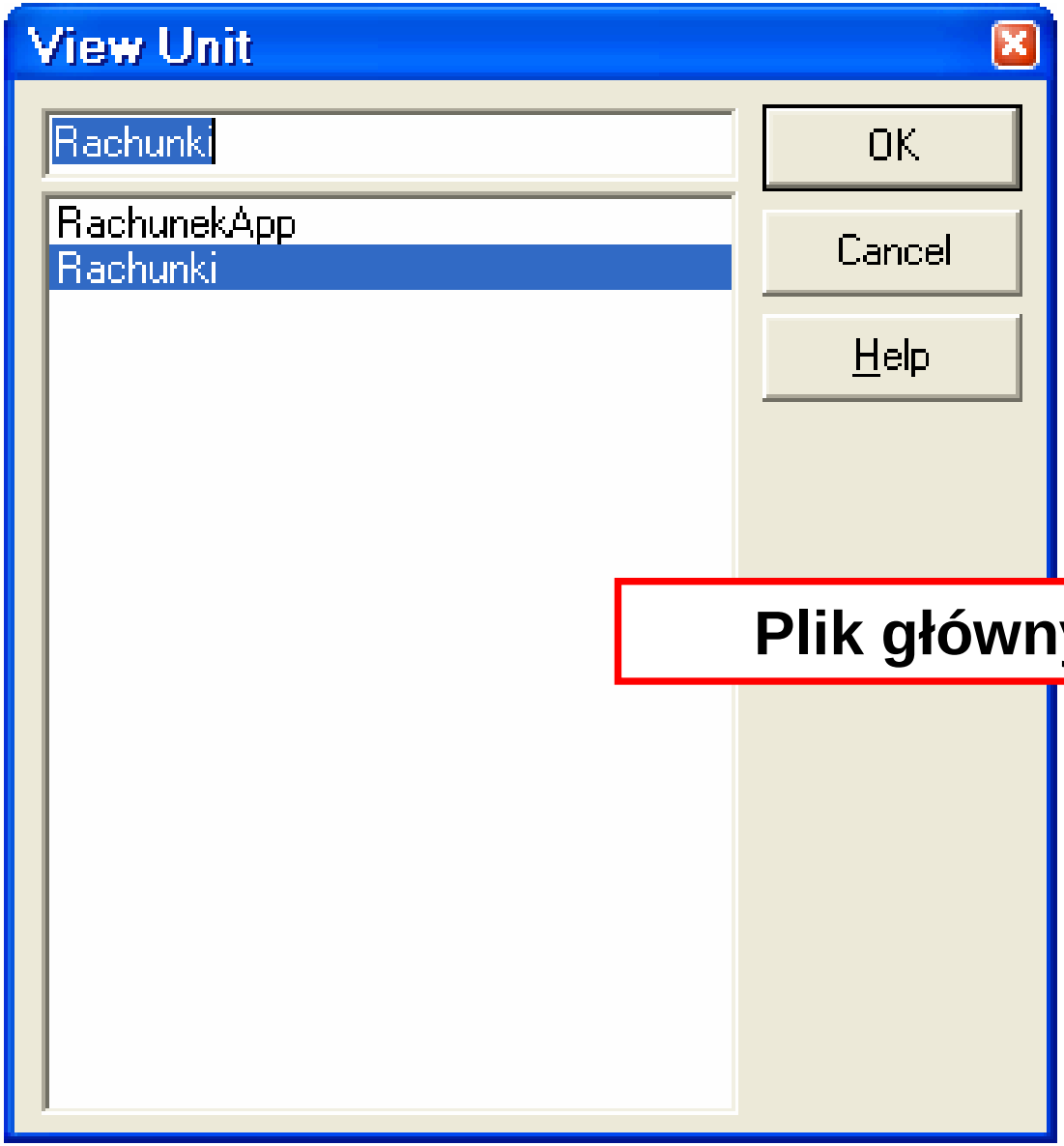

C++Builder 6 - Rachunki
File Edit Search View Project Run Component Tools Window Help
Projekt z formularzem
Object TreeView
Form1
Widok formularzy
View Form
Form1
OK
Cancel
Help
Object Inspector
Properties
Events
Action
ActiveControl
Align
AlphaBlend
Anchors
AutoScroll
AutoSize
BiDiMode
BorderIcons
BorderStyle
BorderWidth
Caption
ClientHeight
ClientWidth
Color
Constraints
TClientSocket [ScktComp]
Add to form
Projekt domyślny
Polski
start
C++Builder 6
Windows Commander...
W_10
15:18

Widok modułów

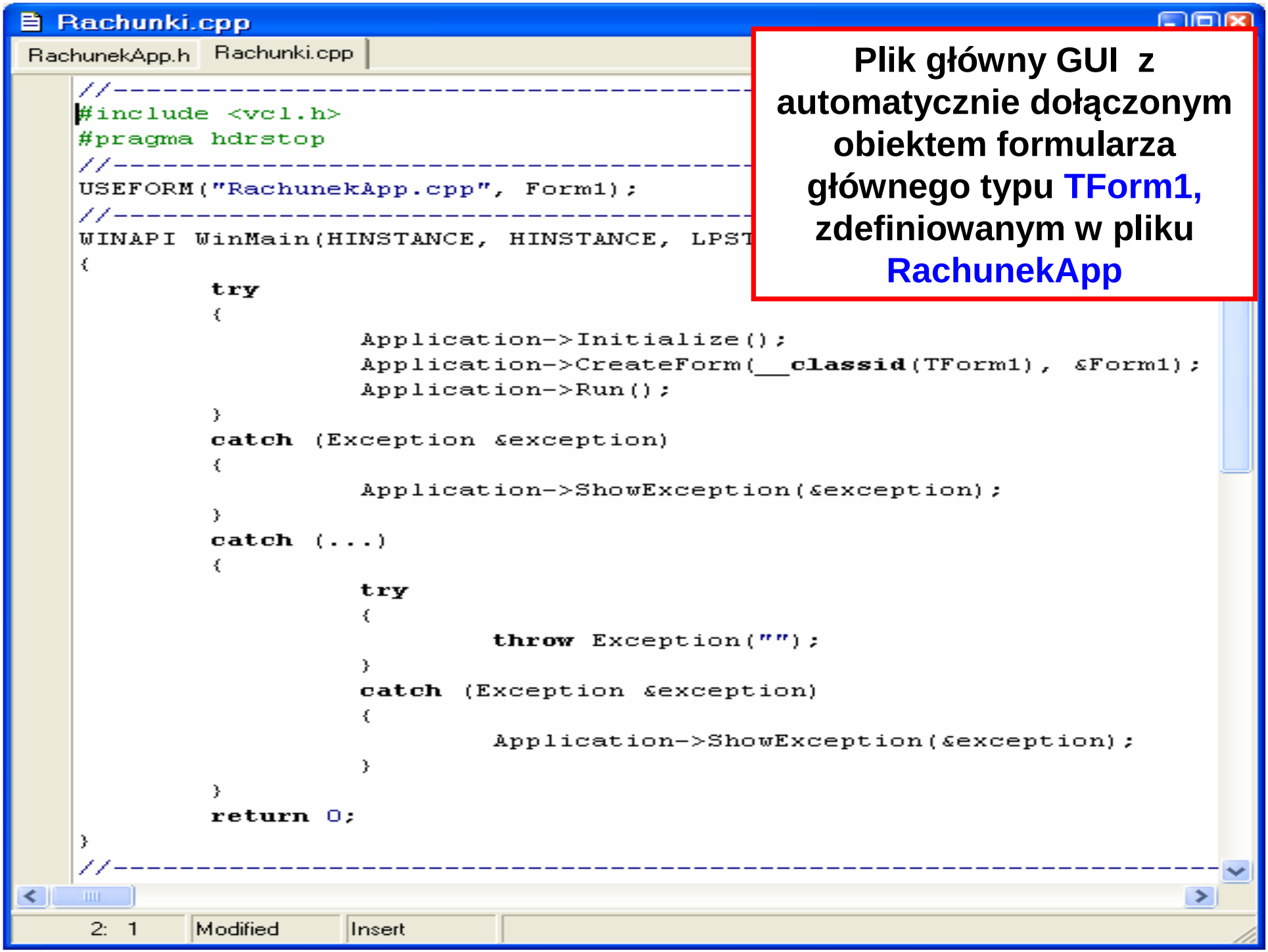
Przed Save Project As

Po Save Project As

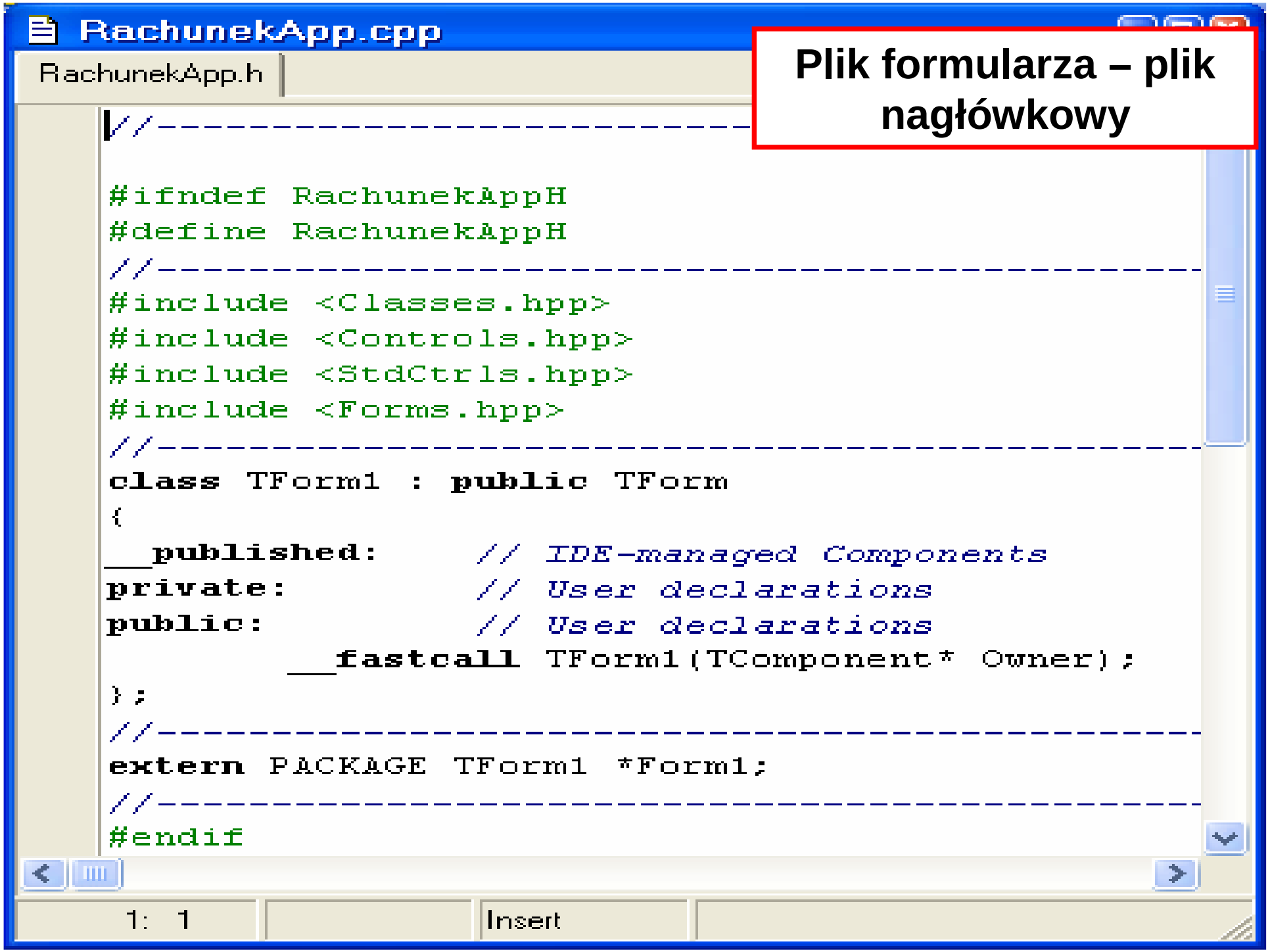


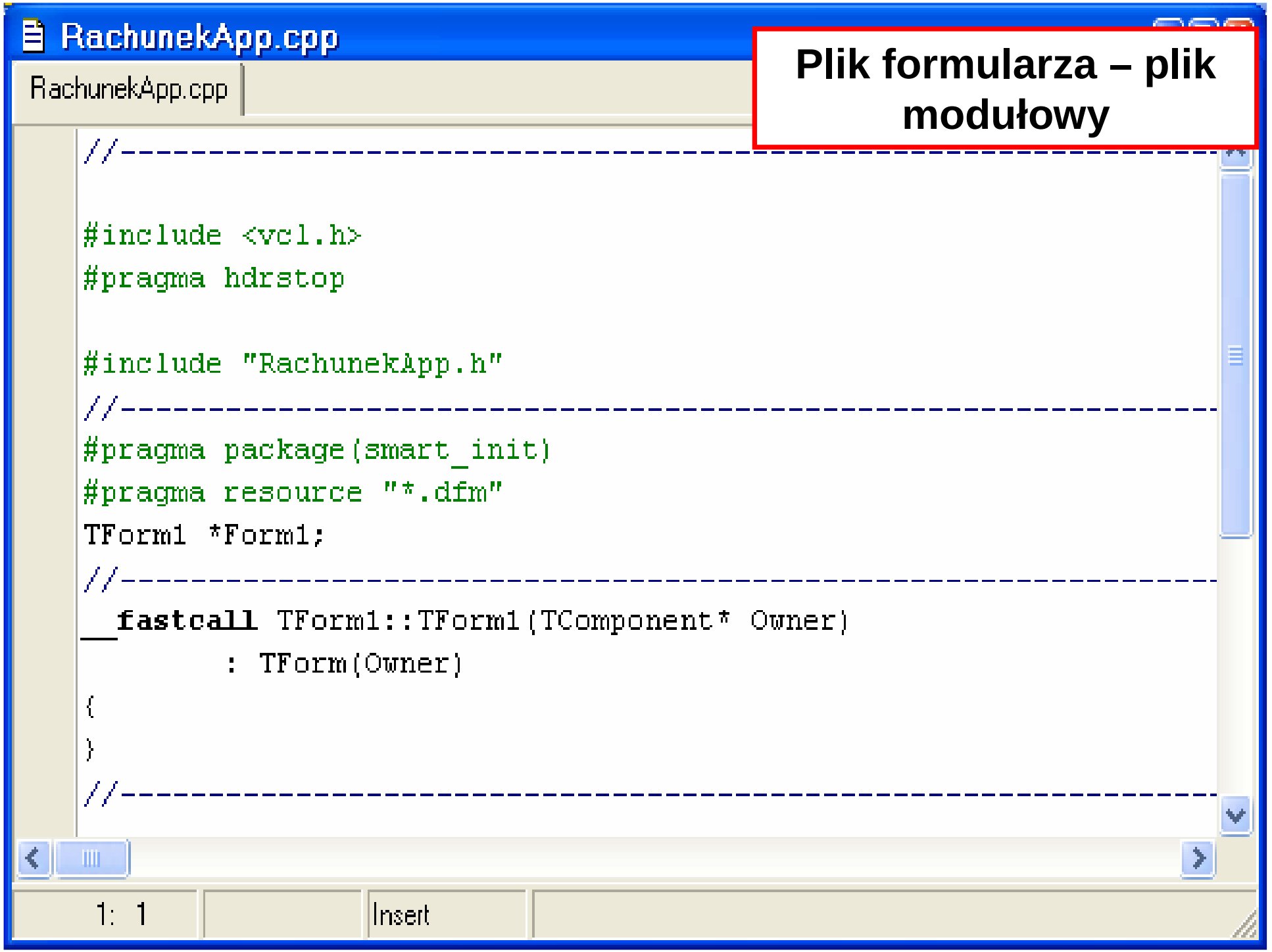


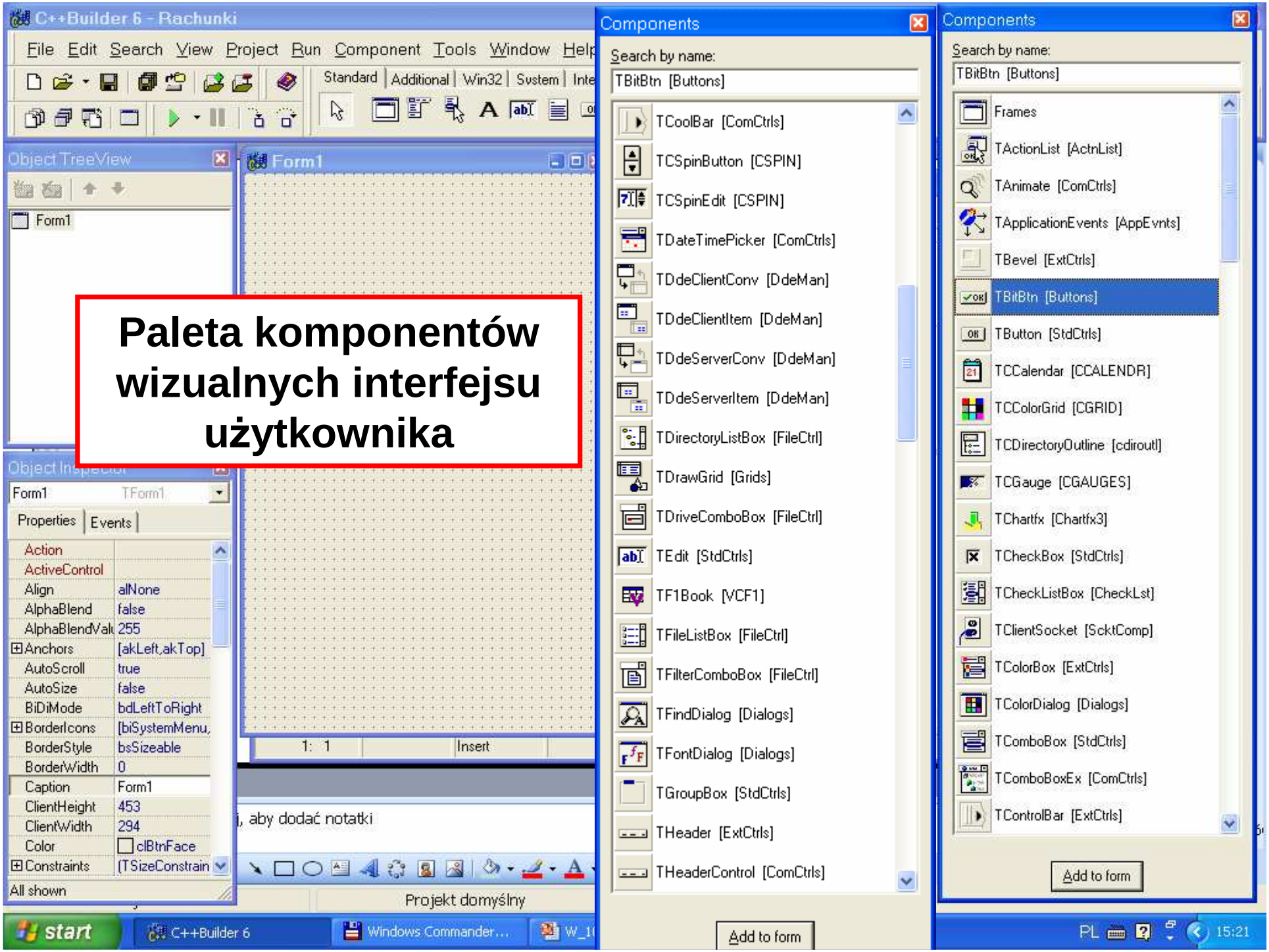
Plik główny GUI



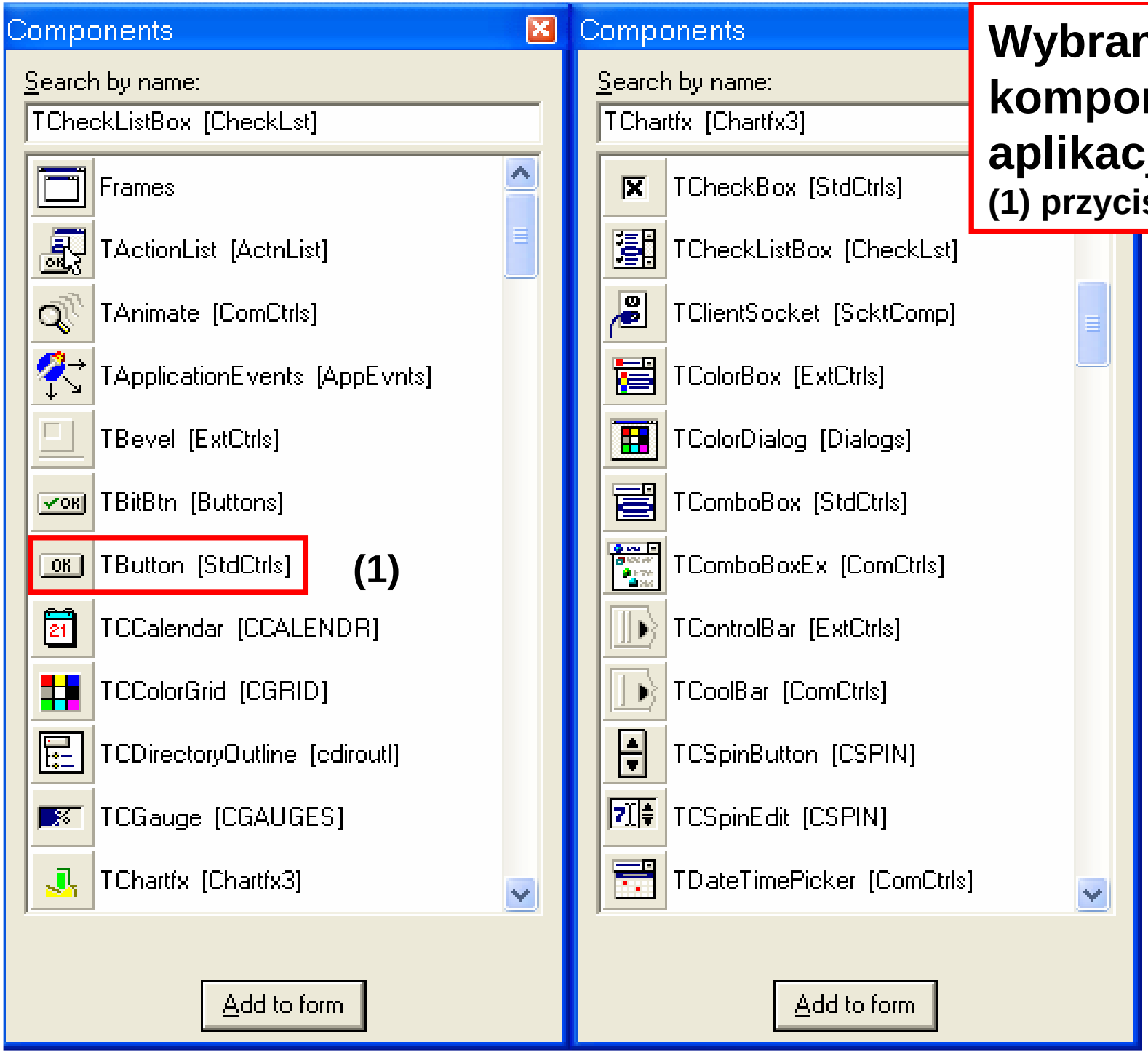
Plik główny GUI z automatycznie dołączonym obiektem formularza głównego typu TForm1, zdefiniowanym w pliku RachunekApp


RachunekApp.cpp
RachunekApp.h
Plik formularza – plik nagłówkowy
//---------------------------
#ifndef RachunekAppH
#define RachunekAppH
//---
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
//---
class TForm1 : public TForm
{
__published: // IDE-managed Components
private: // User declarations
public: // User declarations
__fastcall TForm1(TComponent* Owner);
};
//---
extern PACKAGE TForm1 *Form1;
//---
#endif
1: 1
Insert


RachunekApp.cpp
RachunekApp.cpp
Plik formularza – plik modułowy
//---
#include <vcl.h>
#pragma hdrstop
#include "RachunekApp.h"
//---
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//---
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
}
//---
1: 1
Insert

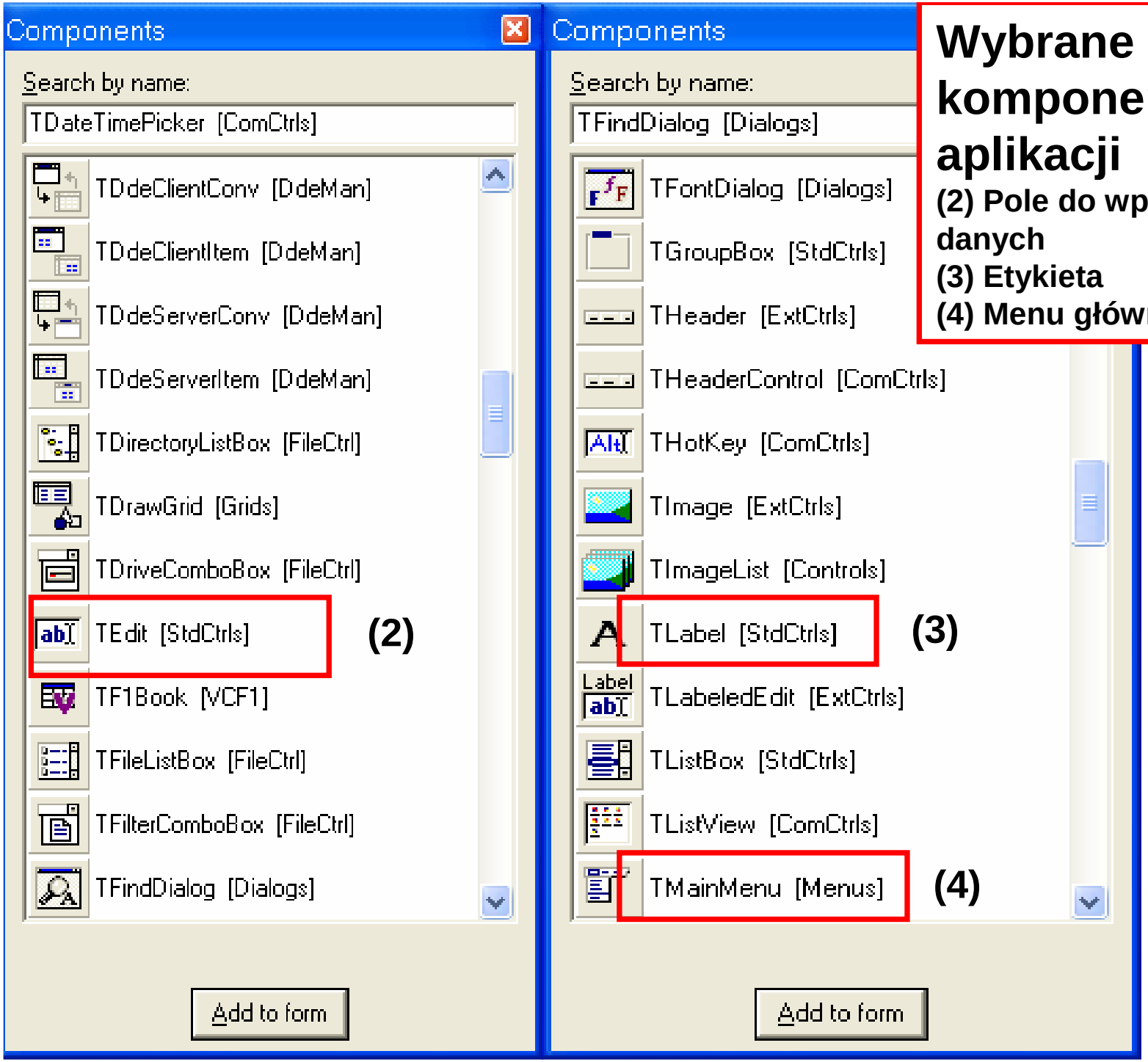


Paleta komponentów wizualnych interfejsu użytkownika



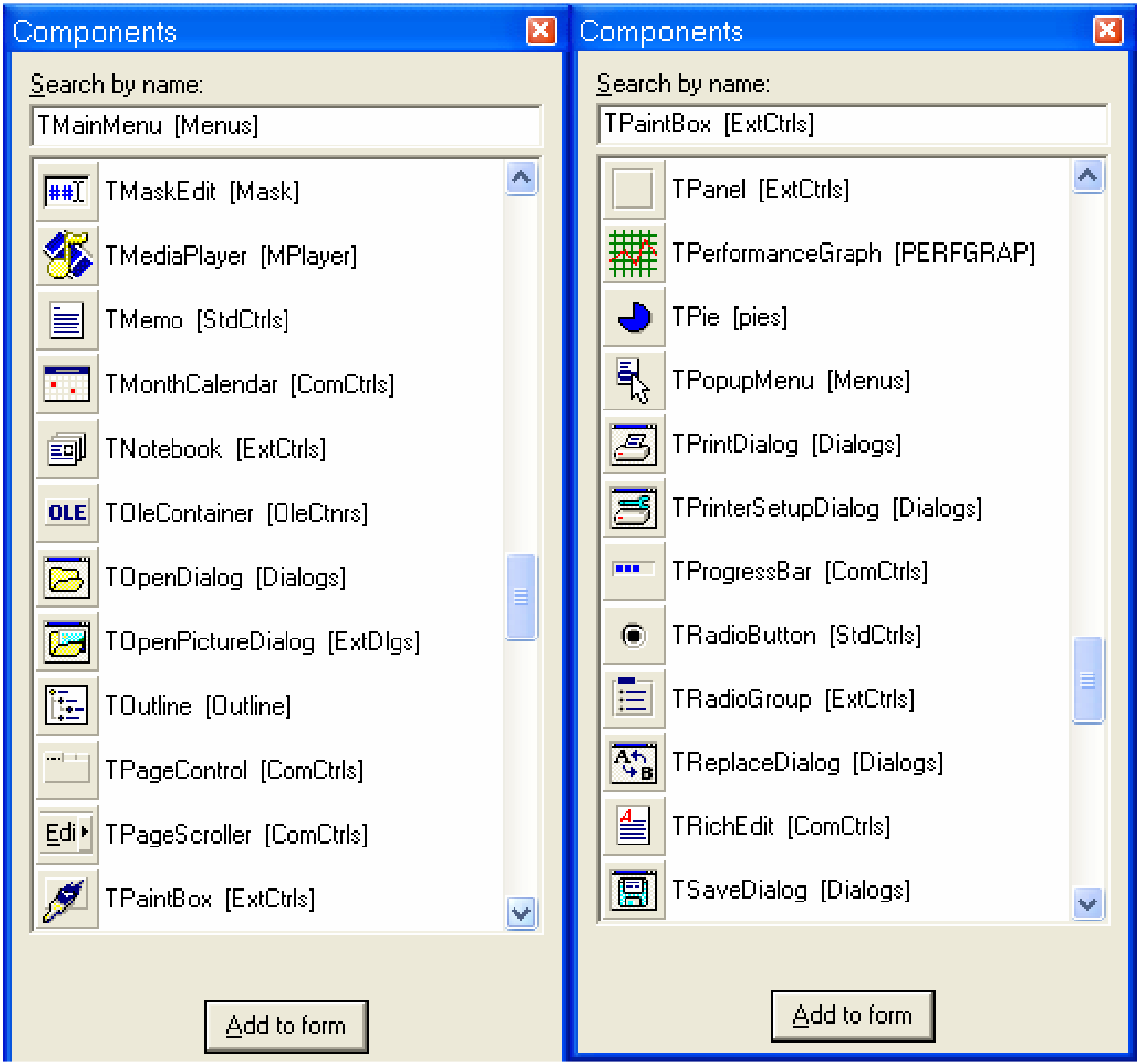
Wybrane komponenty w aplikacji

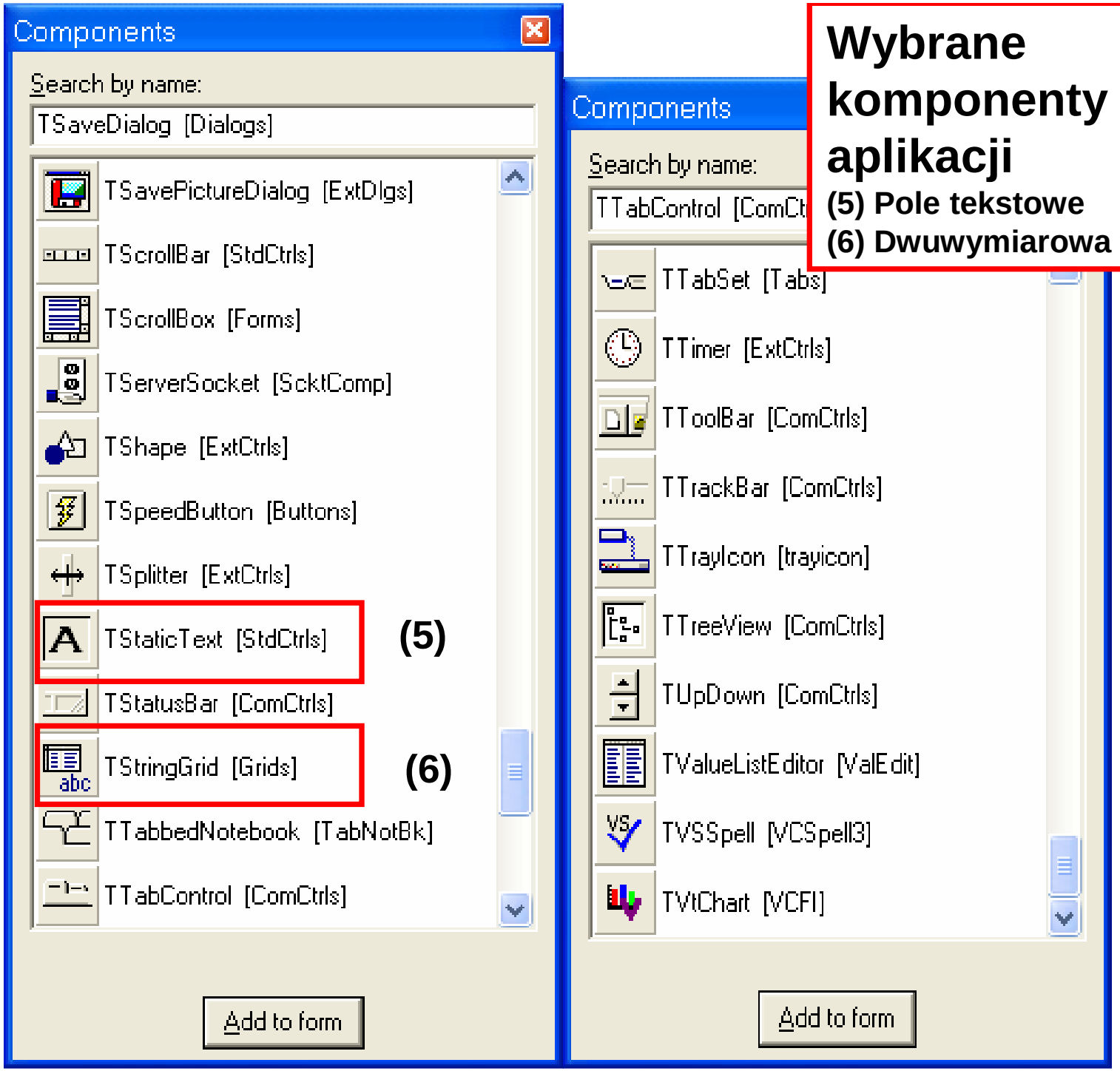
(1) przycisk



Wybrane komponenty w aplikacji

(2) Pole do wprowadzania danych
(3) Etykieta
(4) Menu głównego formularza

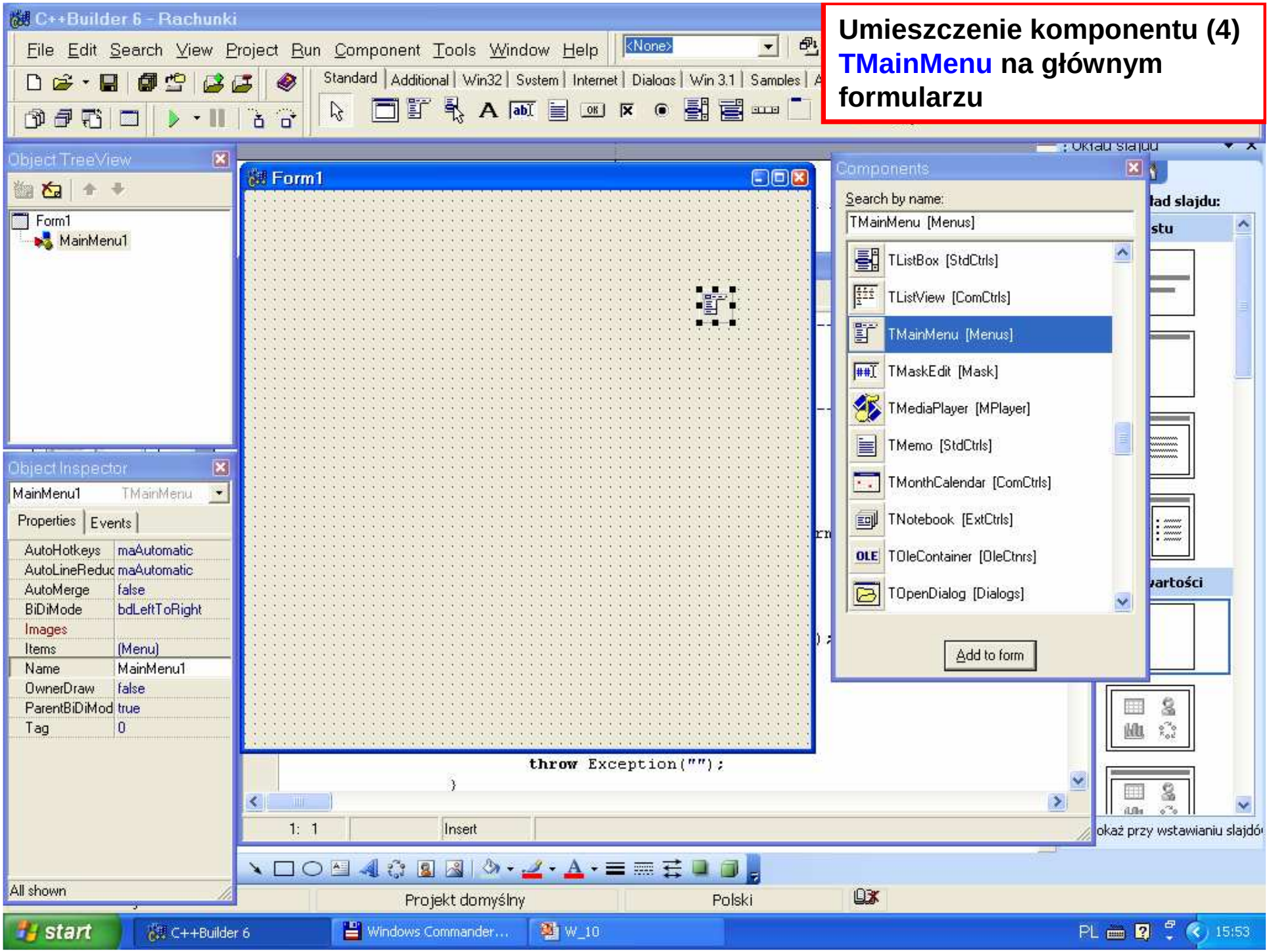

Components
Search by name:
TMainMenu [Menus]
TMaskEdit [Mask]
TMediaPlayer [MPlayer]
TMemo [StdCtrls]
TMonthCalendar [ComCtrls]
TNotebook [ExtCtrls]
TOleContainer [OleCtnrs]
TOpenDialog [Dialogs]
TOpenPictureDialog [ExtDlgs]
TOutline [Outline]
TPageControl [ComCtrls]
TPageScroller [ComCtrls]
TPaintBox [ExtCtrls]
Add to form
Components
Search by name:
TPaintBox [ExtCtrls]
TPanel [ExtCtrls]
TPerformanceGraph [PERFGRAP]
TPie [pies]
TPopupMenu [Menus]
TPrintDialog [Dialogs]
TPrinterSetupDialog [Dialogs]
TProgressBar [ComCtrls]
TRadioButton [StdCtrls]
TRadioGroup [ExtCtrls]
TReplaceDialog [Dialogs]
TRichEdit [ComCtrls]
TSaveDialog [Dialogs]
Add to form



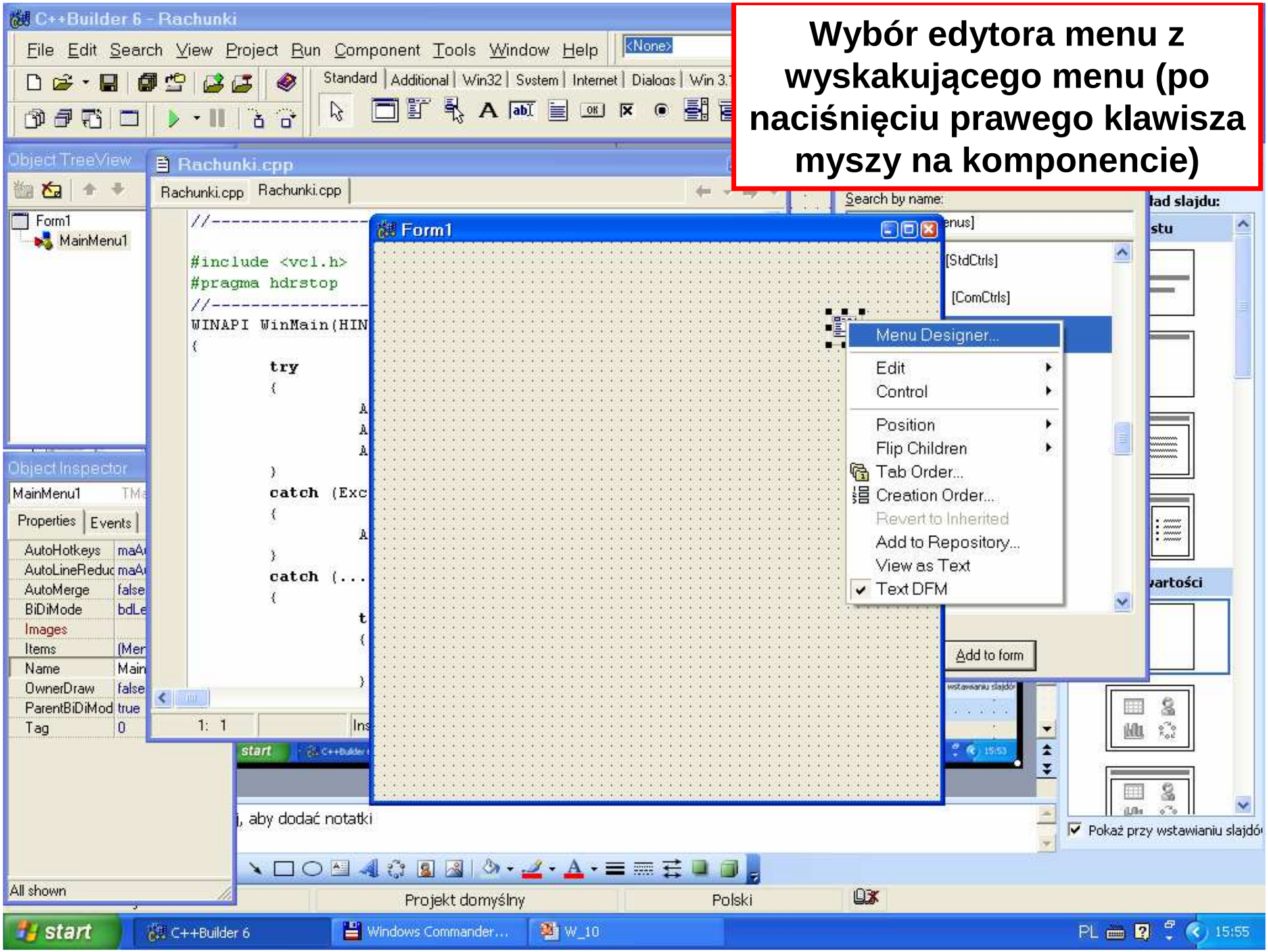
Wybrane komponenty w aplikacji

(5) Pole tekstowe

(6) Dwuwymiarowa tablica

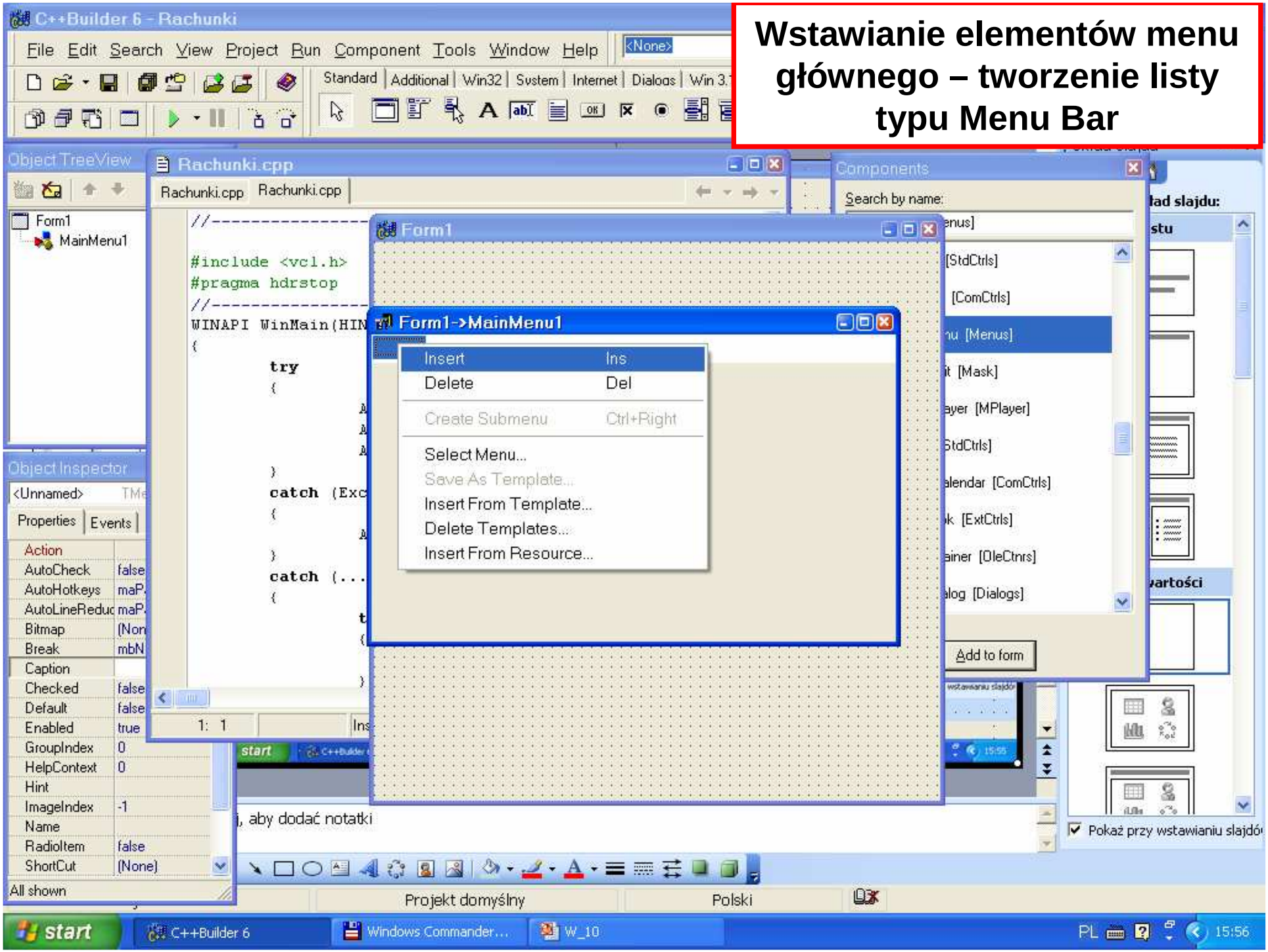


Umieszczenie komponentu (4) TMainMenu na głównym formularzu

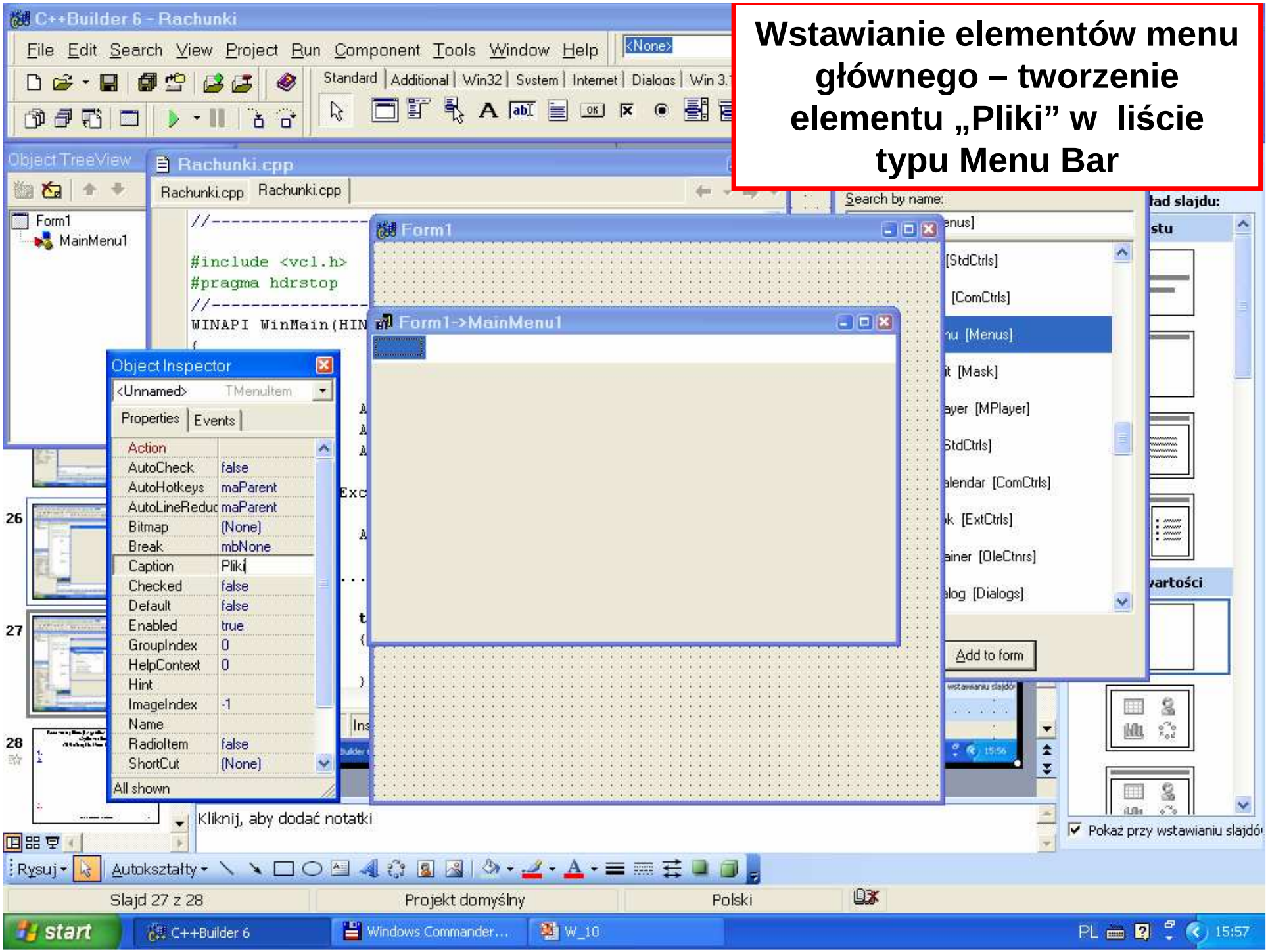


Wybór edytora menu z wyskakującego menu (po naciśnięciu prawego klawisza myszy na komponencie)

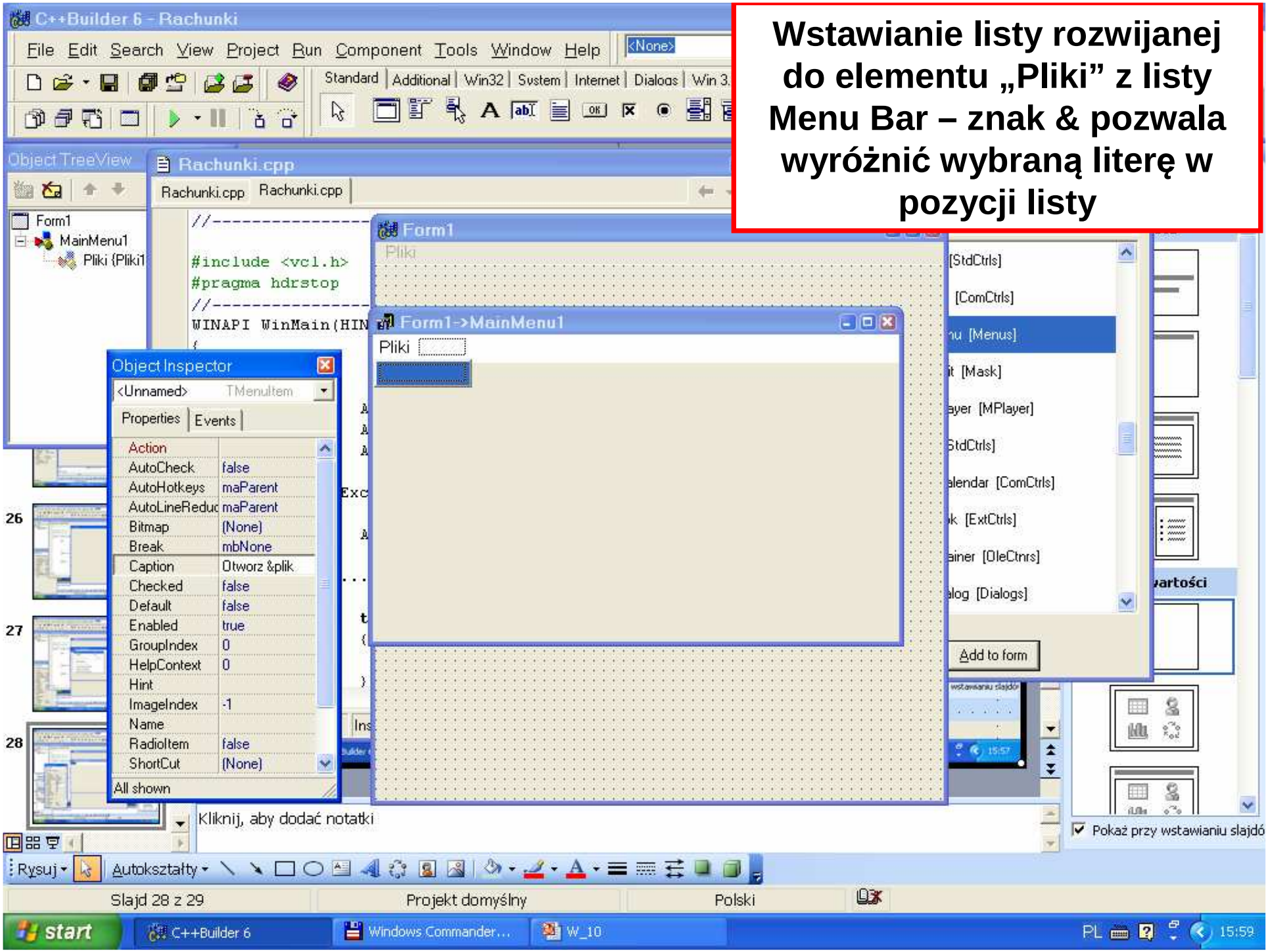
Wstawianie elementów menu głównego – tworzenie listy typu Menu Bar

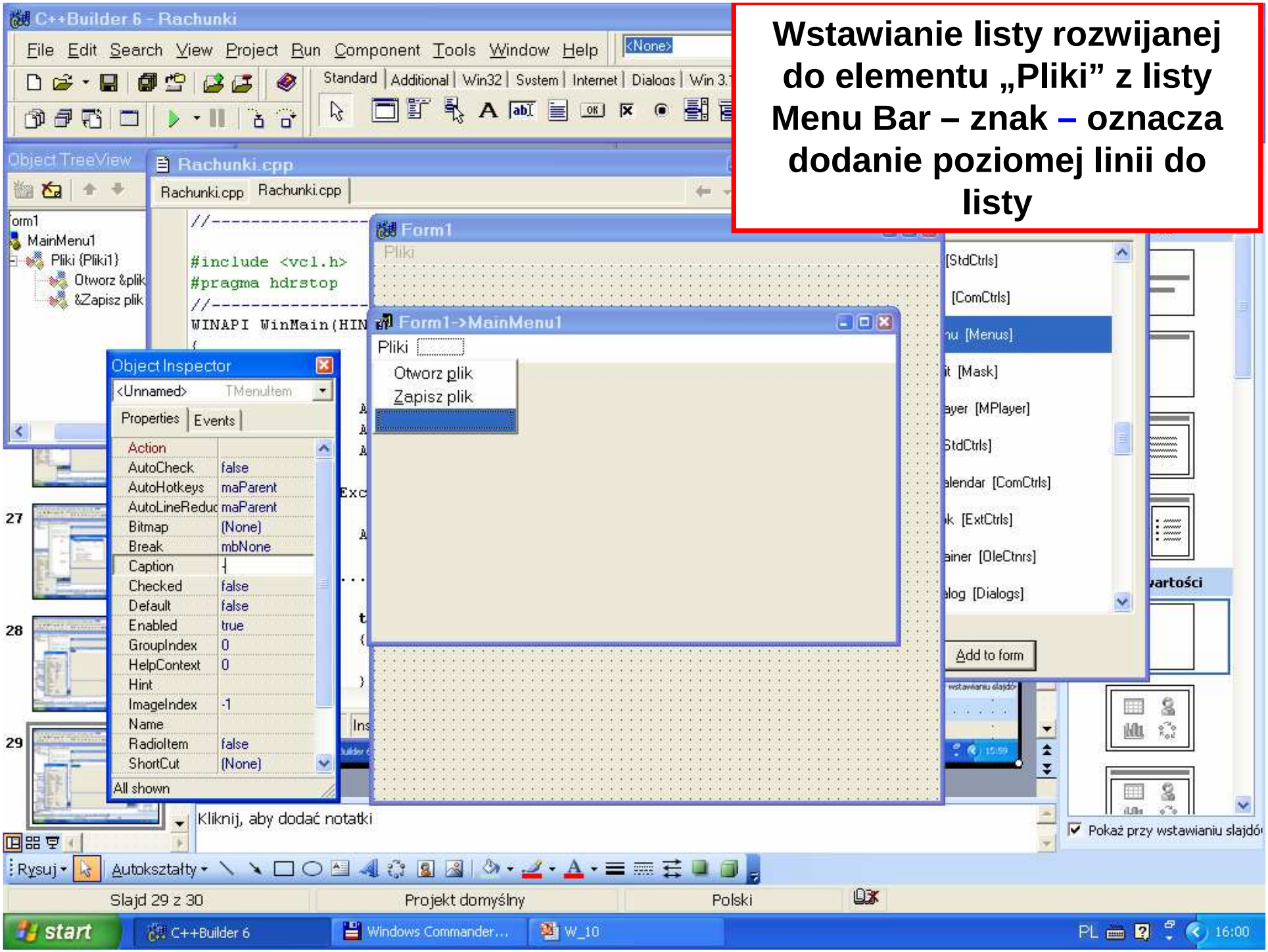


Wstawianie elementów menu głównego – tworzenie elementu „Pliki” w liście typu Menu Bar



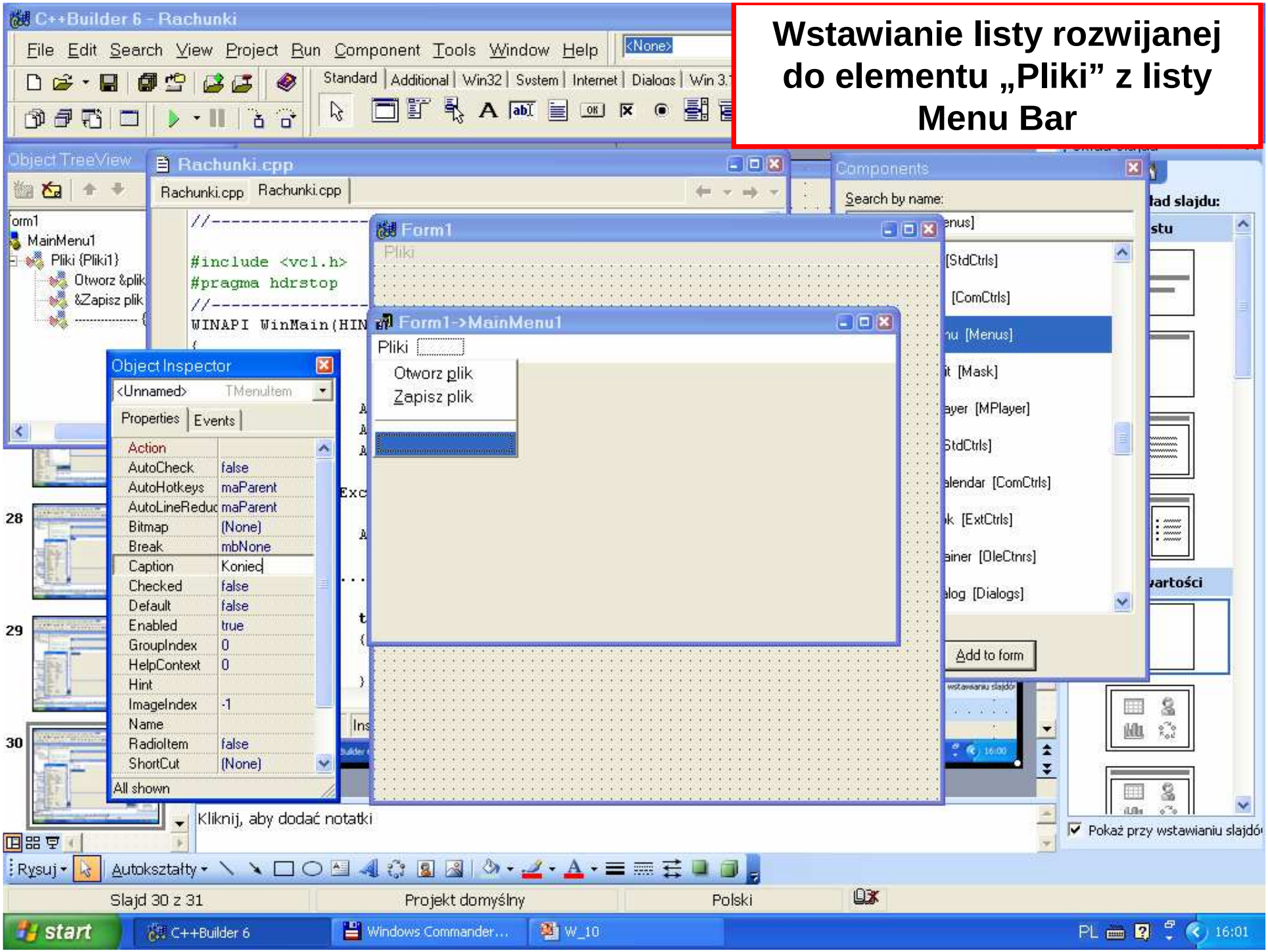
Wstawianie listy rozwijanej do elementu „Pliki” z listy Menu Bar – znak & pozwala wyróżnić wybraną literę w pozycji listy

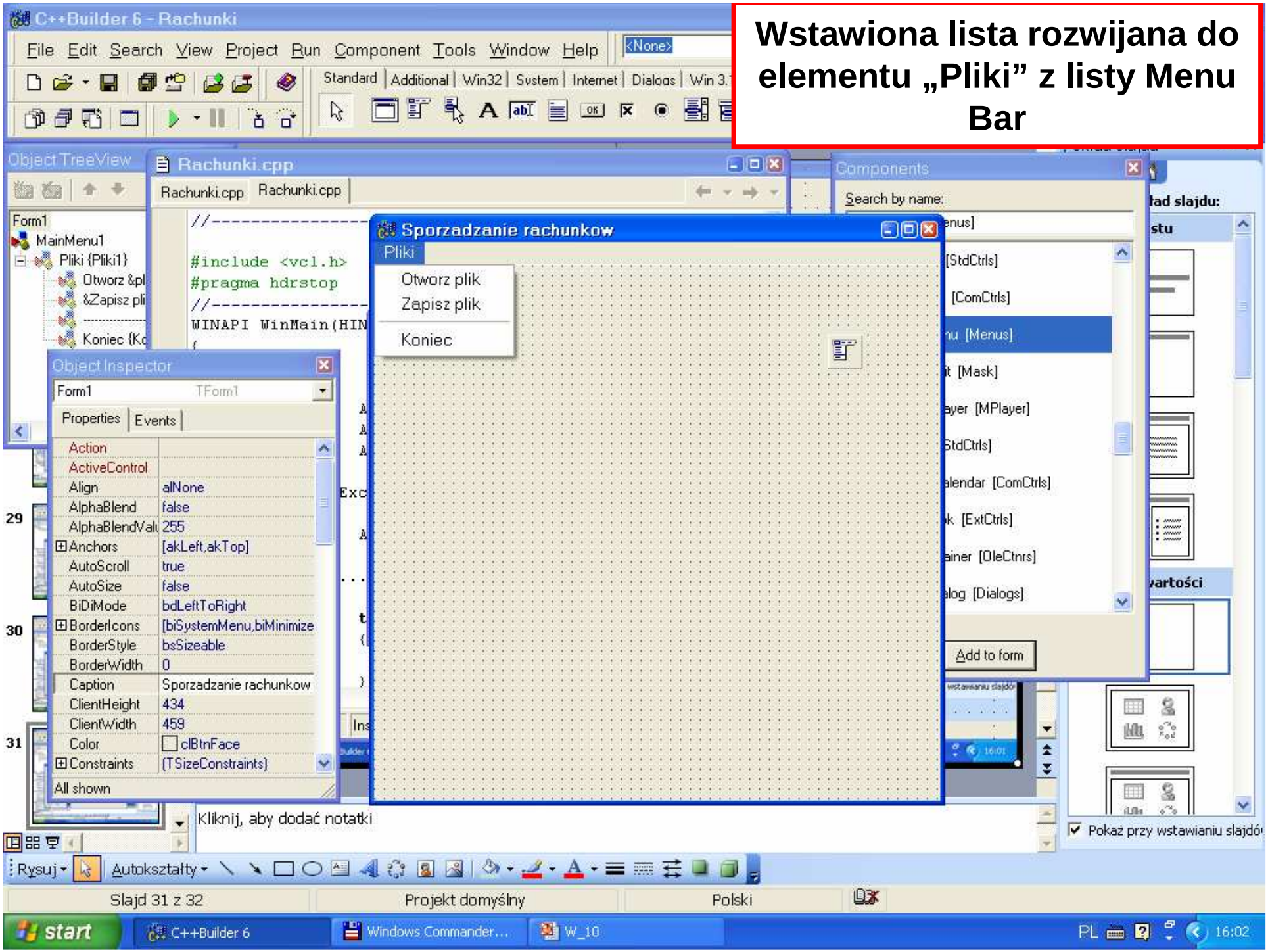




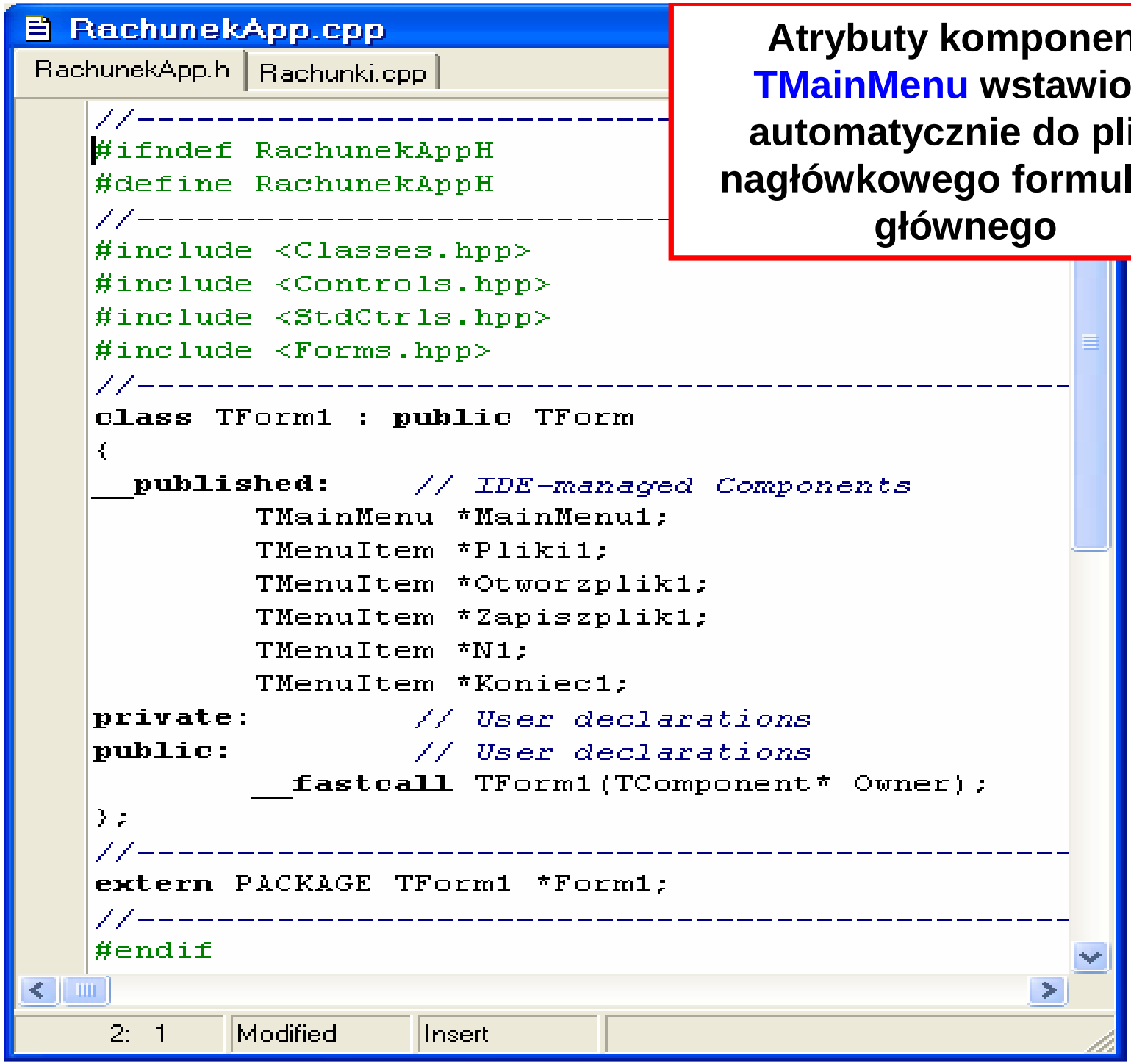
Wstawianie listy rozwijanej do elementu „Pliki” z listy Menu Bar – znak – oznacza dodanie poziomej linii do listy

Wstawianie listy rozwijanej do elementu „Pliki” z listy Menu Bar

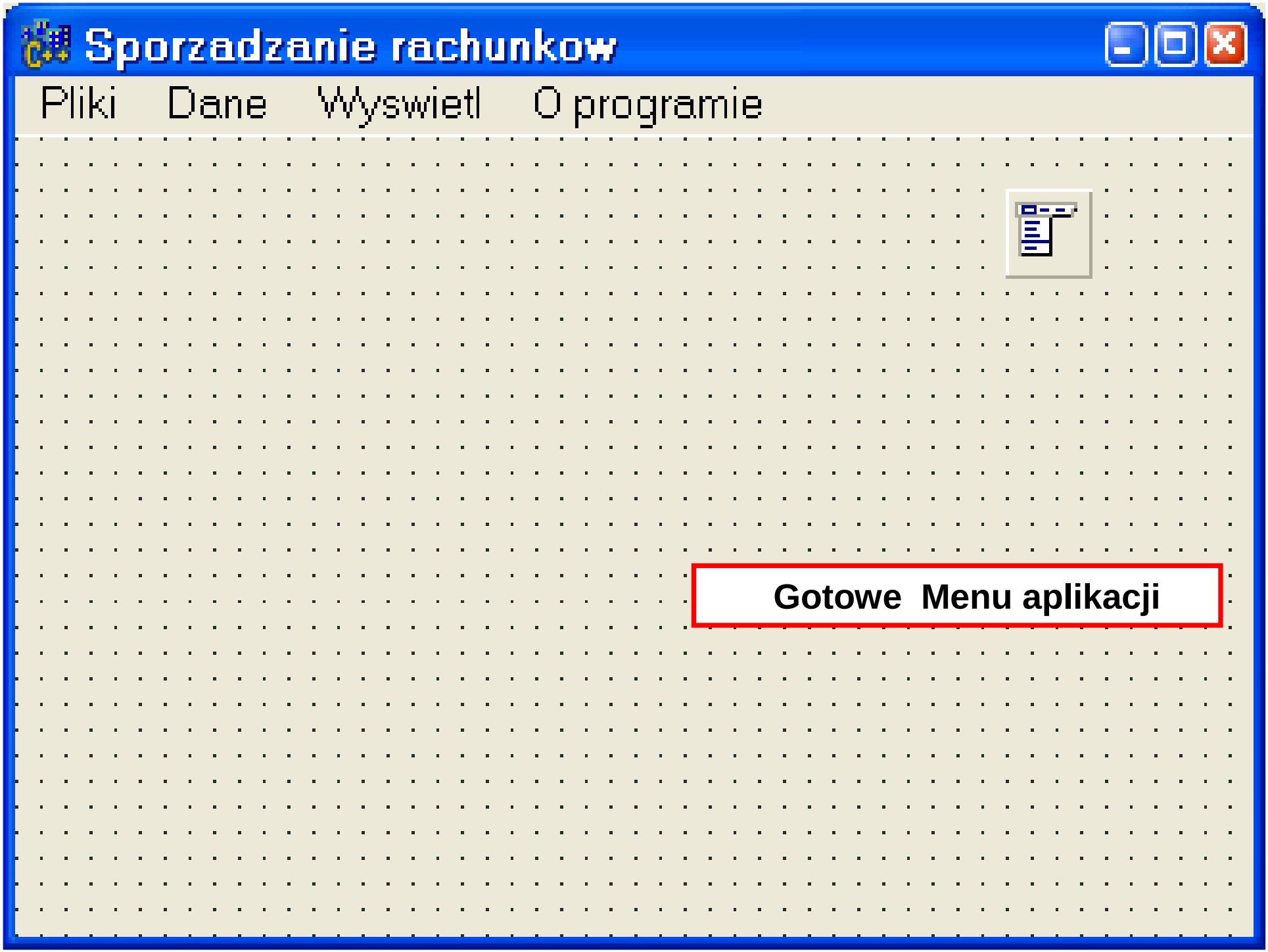




Wstawiona lista rozwijana do elementu „Pliki” z listy Menu Bar

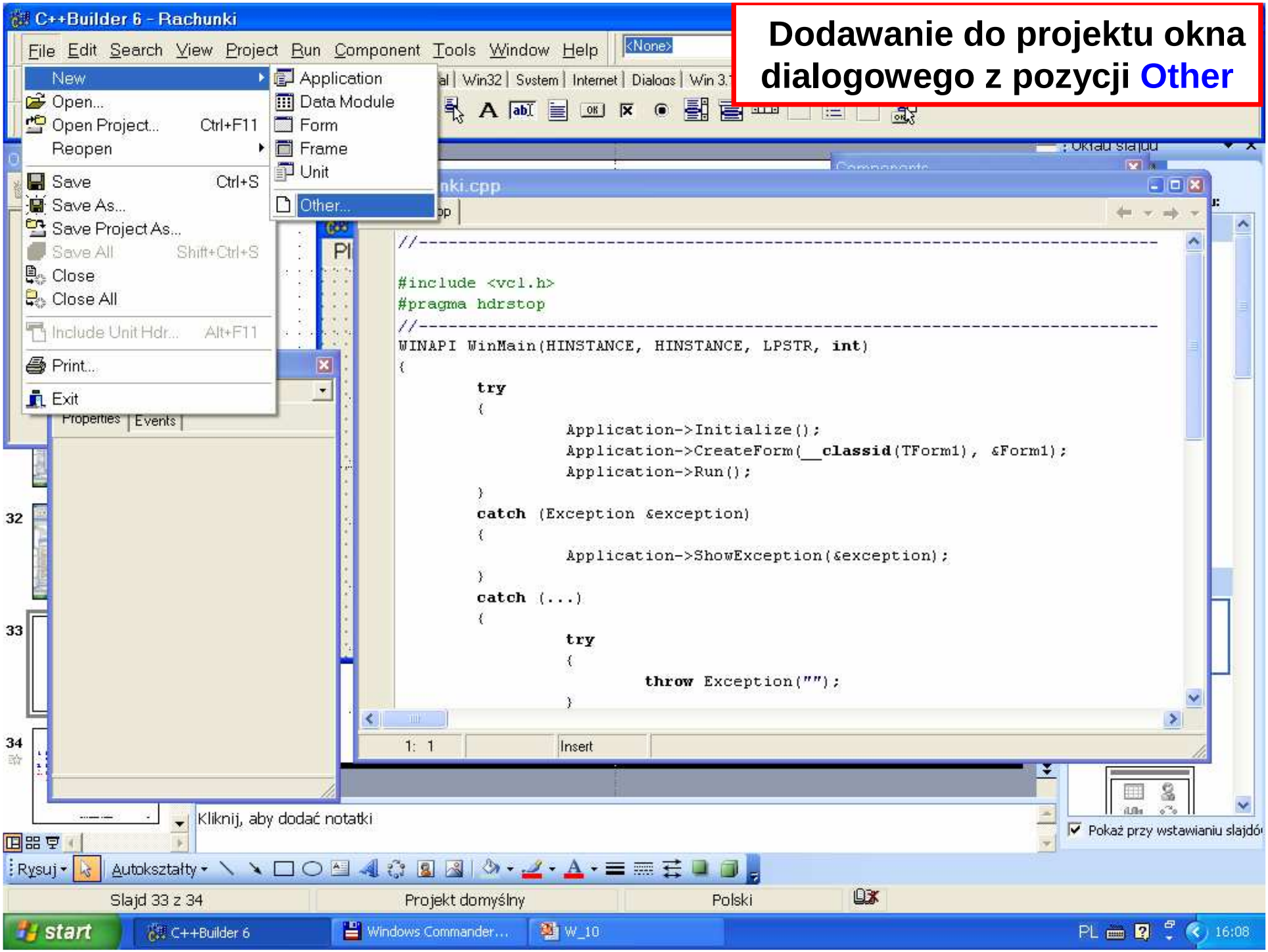


Atrybuty komponentu TMainMenu wstawione automatycznie do pliku nagłówkowego formularza głównego

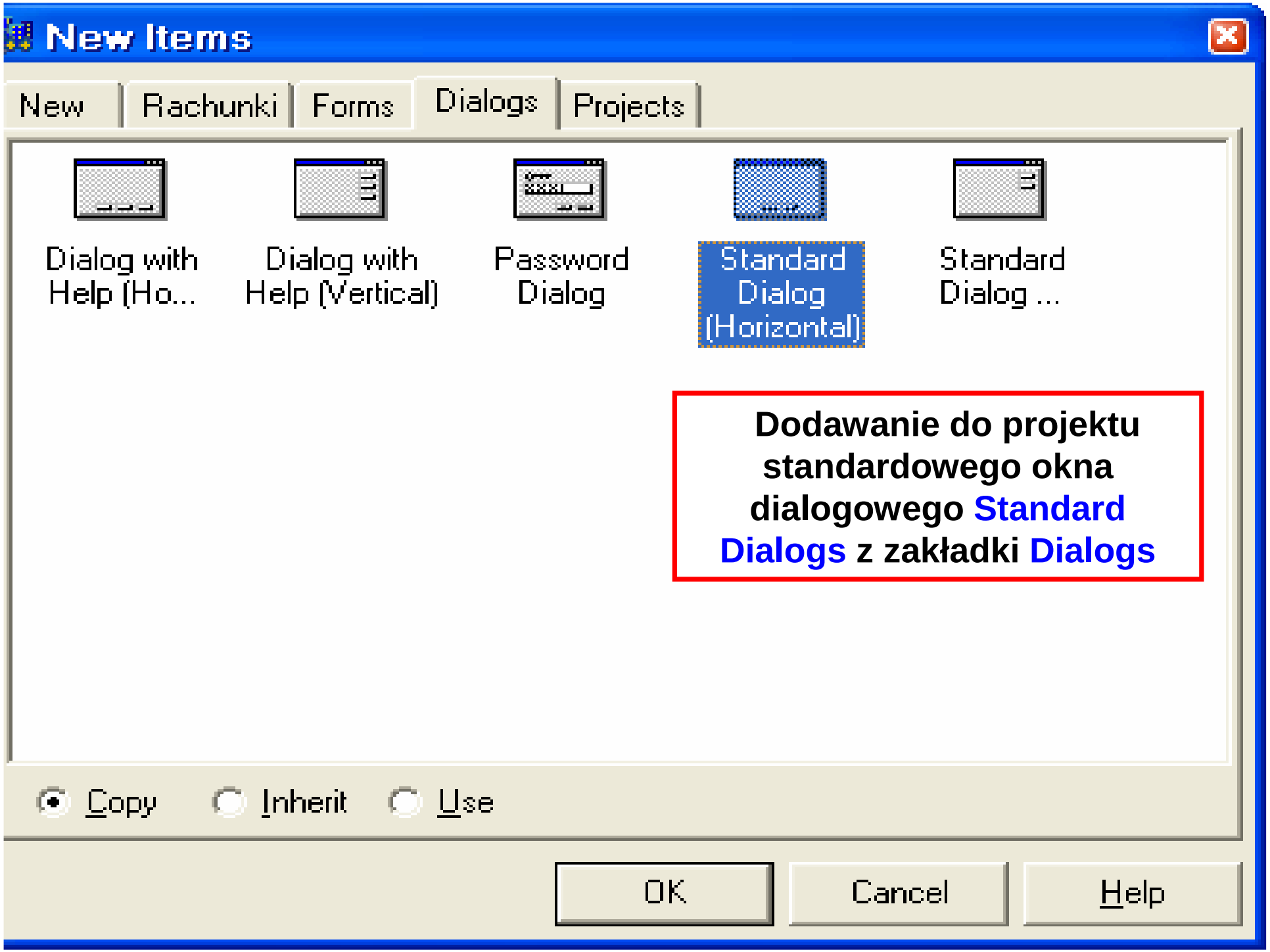

Sporzadzanie rachunkow
Pliki
Dane
Wyswietl
O programie
Gotowe Menu aplikacji

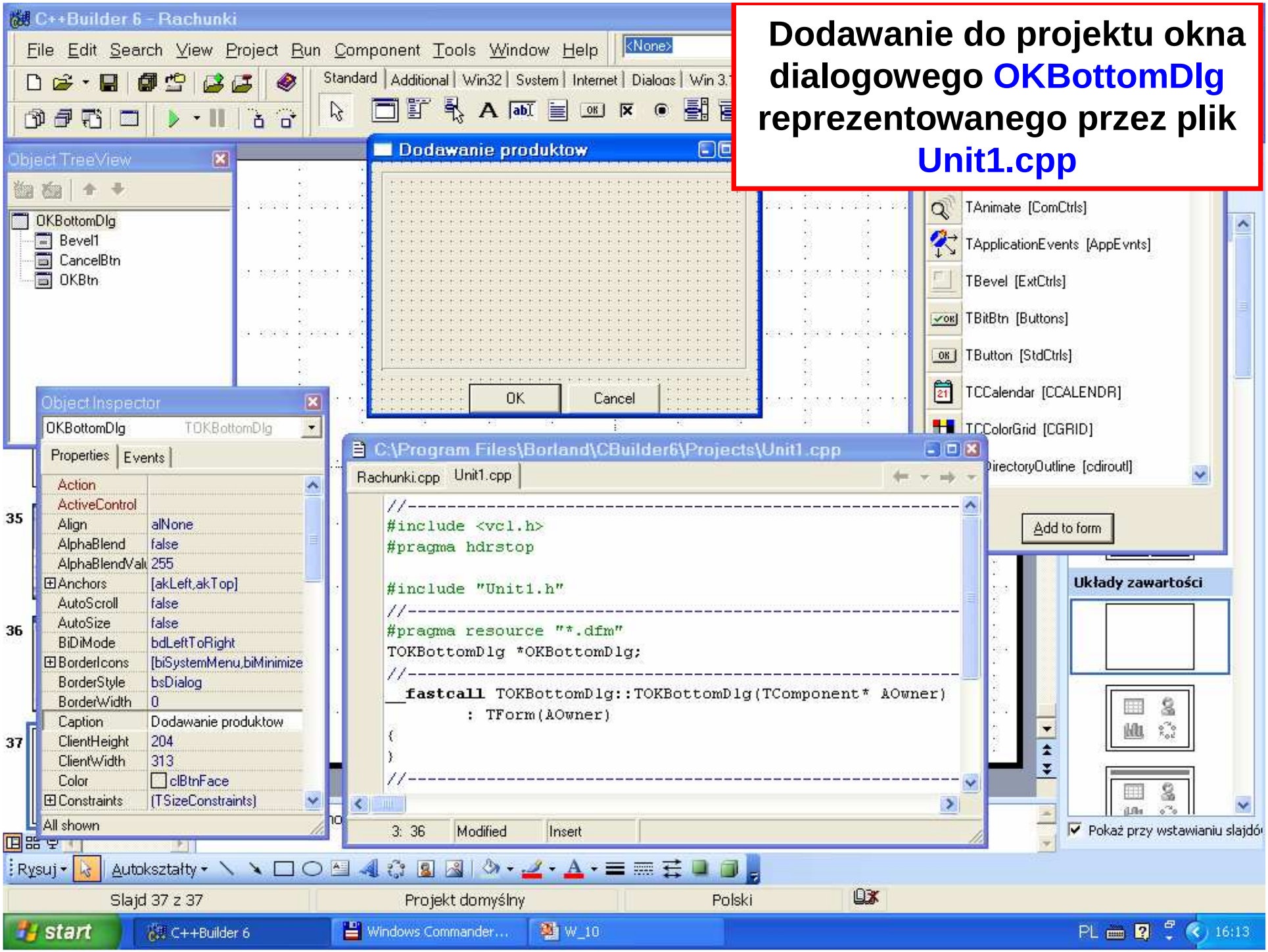
Budowa aplikacji z graficznym interfejsem użytkownika - GUI (Graphic User Interface)

1. Udostępnianie wszystkich prywatnych atrybutów do prezentacji, wprowadzenie standardu nazewnictwa plików – nazwy plików aplikacji poprzedzone literą T
2. Budowa głównego formularza GUI
3. Budowa okienek dialogowych do wprowadzania danych



Dodawanie do projektu okna dialogowego z pozycji Other

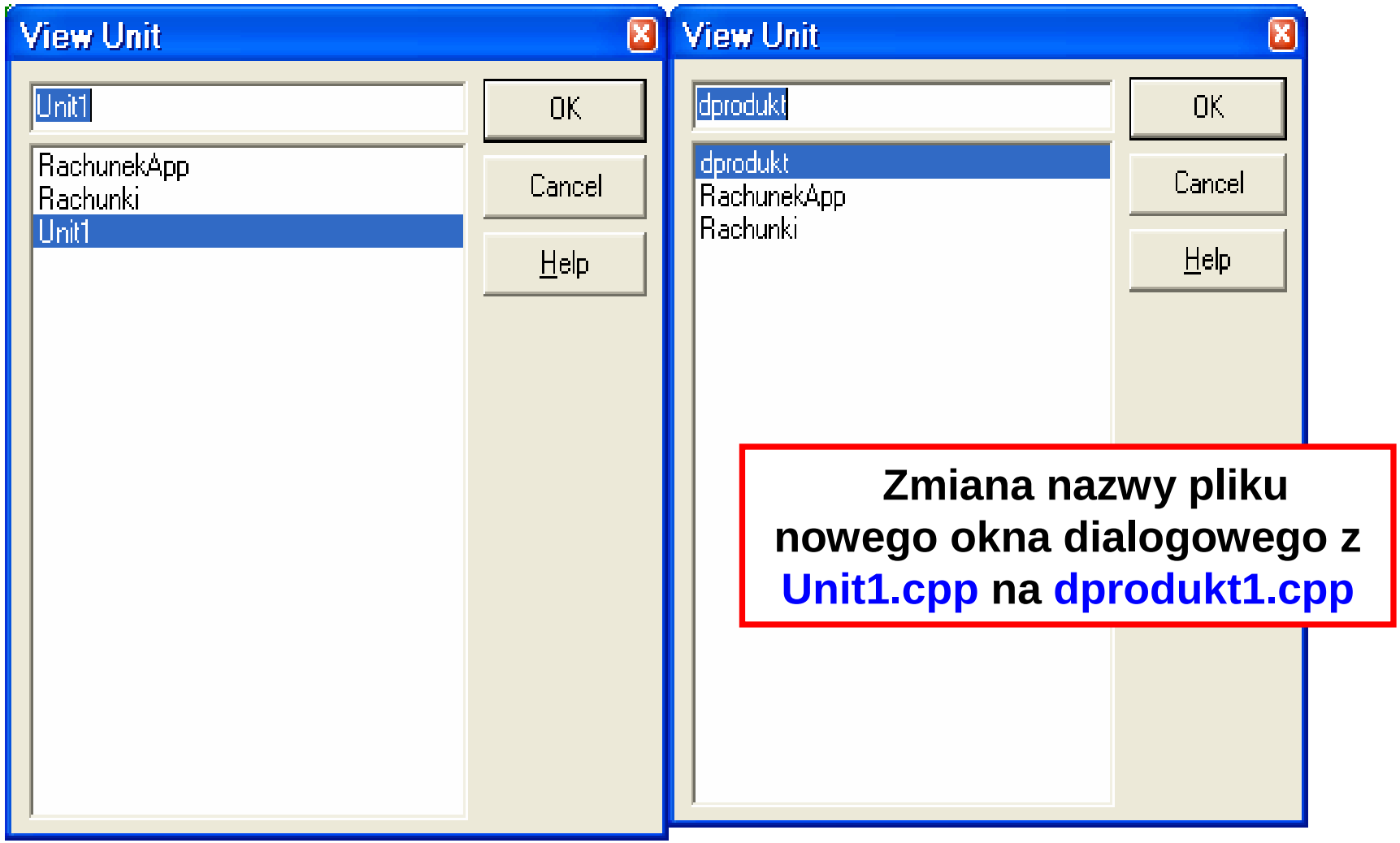

New Items
New
Rachunki
Forms
Dialogs
Projects
Dialog with Help (Ho...
Dialog with Help (Vertical)
Password Dialog
Standard Dialog (Horizontal)
Standard Dialog ...
Dodawanie do projektu standardowego okna dialogowego Standard Dialogs z zakładki Dialogs
Copy
Inherit
Use
OK
Cancel
Help

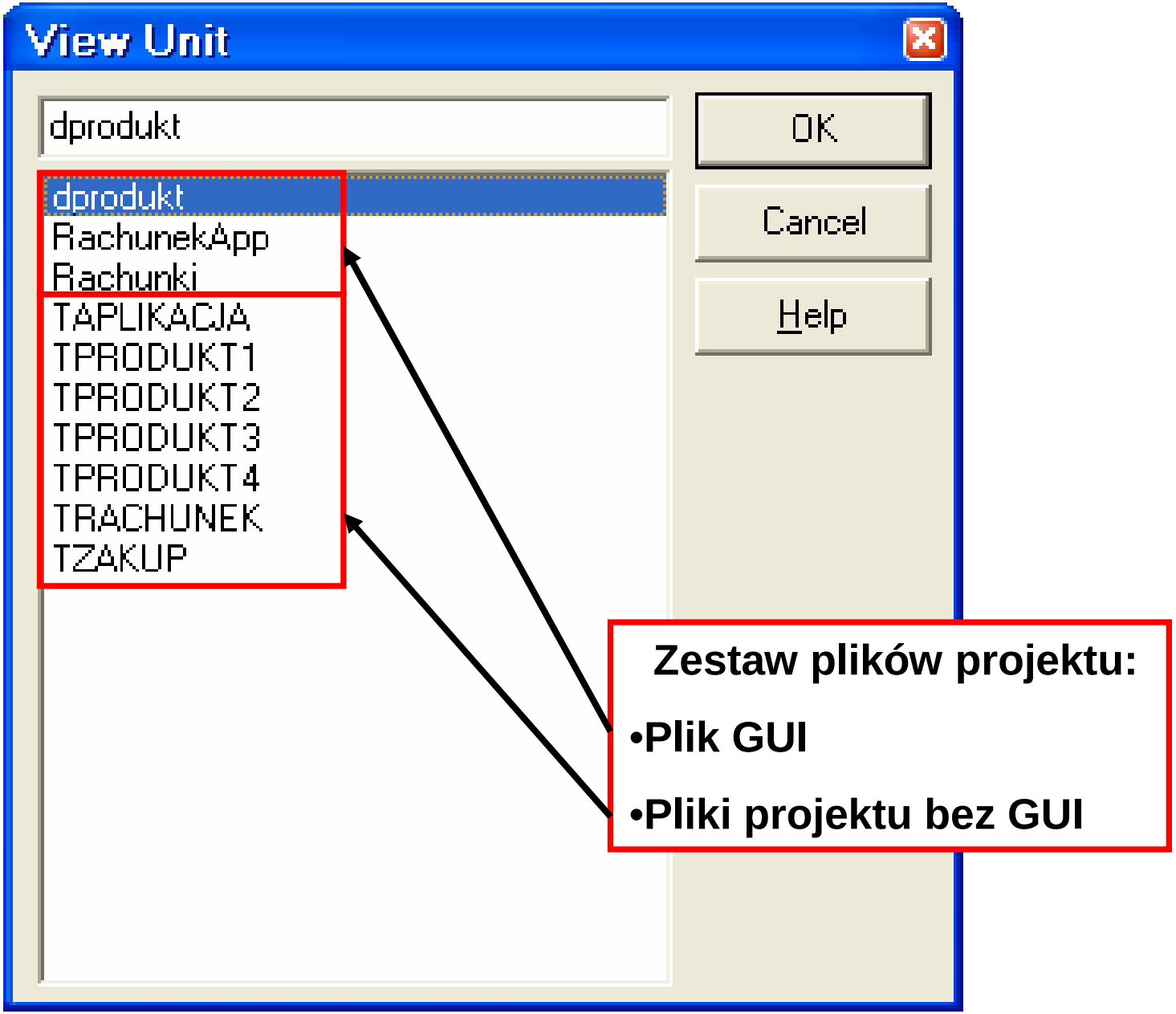

Dodawanie do projektu okna dialogowego OKBottomDlg reprezentowanego przez plik Unit1.cpp
C++Builder 6 - Rachunki
File Edit Search View Project Run Component Tools Window Help
Object TreeView
OKBottomDlg
Bevel1
CancelBtn
OKBtn
Dodawanie produktow
OK
Cancel
Object Inspector
OKBottomDlg TOKBottomDlg
Properties Events
Action
ActiveControl
Align alNone
AlphaBlend false
AlphaBlendVal 255
Anchors [akLeft,akTop]
AutoScroll false
AutoSize false
BiDiMode bdLeftToRight
BorderIcons [biSystemMenu,biMinimize
BorderStyle bsDialog
BorderWidth 0
Caption Dodawanie produktow
ClientHeight 204
ClientWidth 313
Color clBtnFace
Constraints (TSizeConstraints)
All shown
C:\Program Files\Borland\CBuilder6\Projects\Unit1.cpp
Rachunki.cpp Unit1.cpp
//---
#include <vcl.h>
#pragma hdrstop
#include "Unit1.h"
//---
#pragma resource "*.dfm"
TOKBottomDlg *OKBottomDlg;
//---
__fastcall TOKBottomDlg::TOKBottomDlg(TComponent* AOwner)
: TForm(AOwner)
{
}
//---
3: 36 Modified Insert
TAnimate [ComCtrls]
TApplicationEvents [AppEvnts]
TBevel [ExtCtrls]
TBitBtn [Buttons]
TButton [StdCtrls]
TCCalendar [CCALENDR]
TCColorGrid [CGRID]
DirectoryOutline [cdiroutl]
Add to form
Układy zawartości
Pokaż przy wstawianiu slajdó
Rysuj Autokształty
Slajd 37 z 37
Projekt domyślny
Polski
start
C++Builder 6
Windows Commander...
W_10
PL 16:13

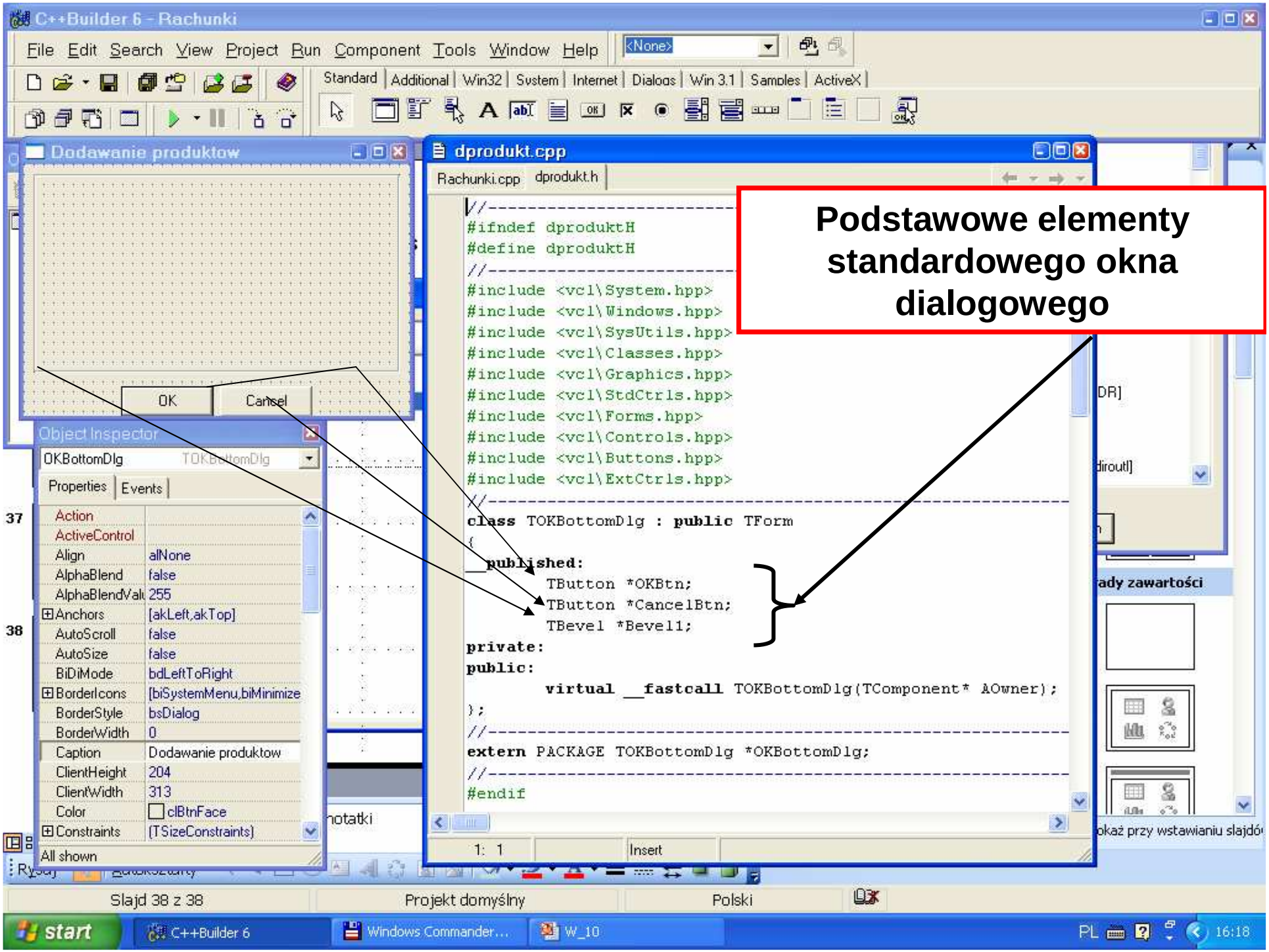
Widok modułów

Przed Save As

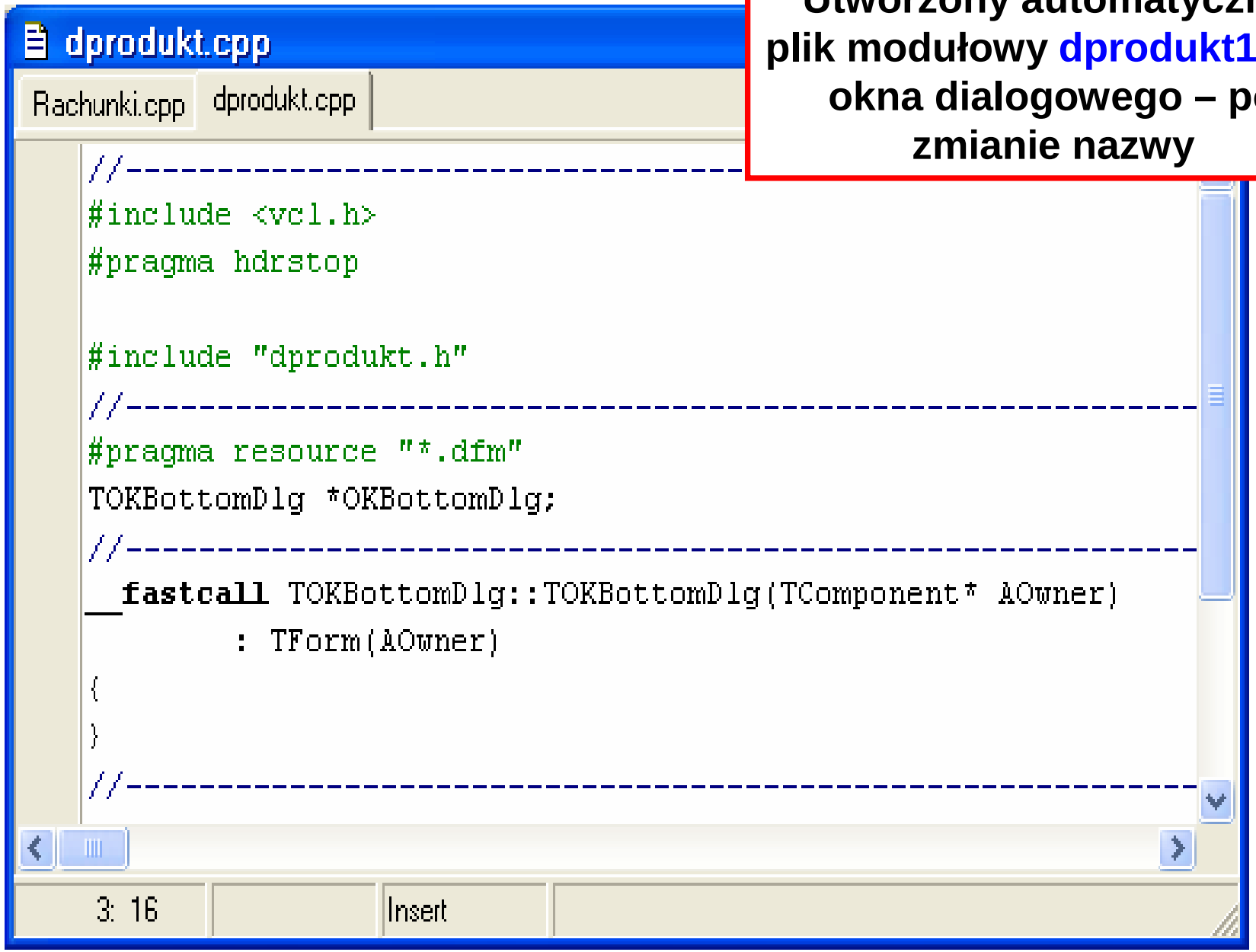
Po Save As




View Unit
dprodukt
OK
Cancel
Help
dprodukt
RachunekApp
Rachunki
TAPLIKACJA
TPRODUKT1
TPRODUKT2
TPRODUKT3
TPRODUKT4
TRACHUNEK
TZAKUP
Zestaw plików projektu:
•Plik GUI
•Pliki projektu bez GUI


C++Builder 6 - Rachunki
File Edit Search View Project Run Component Tools Window Help
Standard Additional Win32 System Internet Dialogs Win 3.1 Samples ActiveX
Dodawanie produktow
OK
Cancel
Object Inspector
OKBottomDlg
Properties Events
Action
ActiveControl
Align alNone
AlphaBlend false
AlphaBlendVal 255
Anchors [akLeft,akTop]
AutoScroll false
AutoSize false
BiDiMode bdLeftToRight
BorderIcons [biSystemMenu,biMinimize
BorderStyle bsDialog
BorderWidth 0
Caption Dodawanie produktow
ClientHeight 204
ClientWidth 313
Color clBtnFace
Constraints (TSizeConstraints)
All shown
dprodukt.cpp
Rachunki.cpp dprodukt.h
//---------------------------
#ifndef dproduktH
#define dproduktH
//---------------------------
#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>
//---------------------------
class TOKBottomDlg : public TForm
{
__published:
TButton *OKBtn;
TButton *CancelBtn;
TBevel *Bevel1;
private:
public:
virtual __fastcall TOKBottomDlg(TComponent* AOwner);
};
//---------------------------
extern PACKAGE TOKBottomDlg *OKBottomDlg;
//---------------------------
#endif
1: 1
Insert
Podstawowe elementy standardowego okna dialogowego
Slajd 38 z 38
Projekt domyślny
Polski
start
C++Builder 6
Windows Commander...
W_10
PL
16:18

Utworzony automatycznie plik modułowy dprodukt1.cpp okna dialogowego – po zmianie nazwy



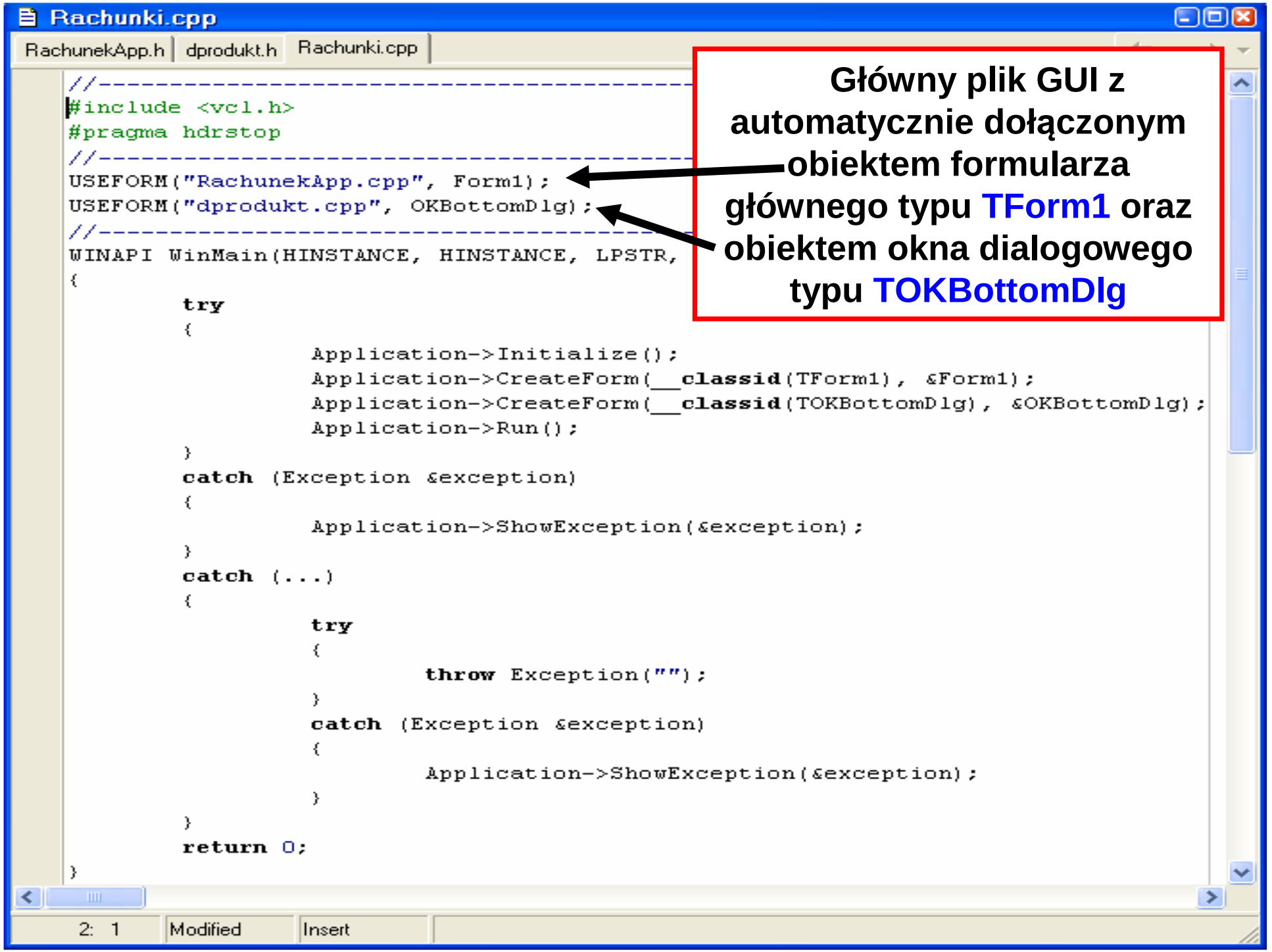
dprodukt.cpp

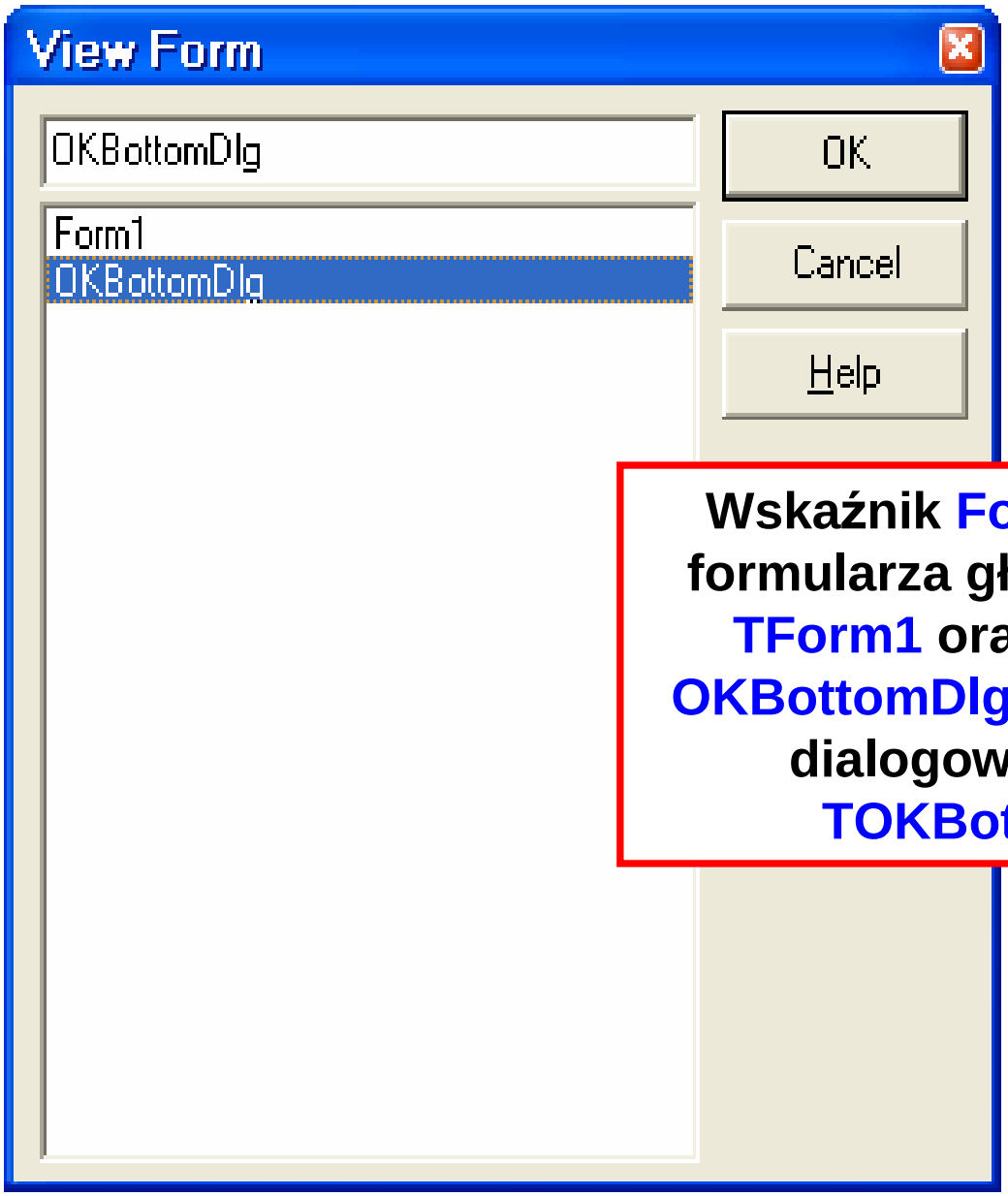
RachunekApp.h | dprodukt.h | Rachunki.cpp

```
//---------------------------------
#ifndef dproduktH
#define dproduktH
//---------------------------------
#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>
//---------------------------------------------------------
class TOKBottomDlg : public TForm
{
__published:
        TButton *OKBtn;
        TButton *CancelBtn;
        TBevel *Bevel1;
private:
public:
        virtual __fastcall TOKBottomDlg(TComponent* AOwner);
};
//---------------------------------------------------------
extern PACKAGE TOKBottomDlg *OKBottomDlg;
//---------------------------------------------------------
#endif
```

1: 1 | Insert

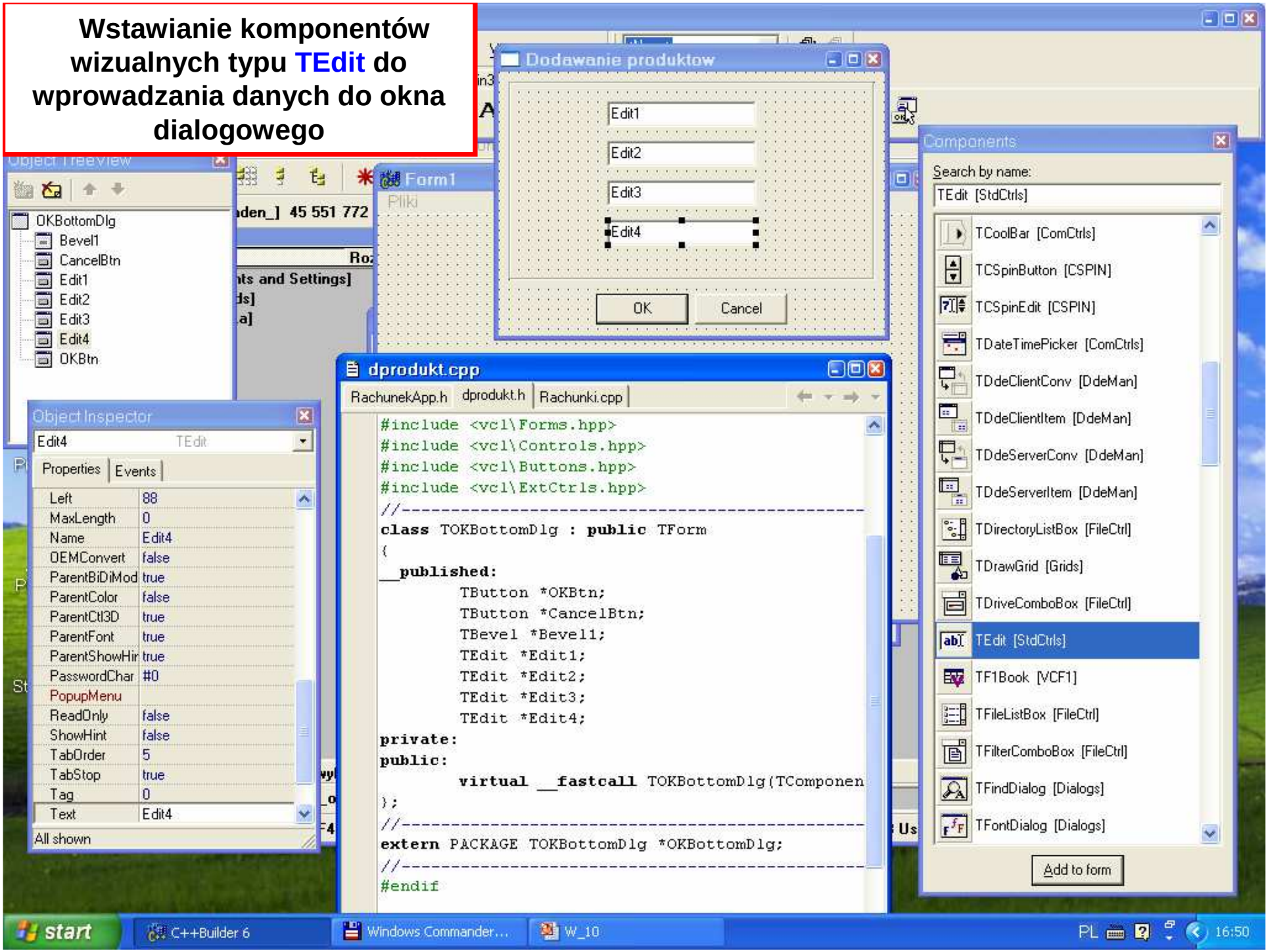
Utworzony automatycznie plik nagłówkowy dprodukt1.h standardowego okna dialogowego - po zmianie nazwy


Rachunki.cpp
RachunekApp.h
dprodukt.h
Rachunki.cpp
//---------------------------------------
#include <vcl.h>
#pragma hdrstop
//---------------------------------------
USEFORM("RachunekApp.cpp", Form1);
USEFORM("dprodukt.cpp", OKBottomDlg);
//---------------------------------------
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR,
{
try
{
Application->Initialize();
Application->CreateForm(__classid(TForm1), &Form1);
Application->CreateForm(__classid(TOKBottomDlg), &OKBottomDlg);
Application->Run();
}
catch (Exception &exception)
{
Application->ShowException(&exception);
}
catch (...)
{
try
{
throw Exception("");
}
catch (Exception &exception)
{
Application->ShowException(&exception);
}
}
return 0;
}
2: 1
Modified
Insert
Główny plik GUI z automatycznie dołączonym obiektem formularza głównego typu TForm1 oraz obiektem okna dialogowego typu TOKBottomDlg

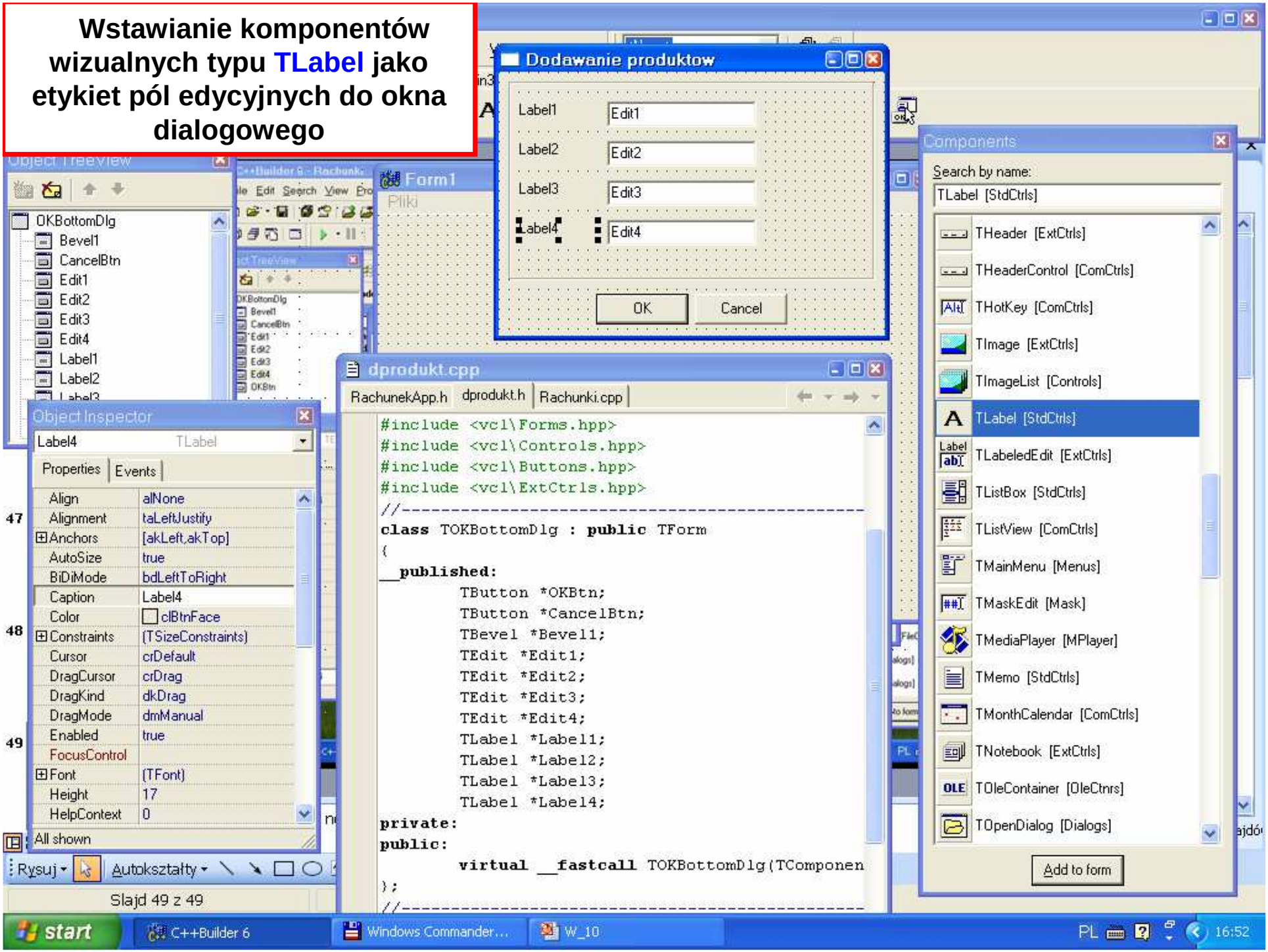


Wskaźnik Form1 obiektu formularza głównego typu TForm1 oraz wskaźnik OKBottomDlg obiektu okna dialogowego typu TOKBottomDlg

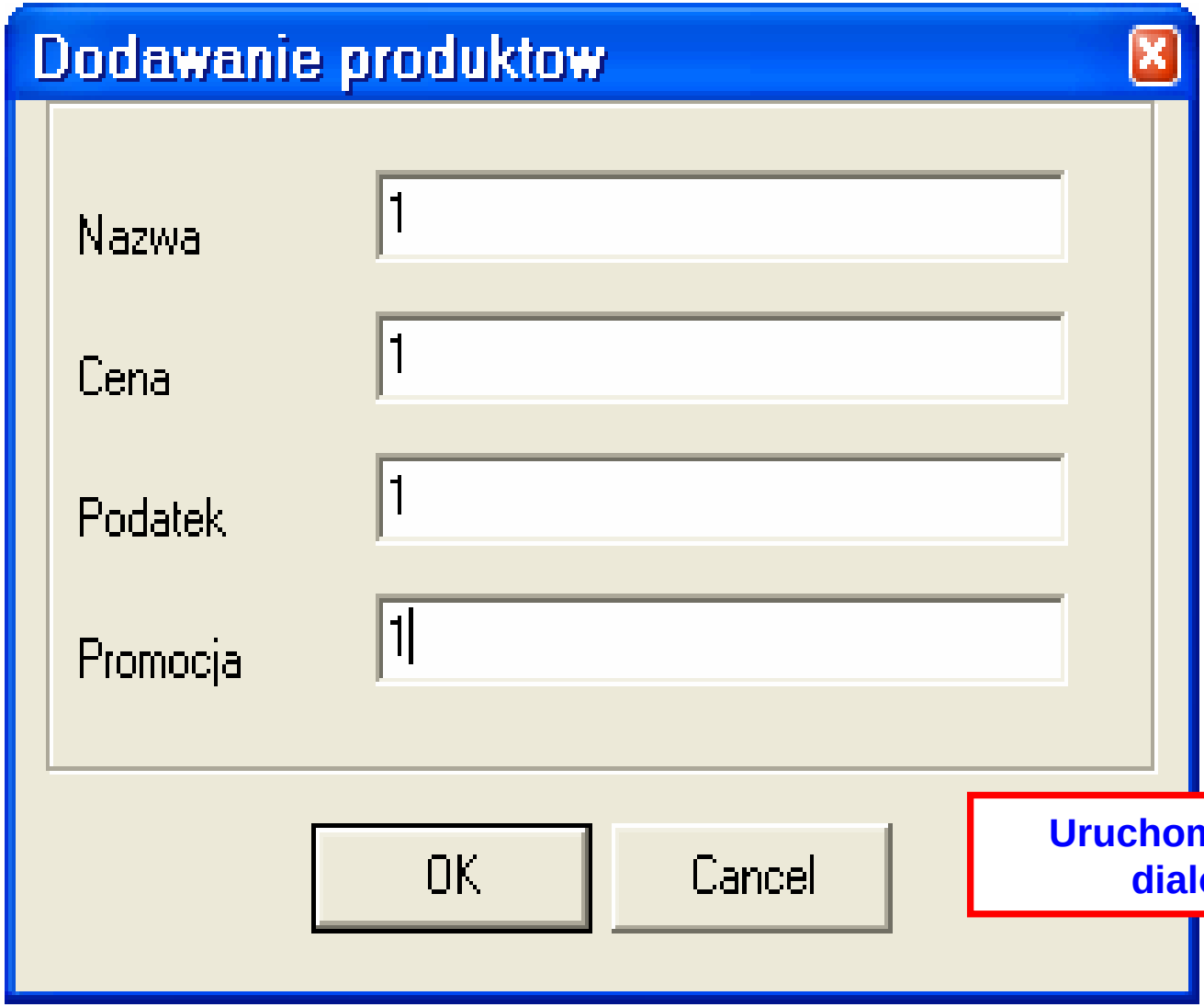
Wstawianie komponentów wizualnych typu TEdit do wprowadzania danych do okna dialogowego



Wstawianie komponentów wizualnych typu TLabel jako etykiet pól edycyjnych do okna dialogowego



Gotowy projekt okna dialogowego



Uruchomione okno dialogowe

dprodukt.cpp

RachunekApp.cpp | dprodukt.h

```
#ifndef dproduktH
#define dproduktH
//---------------------------------------------
#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 TOKBottomDlg : public TForm
{
__published:
        TButton *OKBtn;
        TButton *CancelBtn;
        TBevel *Bevel1;
        TEdit *Edit1;
        TEdit *Edit2;
        TEdit *Edit3;
        TEdit *Edit4;
        TLabel *Label1;
        TLabel *Label2;
        TLabel *Label3;
        TLabel *Label4;
private:
public: string tab[5]; //recznie - tablica na atrybuty lancuchowe produktu
        bool __fastcall Execute();  //recznie - obsluga okienka dialogowego
        virtual __fastcall TOKBottomDlg(TComponent* AOwner);
};
//---------------------------------------------------------------------------
extern PACKAGE TOKBottomDlg *OKBottomDlg;  //deklaracja globalnego wskaznika
```

Zawartość pliku nagłówkowego okna dialogowego po wykonaniu projektu

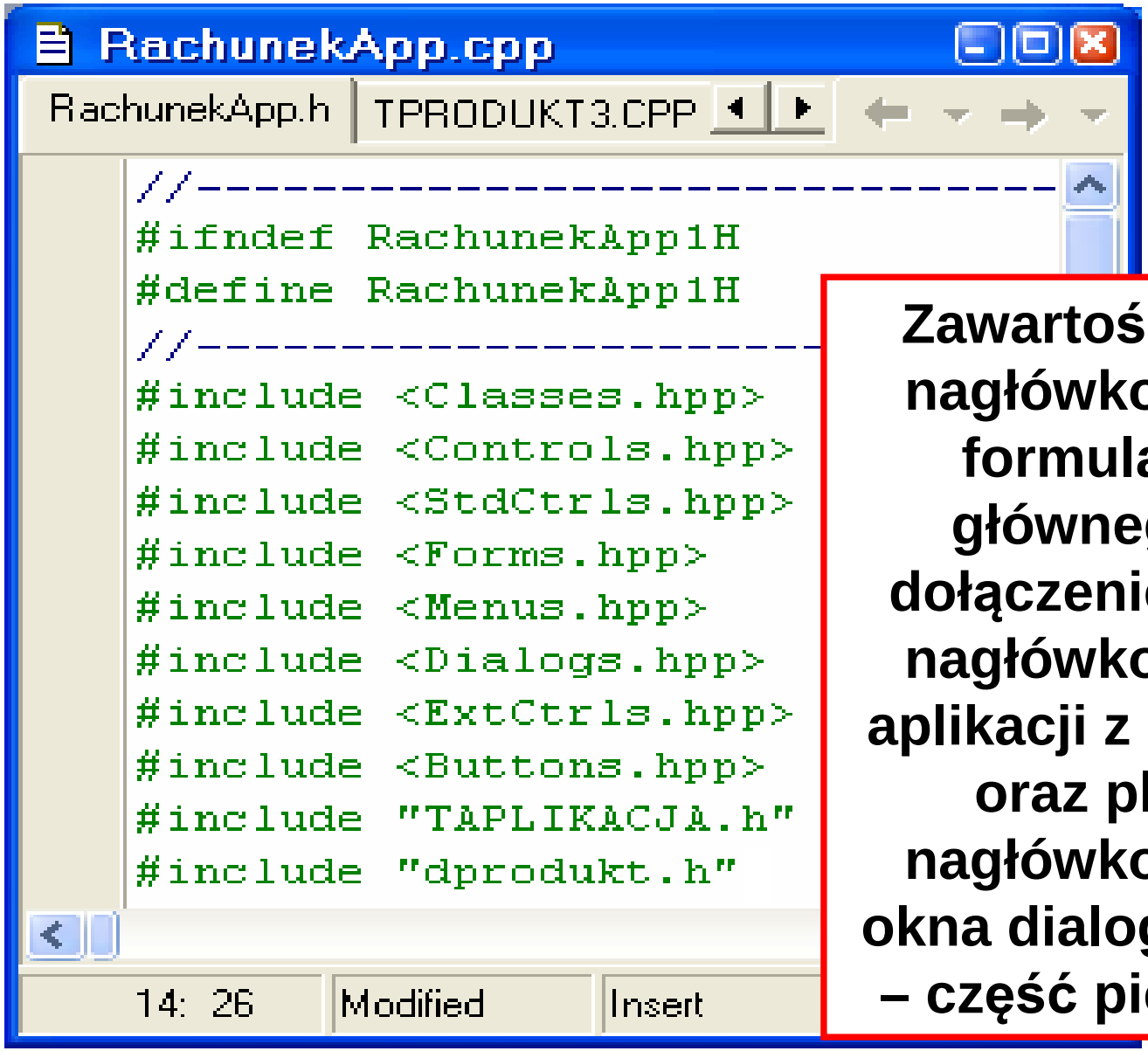
2: 1 | Insert

Definicja metody Execute() do wprowadzania danych w oknie dialogowym

dprodukt.cpp

RachunekApp.cpp | dprodukt.cpp

```
#include <vcl.h>
#pragma hdrstop
#include "dprodukt.h"
//---------------------------
#pragma resource "*.dfm"
TOKBottomDlg *OKBottomDlg;    //wskaznik globalny okna dialogowego
//---------------------------------------------------------------------------
__fastcall TOKBottomDlg::TOKBottomDlg(TComponent* AOwner): TForm(AOwner)
    {      }
//---------------------------------------------------------------------------
bool __fastcall TOKBottomDlg::Execute()    //recznie
{   Edit1->Text="";          //recznie
    Edit2->Text="";
    Edit3->Text="";
    Edit4->Text="";
    if (ShowModal() ==mrOk) //przejscie do trybu modalnego, ktory konczy sie
                            //nacisnieciem klawisza OK lub Cancel;
    {   //pobranie atrybutow produktu z pol tekstowych wejsciowych
      tab[1]=Edit1->Text.c_str();     //nazwa
      tab[2]=Edit2->Text.c_str();     //cena netto
      tab[3]=Edit3->Text.c_str();     //podatek
      tab[4]=Edit4->Text.c_str();     //promocja
      //oraz wyznaczenie typu produktu w tab[0] w zaleznosci, czy wprowadzono
      //podatek oraz promocje
      if(tab[3]!="")
       if(tab[4]!="")
        tab[0]="4";          //jest promocja i podatek - typ TProdukt4
       else tab[0]="2";      //jest podatek - typ TProdukt2
      else
        if(tab[4]!="")
          { tab[0]="3";        //jest promocja - typ TProdukt3
            tab[3]=tab[4];     //korekta przekazania promocji, jesli brak podatk
          }
        else tab[0]="1";     //brak podatku i promocji - typ TProdukt1
      return true;}
  else
   return false;
}
```



Zawartość pliku nagłówkowego formularza głównego – dołączenie pliku nagłówkowego aplikacji z danymi oraz pliku nagłówkowego okna dialogowego – część pierwsza

C:\Settings\dydaktyka\Programowanie_obiektowe\w_10_2\RachunekApp.cpp

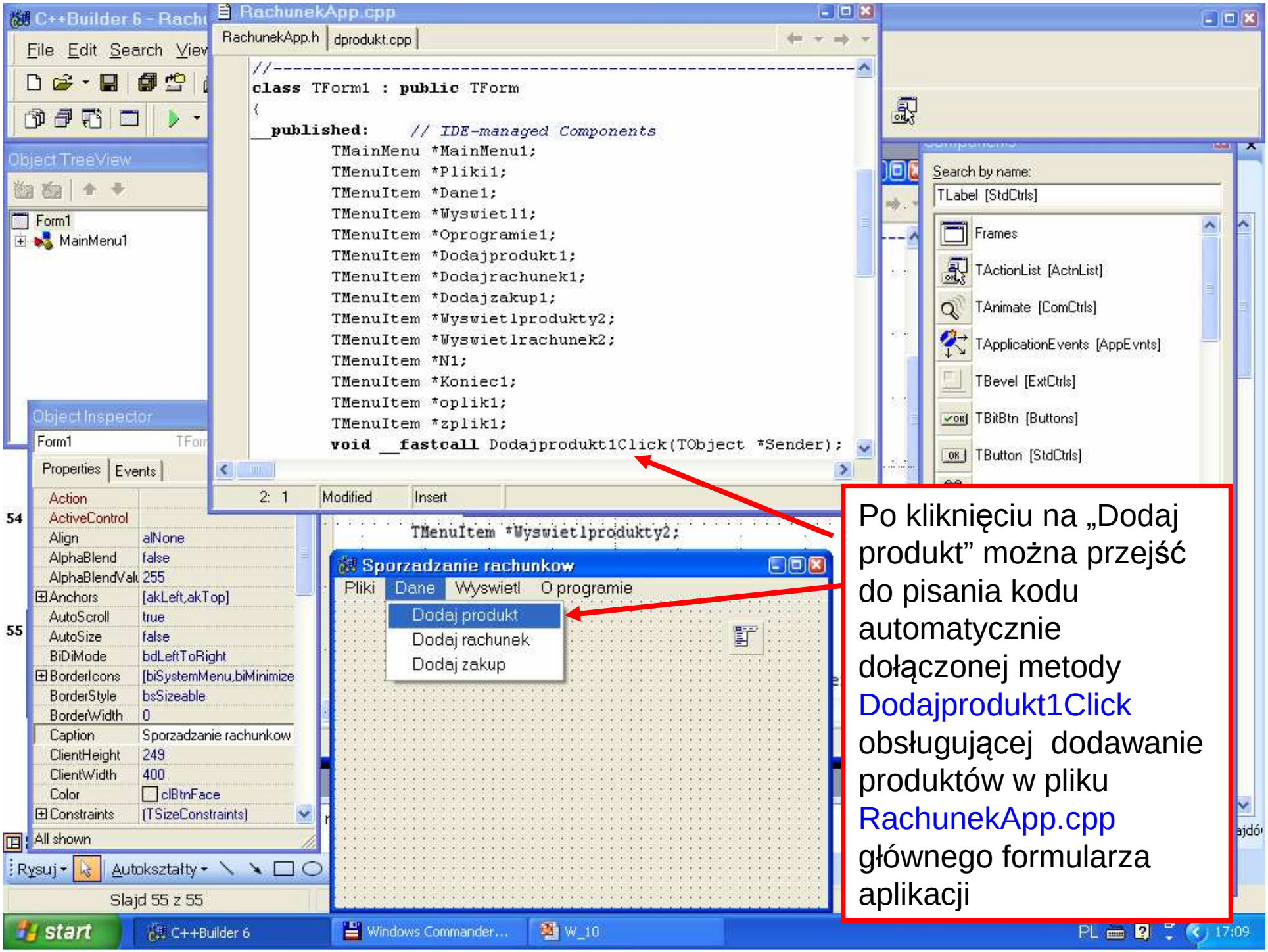
RachunekApp1.cpp | RachunekApp.h

```
#include <Dialogs.hpp>
#include <ExtCtrls.hpp>
#include <Buttons.hpp>
#include "TAplikacja.h"
#include "dprodukt.h"
//---------------------------------------------------------------------------
class TForm1 : public TForm
{
__published:    // IDE-managed Components
        TMainMenu *MainMenu1;
        TMenuItem *Pliki1;
        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);
private:        // User declarations
public:         // User declarations
        TAplikacja aplikacja;     //połączenie z obiektem uzytkownika
        __fastcall TForm1(TComponent* Owner);
};
//---------------------------------------------------------------------------
extern PACKAGE TForm1 *Form1;   //deklaracja globalnego wskaznika
//---------------------------------------------------------------------------
#endif
```

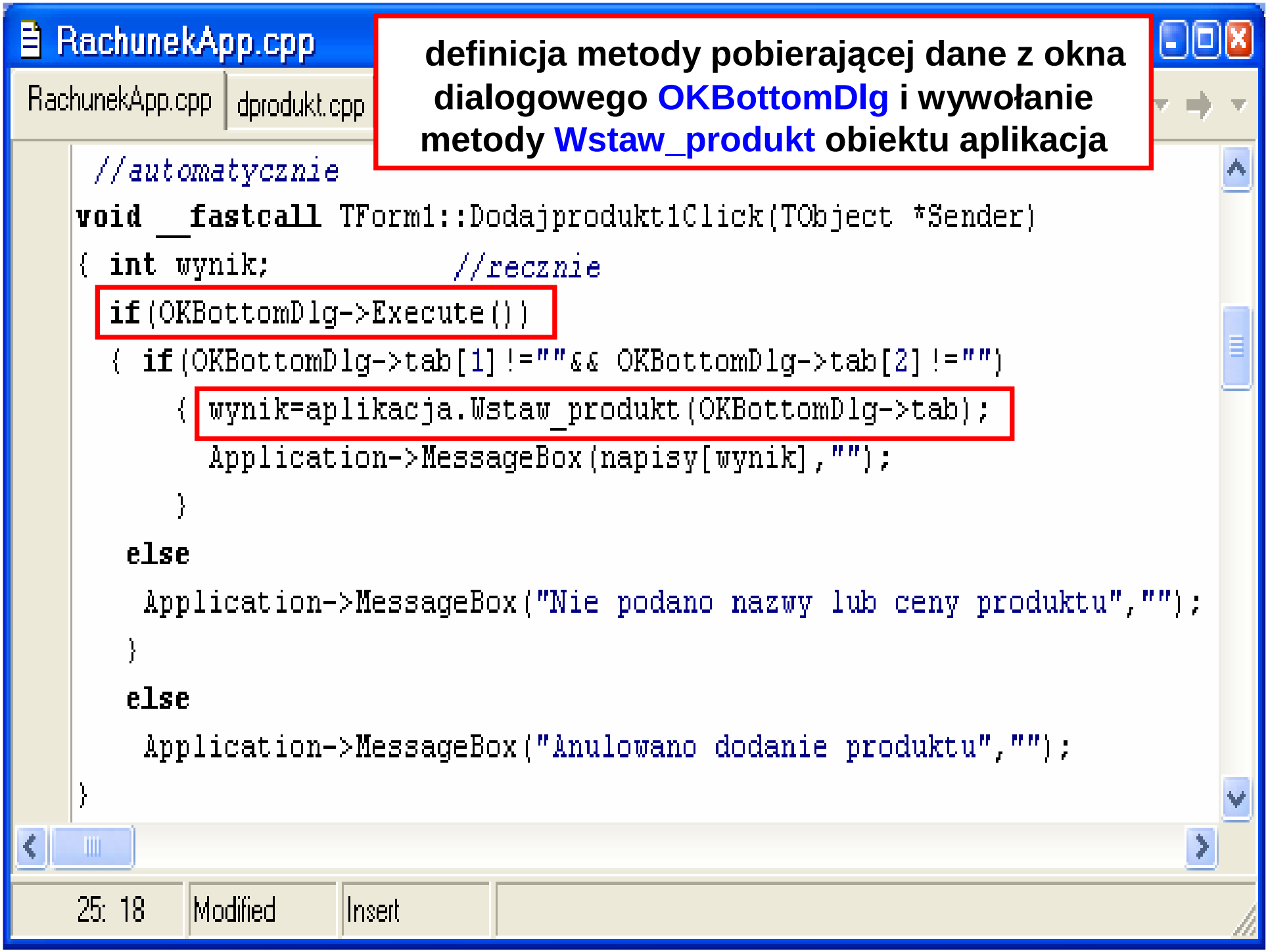
34: 1 | Modified | Insert

Zawartość pliku nagłówkowego okna formularza głównego po wykonaniu projektu – cd.

Połączenie GUI z obiektami aplikacji za pomocą obiektu hermetyzujacego aplikacja typu TAplikacja



Po kliknięciu na „Dodaj produkt” można przejść do pisania kodu automatycznie dołączonej metody Dodajprodukt1Click obsługującej dodawanie produktów w pliku RachunekApp.cpp głównego formularza aplikacji


RachunekApp.cpp
RachunekApp.cpp
dprodukt.cpp
definicja metody pobierającej dane z okna dialogowego OKBottomDlg i wywołanie metody Wstaw_produkt obiektu aplikacja
//automatycznie
void __fastcall TForm1::Dodajprodukt1Click(TObject *Sender)
{ int wynik; //recznie
if(OKBottomDlg->Execute())
{ if(OKBottomDlg->tab[1]!=""&& OKBottomDlg->tab[2]!="")
{ wynik=aplikacja.Wstaw_produkt(OKBottomDlg->tab);
Application->MessageBox(napisy[wynik],"");
}
else
Application->MessageBox("Nie podano nazwy lub ceny produktu","");
}
else
Application->MessageBox("Anulowano dodanie produktu","");
}
25: 18
Modified
Insert

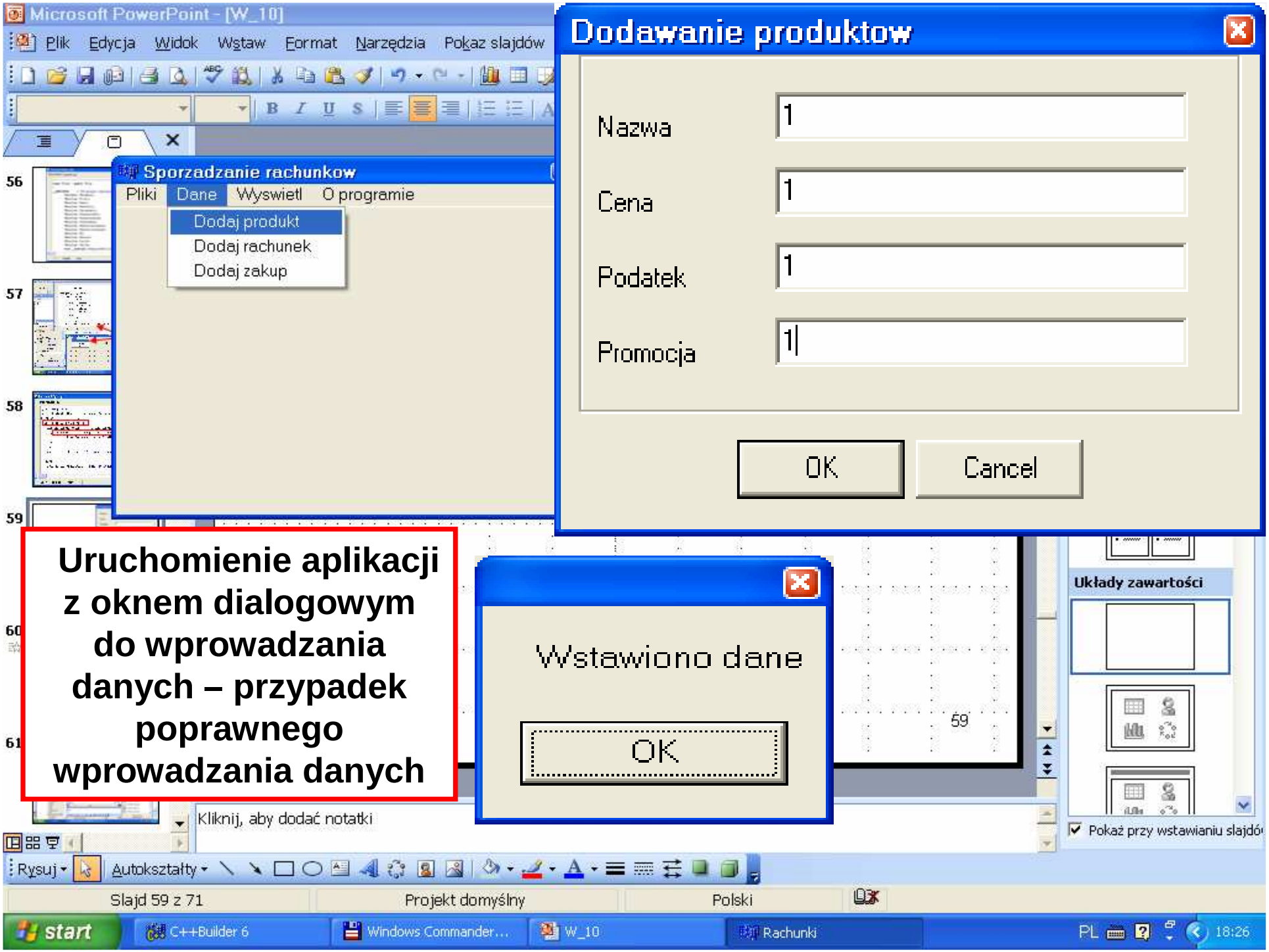
RachunekApp.cpp

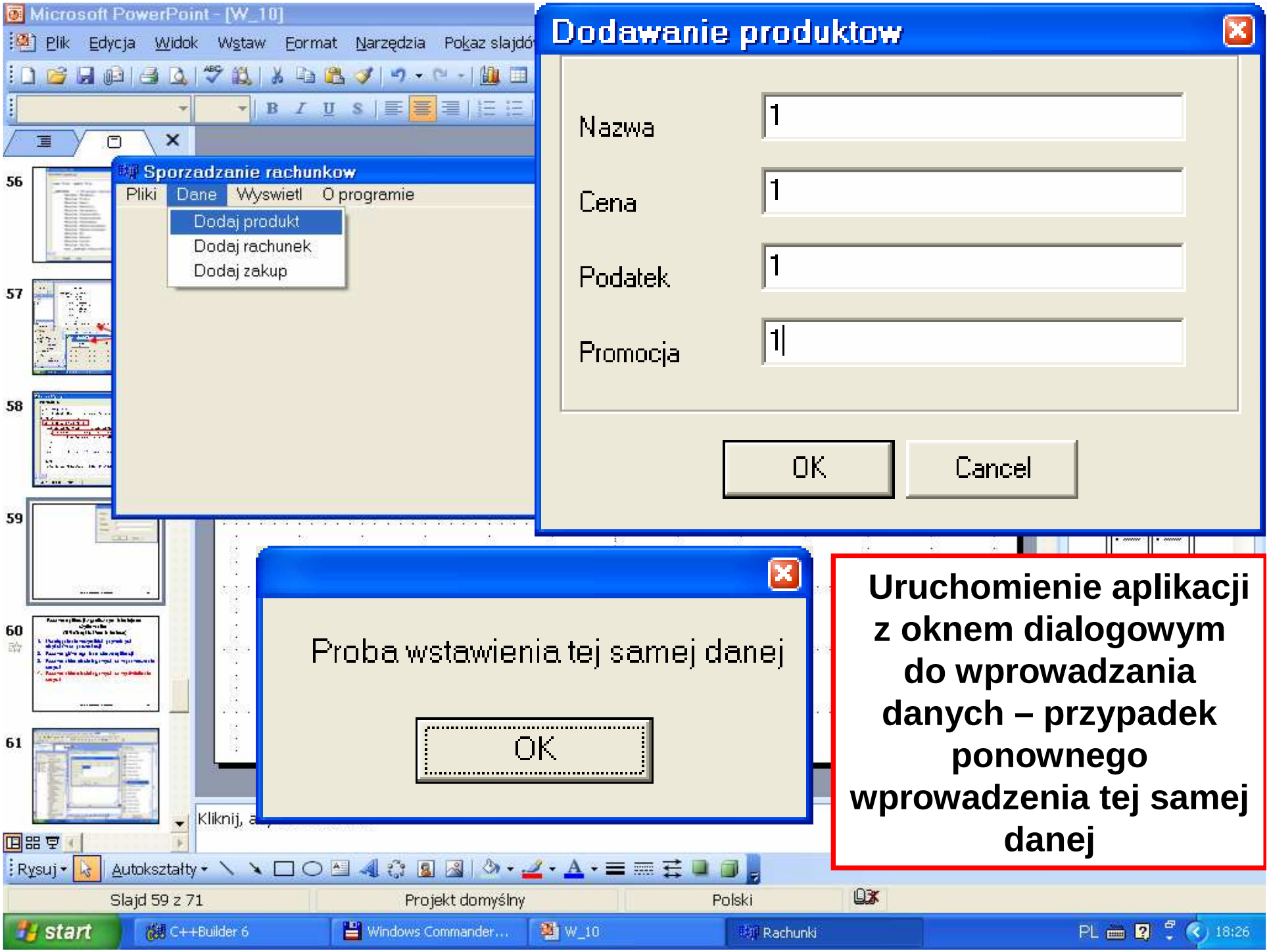
dprodukt.cpp | drachunek.cpp | RachunekApp.cpp

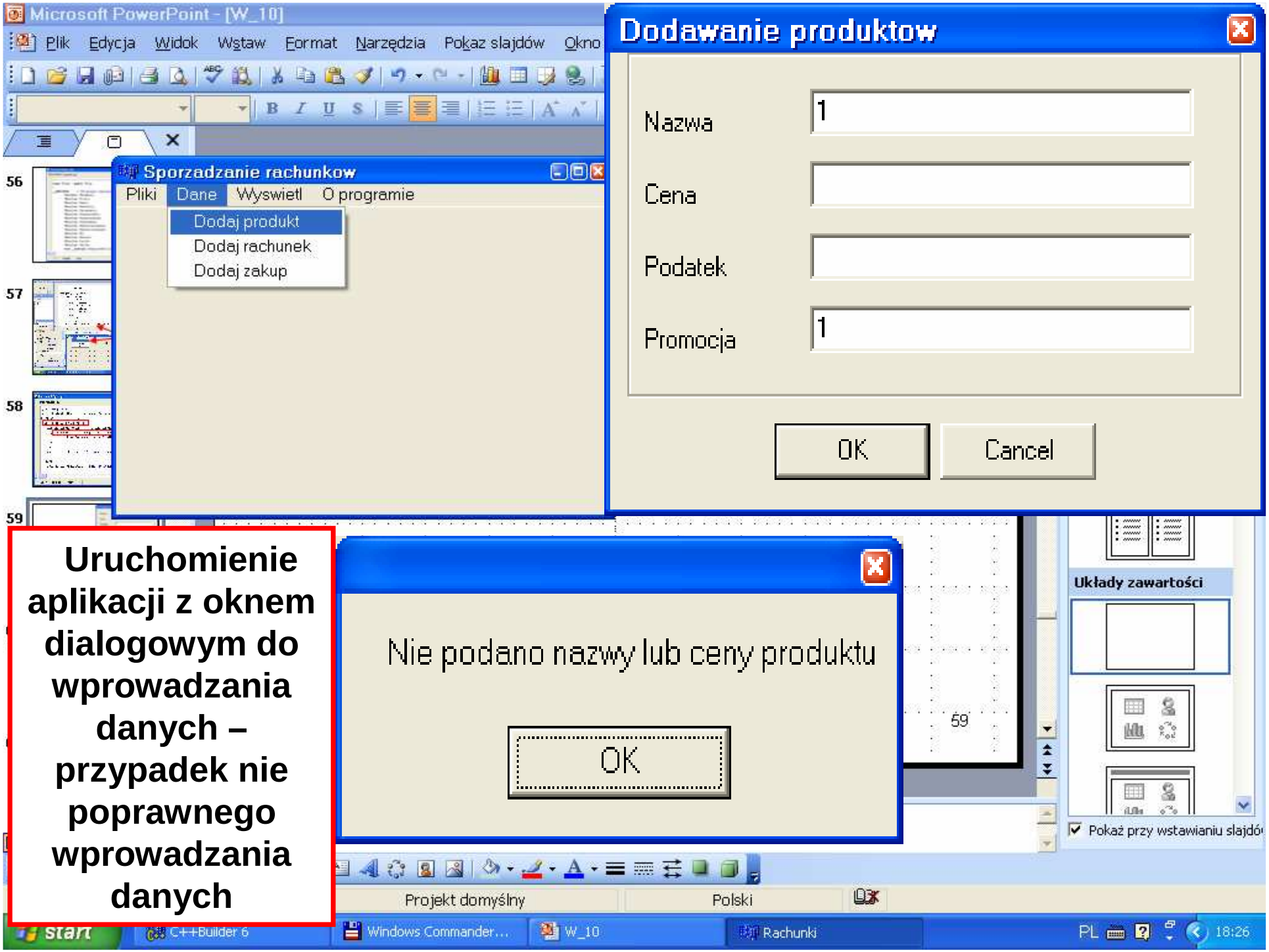
Zawartość pliku modułowego okna formularza głównego – wstawienie tablicy napisy z komunikatami

```
#include <vcl.h>
#pragma hdrstop
#include "RachunekApp.h"
//---------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;  //wskaznik globalny okna formularza
char* napisy[]={"Brak miejsca w kolekcji",
                "Proba wstawienia tej samej danej",
                "Wstawiono dane",
                "Brak pamieci",
                "Brak rachunku",
                "Brak produktu",
                "Wybrany produkt kupiono wczesniej - powiekszono jego ilosc"};
//---------------------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)    //automatycznie
        : TForm(Owner)
{     }
```

5: 1 | Modified | Insert


Microsoft PowerPoint - [W_10]
Plik Edycja Widok Wstaw Format Narzędzia Pokaz slajdów
Sporzadzanie rachunkow
Pliki Dane Wyswietl O programie
Dodaj produkt
Dodaj rachunek
Dodaj zakup
Dodawanie produktow
Nazwa
1
Cena
1
Podatek
1
Promocja
1
OK
Cancel
Wstawiono dane
OK
Uruchomienie aplikacji z oknem dialogowym do wprowadzania danych – przypadek poprawnego wprowadzania danych
Układy zawartości
Kliknij, aby dodać notatki
Rysuj Autokształty
Slajd 59 z 71
Projekt domyślny
Polski
start
C++Builder 6
Windows Commander...
W_10
Rachunki
PL
18:26


Microsoft PowerPoint - [W_10]
Plik Edycja Widok Wstaw Format Narzędzia Pokaz slajdów
Sporzadzanie rachunkow
Pliki Dane Wyswietl O programie
Dodaj produkt
Dodaj rachunek
Dodaj zakup
Dodawanie produktow
Nazwa
1
Cena
1
Podatek
1
Promocja
1
OK
Cancel
Proba wstawienia tej samej danej
OK
Uruchomienie aplikacji z oknem dialogowym do wprowadzania danych – przypadek ponownego wprowadzenia tej samej danej
Rysuj
Autokształty
Slajd 59 z 71
Projekt domyślny
Polski
start
C++Builder 6
Windows Commander...
W_10
Rachunki
PL
18:26

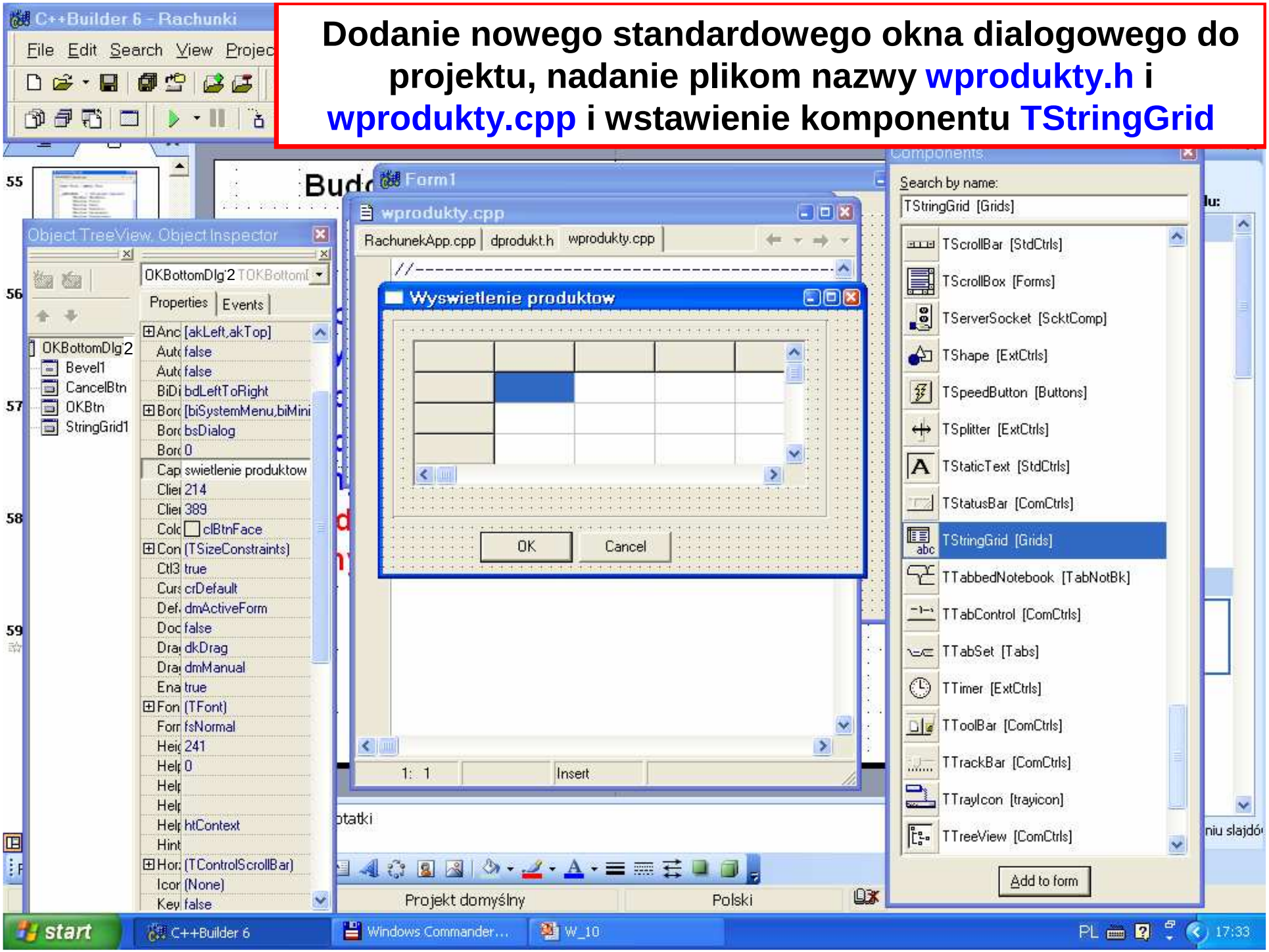


Uruchomienie aplikacji z oknem dialogowym do wprowadzania danych – przypadek nie poprawnego wprowadzania danych

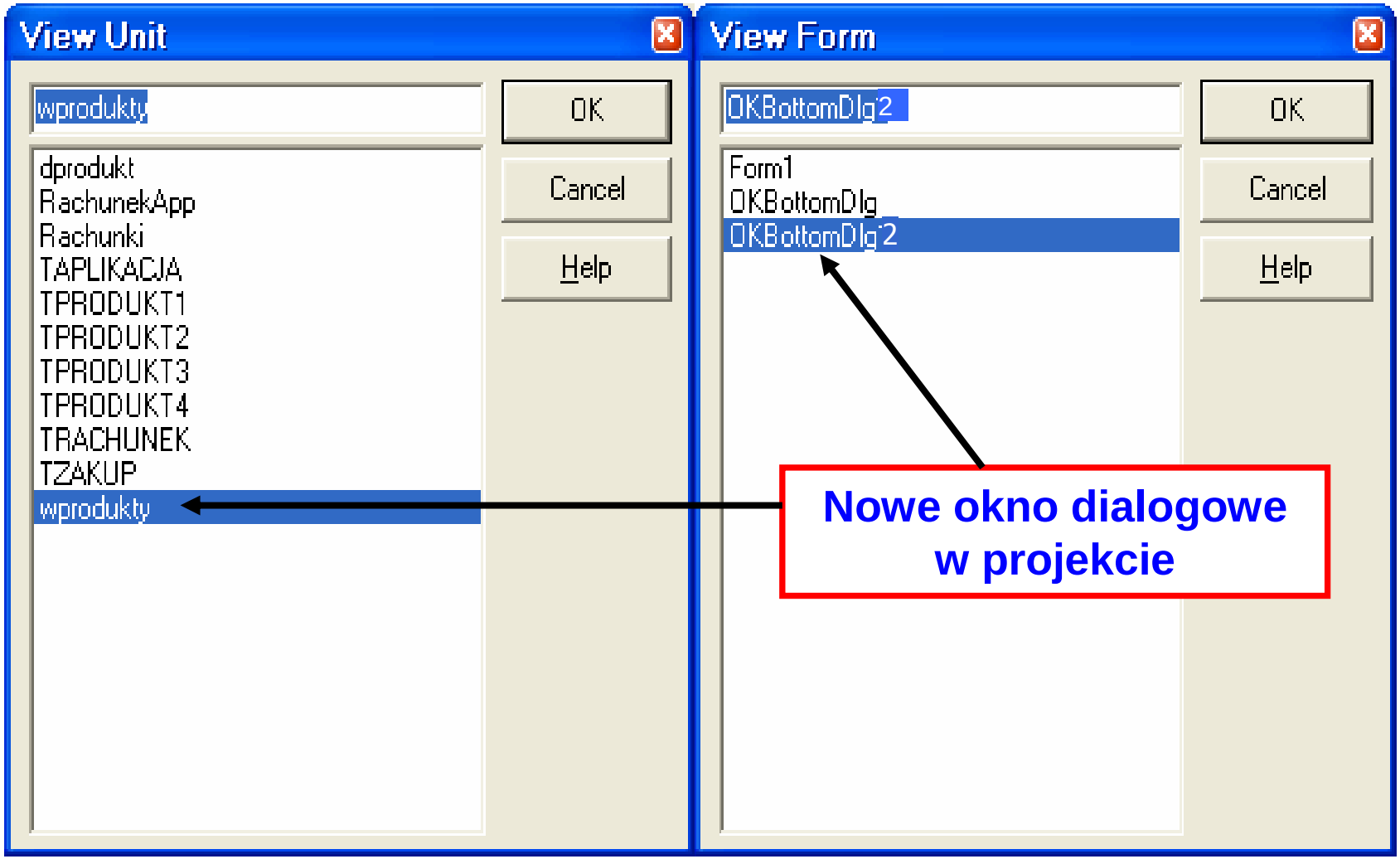
Budowa aplikacji z graficznym interfejsem użytkownika - GUI (Graphic User Interface)

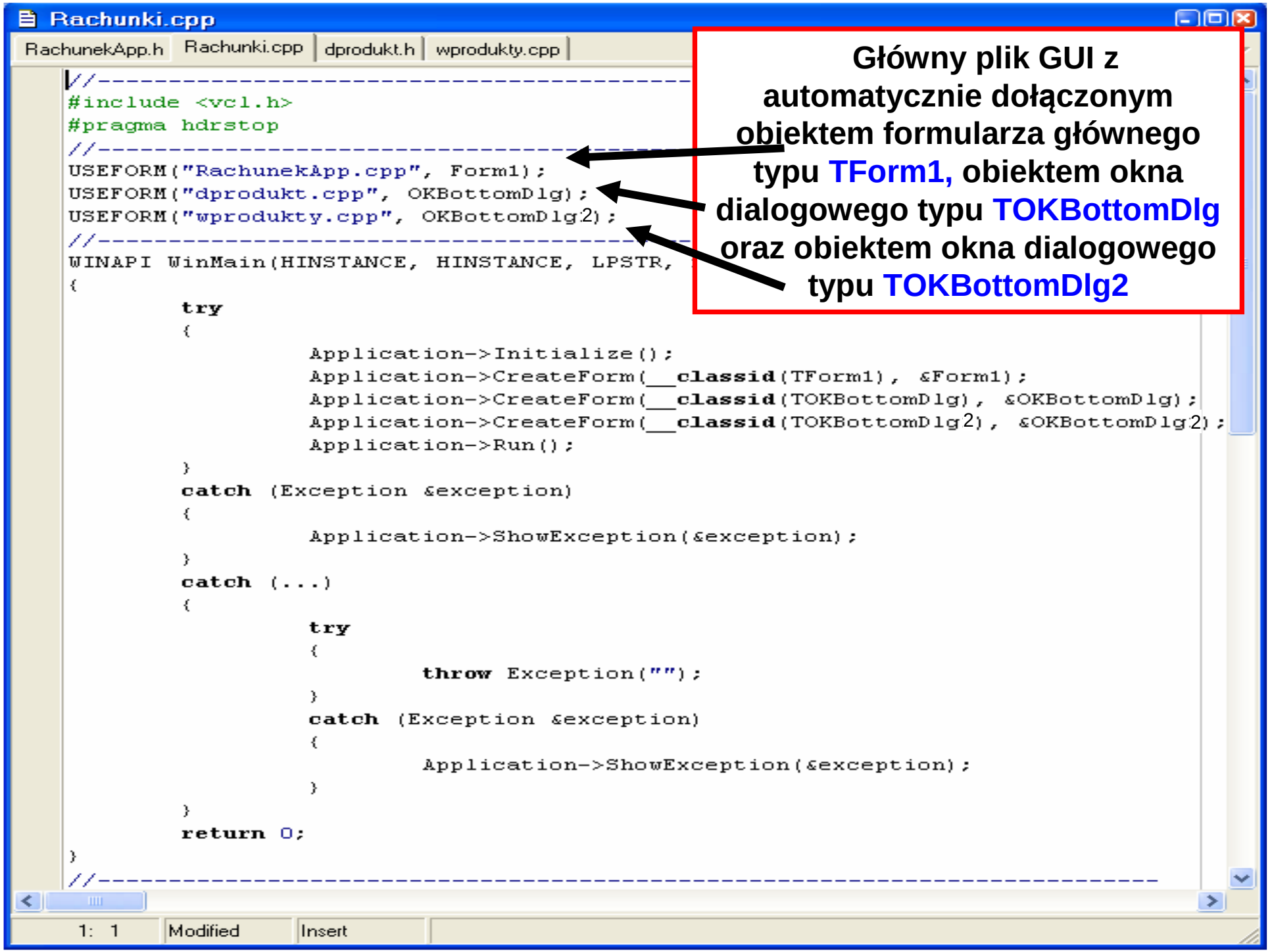
1. Udostępnianie wszystkich prywatnych atrybutów do prezentacji, wprowadzenie standardu nazewnictwa plików – nazwy plików aplikacji poprzedzone literą T
2. Budowa głównego formularza GUI
3. Budowa okienek dialogowych do wprowadzania danych
4. Budowa okienek dialogowych do wyświetlania danych

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



Dodanie nowego okna dialogowego do projektu




Rachunki.cpp
RachunekApp.h
Rachunki.cpp
dprodukt.h
wprodukty.cpp
//---------------------------------------
#include <vcl.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;
}
//---------------------------------------
1: 1
Modified
Insert
Główny plik GUI z automatycznie dołączonym obiektem formularza głównego typu TForm1, obiektem okna dialogowego typu TOKBottomDlg oraz obiektem okna dialogowego typu TOKBottomDlg2

wprodukty.cpp

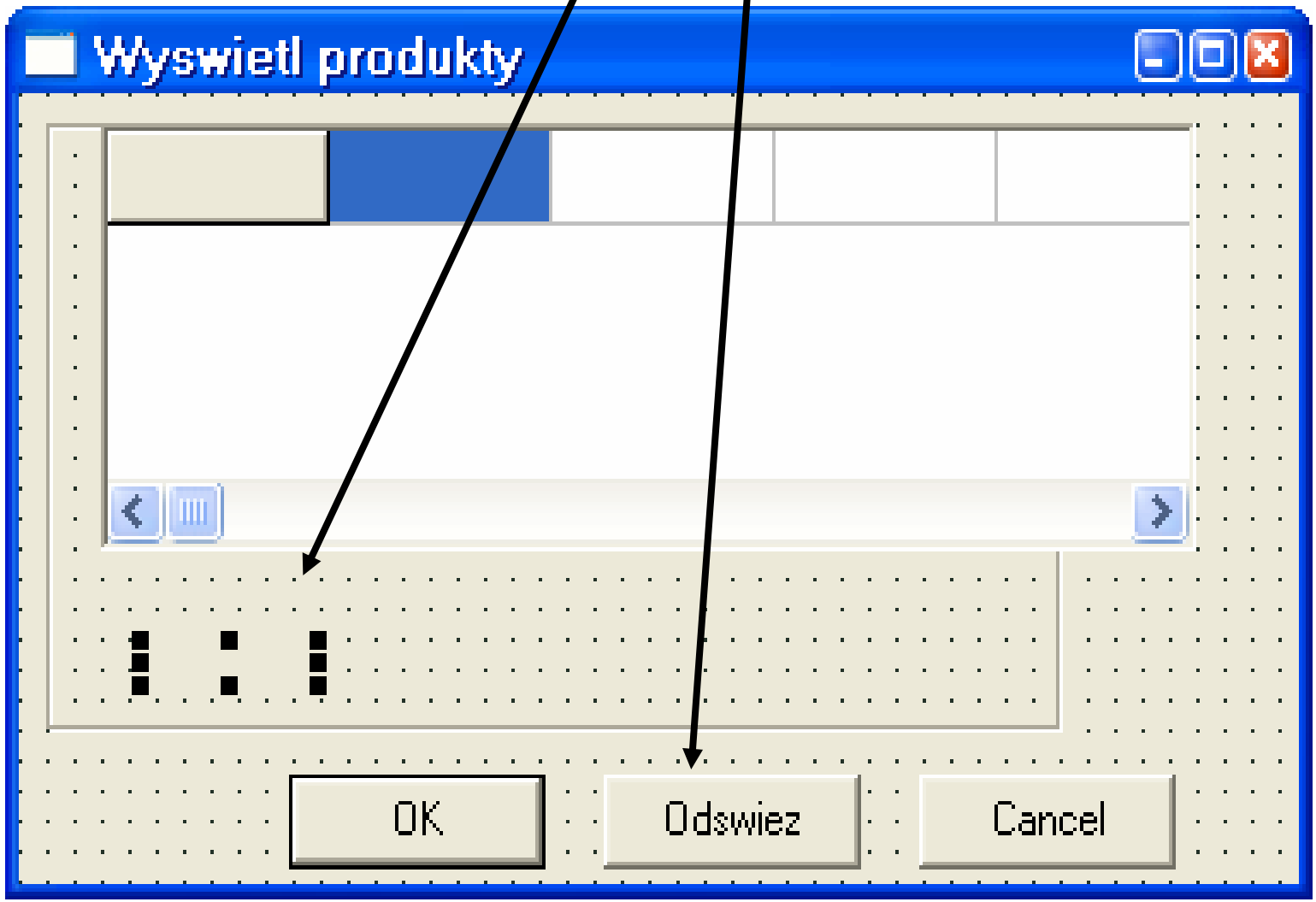
RachunekApp.h | Rachunki.cpp | dprodukt.h | wprodukty.h

```
//---------------------------------------------------------------------------
#ifndef wproduktyH
#define wproduktyH
//---------------------------------------------------------------------------
#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 <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 nagłówkowy standardowego okna dialogowego po wstawieniu komponentu TStringGrid

1: 1 | Insert

Wstawienie przycisku typu TButton do odświeżania zawartości okna oraz pola komunikatów typu TStaticText



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;  //recznie - zmienna na wybrany rowek oprocz pierwszego
        string tab[5]; //recznie - tablica na atrybuty lancuchowe produktu
        bool _fastcall Execute();  //recznie - obsluga okienka dialogowego
        virtual __fastcall TOKBottomDlg2(TComponent* AOwner);
};
//---------------------------------------------------------------------------
extern PACKAGE TOKBottomDlg2 *OKBottomDlg2; //deklaracja globalnego wskaznika
//---------------------------------------------------------------------------
#endif
```

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

12: 55 Insert

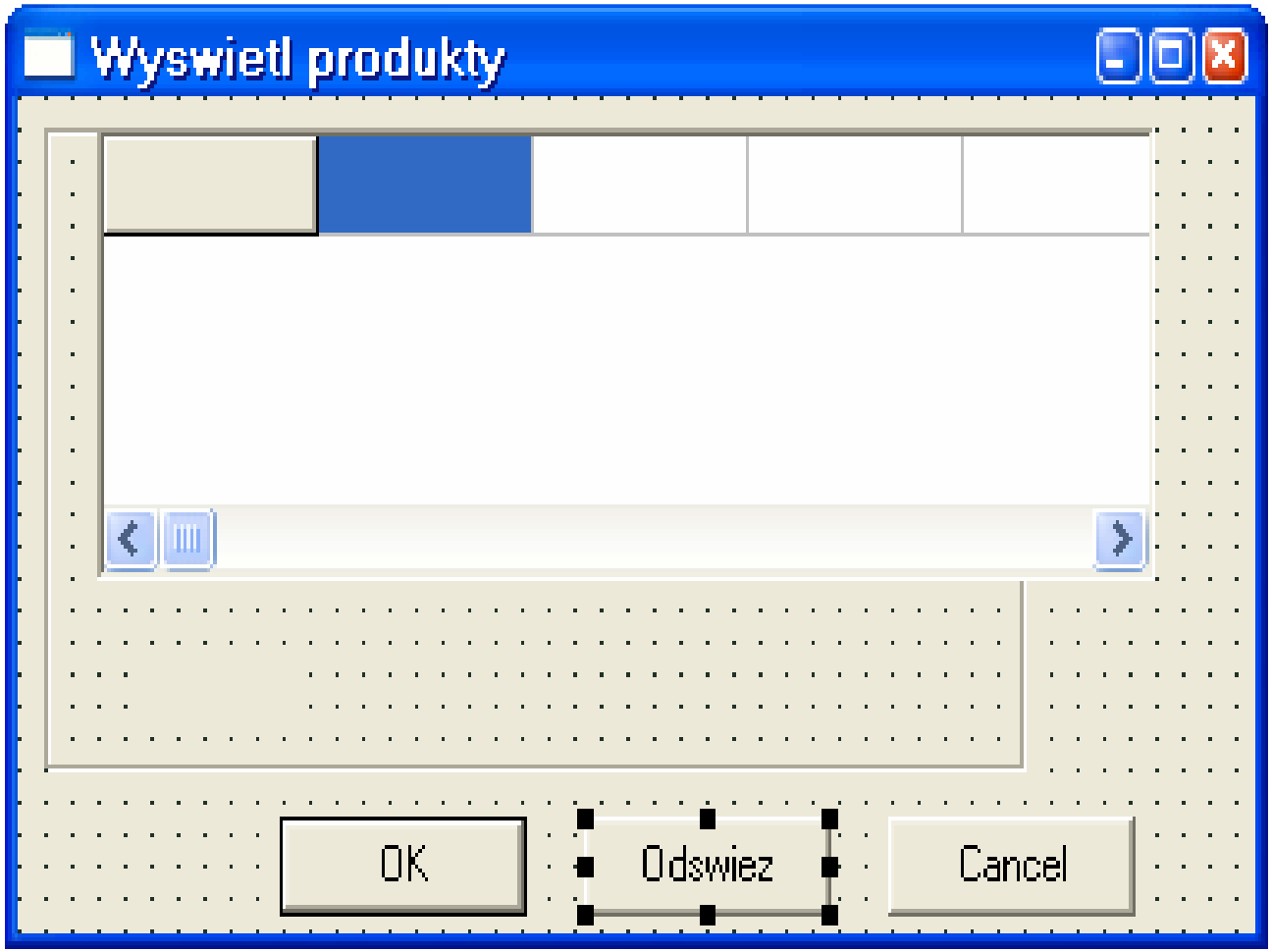
wprodukty.cpp

RachunekApp.cpp | wprodukty.cpp | dprodukt.cpp

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

```
#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";   //naglówka 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, ktory konczy sie
   return true;  //nacisnieciem klawisza OK lub Cancel;
 }
```

2: 49 | Insert



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;  //recznie - zmienna na wybrany rowek oprocz pierwszego
        string tab[5]; //recznie - tablica na atrybuty lancuchowe produktu
        bool _fastcall Execute();  //recznie - obsluga okienka dialogowego
        virtual __fastcall TOKBottomDlg2(TComponent* AOwner);
};
//---------------------------------------------------------------------------
extern PACKAGE TOKBottomDlg2 *OKBottomDlg2; //deklaracja globalnego wskaznika
//---------------------------------------------------------------------------
#endif
```

Metoda wywoływana podczas działania metody Execute()

12: 55 | Insert

wprodukty.cpp

Definicja metody wstawiającej dane z kolekcji produkty do komponentu typu TStringGrid

RachunekApp.cpp | wprodukty.cpp | dprodukt.cpp

```
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 poczatku
   while(produkty.Koniec()==0)        //odswiezanie  zawartosci komorek tabeli
   { produkt=produkty.Podaj_nast();
     i++;                             //zmienna do liczenia rowkow tabeli
     StringGrid1->RowCount=i+1;       //dodanie nowego rowka do tabeli
     StringGrid1->Cells[0][i]=i; //numer pozycji w rachunku
     StringGrid1->Cells[1][i]=produkt->Podaj_nazwe().c_str();
     StringGrid1->Cells[2][i]=gcvt(produkt->Podaj_cene(),3,lan);
     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,lan);
     pom=produkt->Podaj_promocje();
     if (pom==-1)          //jesli brak promocji wyswietla sie pusty lancuch
      StringGrid1->Cells[4][i]="";
     else
      StringGrid1->Cells[4][i]=gcvt(produkt->Podaj_promocje(),3,lan);
   }
   StringGrid1->Visible=true;
}
```

89: 26 | Modified | Insert

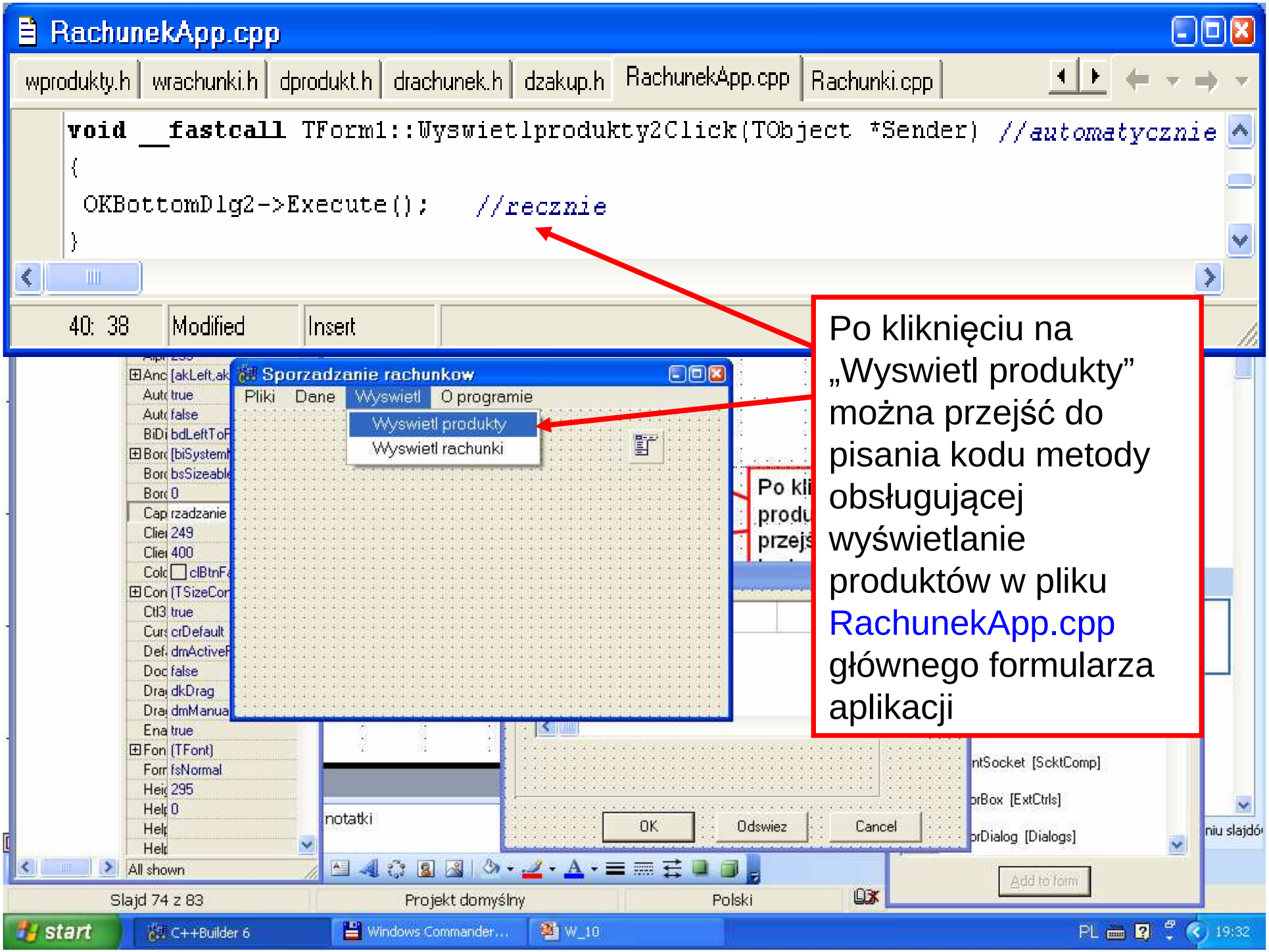
C:\Settings\dydaktyka\Programowanie_obiektowe\w_10_2\RachunekApp.cpp

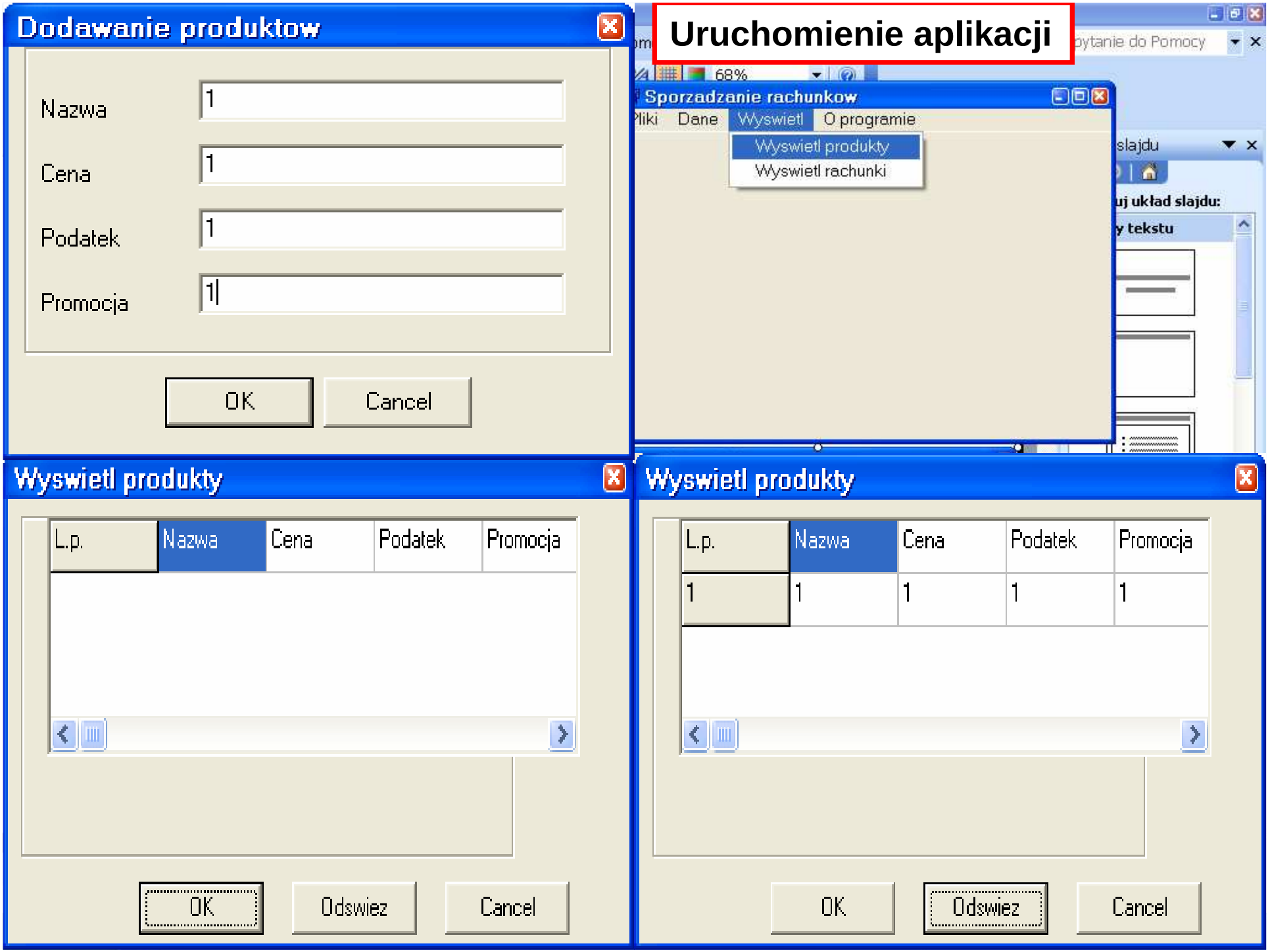
RachunekApp1.h | RachunekApp.h

```
#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 *Pliki1;
        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

35: 9 | Modified | Insert


RachunekApp.cpp
wprodukty.h
wrachunki.h
dprodukt.h
drachunek.h
dzakup.h
RachunekApp.cpp
Rachunki.cpp
void __fastcall TForm1::Wyswietlprodukty2Click(TObject *Sender) //automatycznie
{
OKBottomDlg2->Execute(); //recznie
}
40: 38
Modified
Insert
Sporzadzanie rachunkow
Pliki
Dane
Wyswietl
O programie
Wyswietl produkty
Wyswietl rachunki
notatki
OK
Odswiez
Cancel
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 produktow
Nazwa
1
Cena
1
Podatek
1
Promocja
1
OK
Cancel
Sporzadzanie rachunkow
Pliki
Dane
Wyswietl
O programie
Wyswietl produkty
Wyswietl rachunki
Wyswietl produkty
L.p.
Nazwa
Cena
Podatek
Promocja
OK
Odswiez
Cancel
Wyswietl produkty
L.p.
Nazwa
Cena
Podatek
Promocja
1
1
1
1
1
OK
Odswiez
Cancel

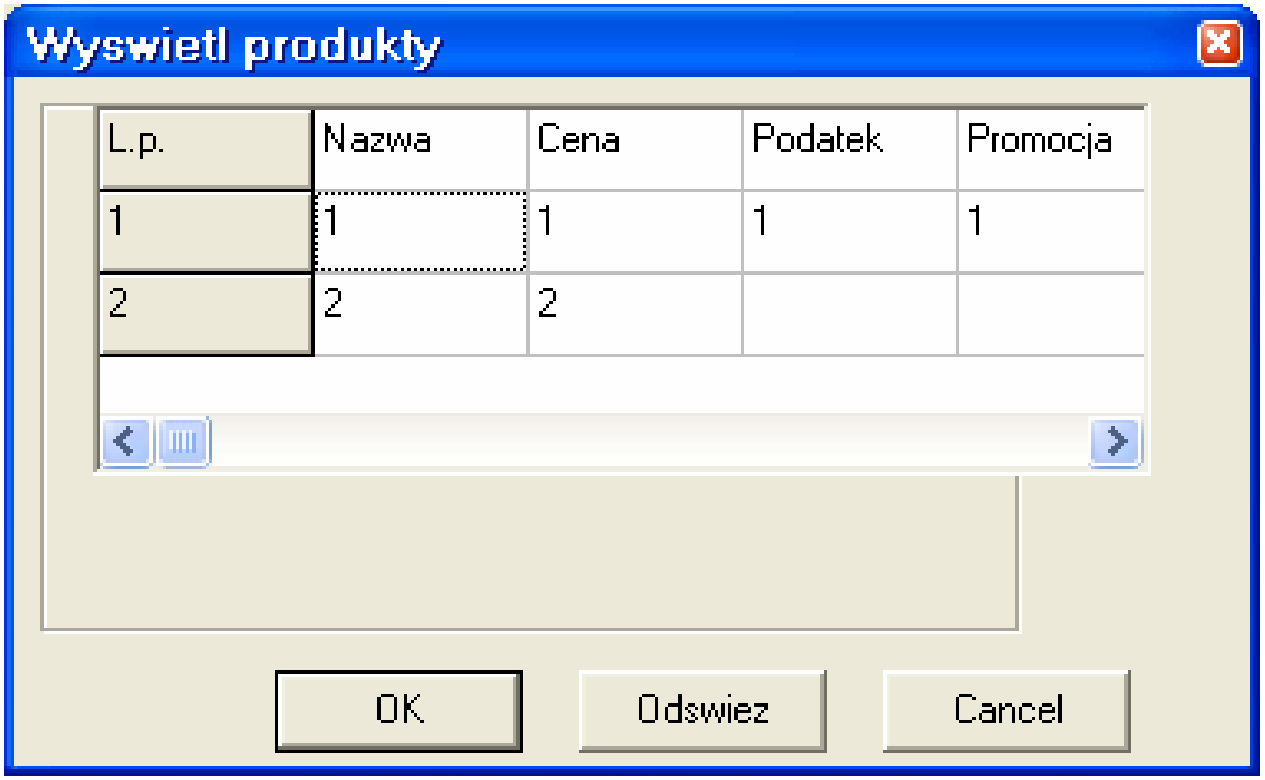
wprodukty.cpp

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;  //recznie - zmienna na wybrany rowek oprocz pierwszego
        string tab[5]; //recznie - tablica na atrybuty lancuchowe produktu
        bool _fastcall Execute();  //recznie - obsluga okienka dialogowego
        virtual __fastcall TOKBottomDlg2(TComponent* AOwner);
};
//---------------------------------------------------------------------------
extern PACKAGE TOKBottomDlg2 *OKBottomDlg2; //deklaracja globalnego wskaznika
//---------------------------------------------------------------------------
#endif
```

12: 55 Insert

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()


Wyswietl produkty
L.p.
Nazwa
Cena
Podatek
Promocja
1
1
1
1
1
2
2
2
OK
Odswiez
Cancel

wprodukty.cpp

RachunekApp.cpp | wprodukty.cpp | dprodukt.cpp

```
void __fastcall TOKBottomDlg2::StringGrid1Click(TObject *Sender) //automatycznie
{  char pom[10];                          //ręcznie
   TStrings* 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 oraz promocje
   if(tab[3]!="")
      if(tab[4]!="")
       { tab[0]="4";          //jest promocja i podatek = typ TProdukt4
         tab[2]=cena_netto4(tab[2],tab[3],tab[4]);} //obliczanie ceny netto
      else
        { tab[0]="2";         //jest promocja = typ TProdukt2
            tab[2]=cena_netto2(tab[2],tab[3]);}    //obliczanie ceny netto
   else
      if(tab[4]!="")
        { tab[0]="3";         //jest promocja = typ TProdukt3
          tab[2]=cena_netto3(tab[2],tab[4]);       //obliczanie ceny netto
          tab[3]=tab[4];   //korekta przekazania promocji, gdy brak podatku
          }
      else
          tab[0]="1";         //brak promocji i podatku - jest TProdukt1
}
```

89: 26 | Modified | Insert

wprodukty.cpp

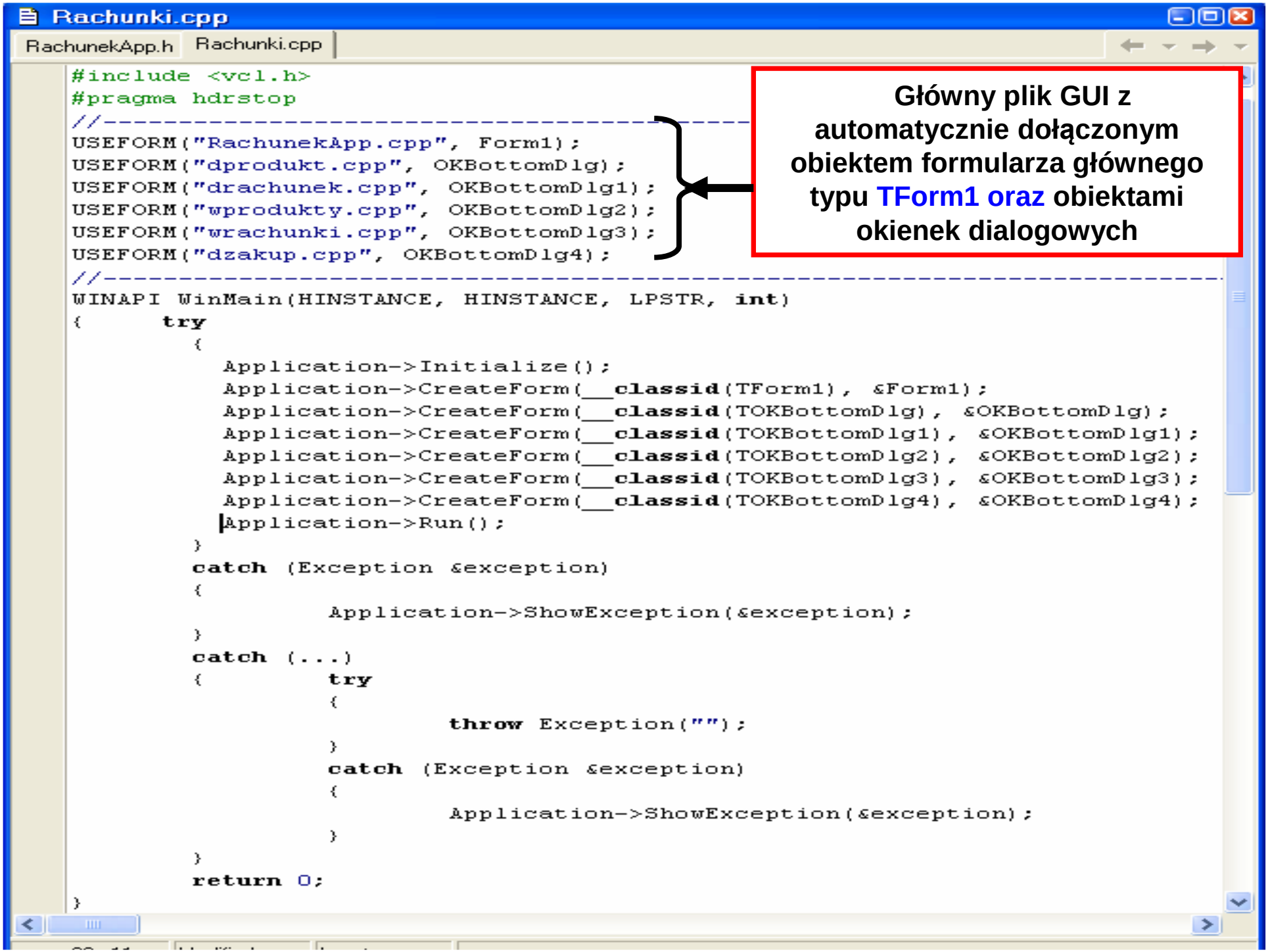
RachunekApp.cpp | wprodukty.cpp | dprodukt.cpp

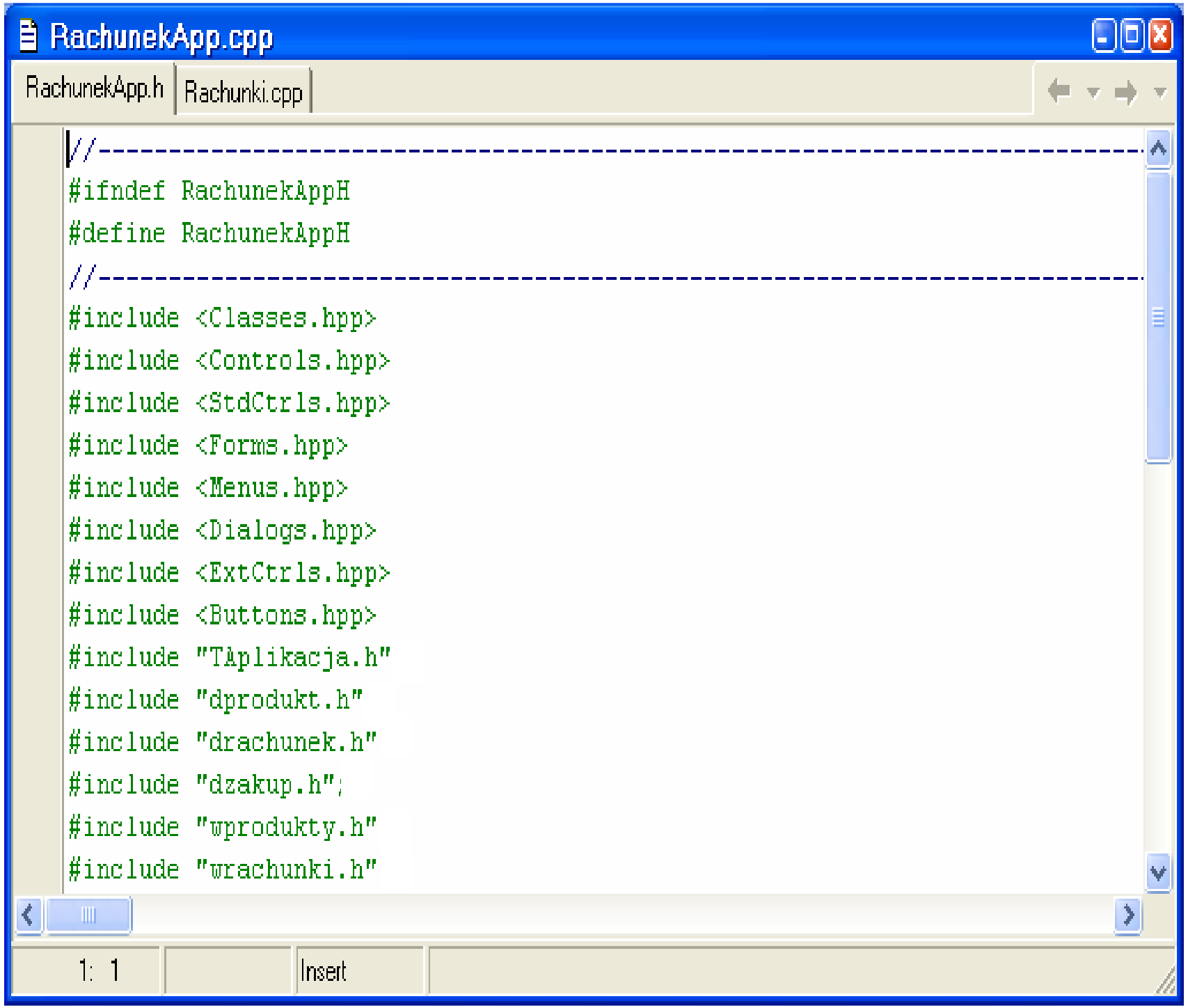
```
//ręcznie
 string __fastcall TOKBottomDlg2::cena_netto4(string a, string b, string c)
 { float x,y,z; char lan[10];
   x=atof(a.c_str()); y=atof(b.c_str()); z=atof(c.c_str());//cena netto
   x=(x*100)/(100+y-z); //na podstawie ceny brutto x, podatku y i promocji z
   return gcvt(x,3,lan);
 }

string __fastcall TOKBottomDlg2::cena_netto3(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 promocji y
   return gcvt(x,3,lan);
 }

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);
 }
```

23: 77 | Modified | Insert

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

1. Udostępnianie wszystkich prywatnych atrybutów do prezentacji, wprowadzenie standardu nazewnictwa plików – nazwy plików aplikacji poprzedzone literą T
2. Budowa głównego formularza GUI
3. Budowa okienek dialogowych do wprowadzania danych
4. Budowa okienek dialogowych do wyświetlania danych
5. Aplikacja do sporządzania rachunków - uzupełnienie


Rachunki.cpp
RachunekApp.h
Rachunki.cpp
#include <vcl.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


RachunekApp.cpp
RachunekApp.h
Rachunki.cpp
//---
#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 "TAplikacja.h"
#include "dprodukt.h"
#include "drachunek.h"
#include "dzakup.h";
#include "wprodukty.h"
#include "wrachunki.h"
1: 1
Insert

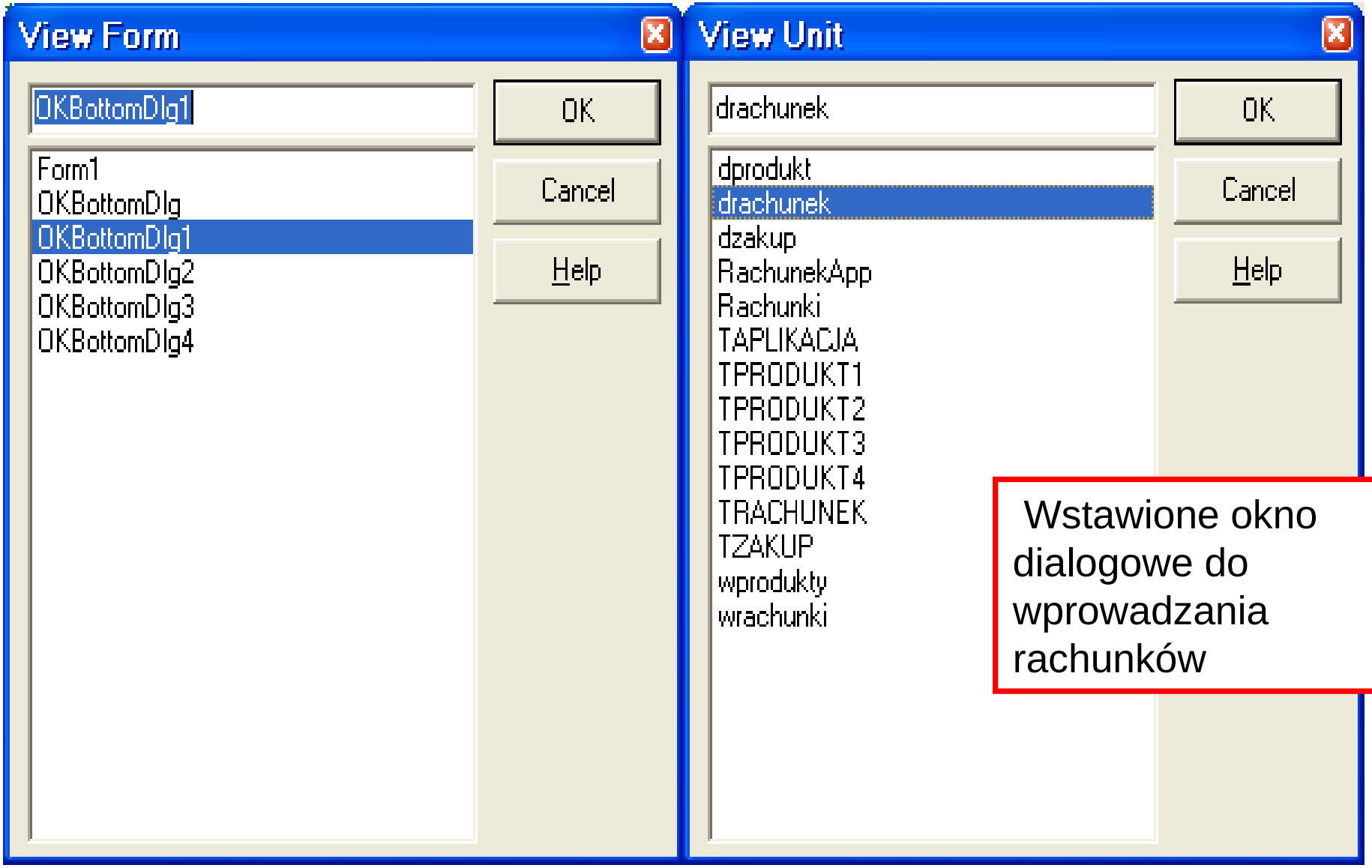
RachunekApp.cpp

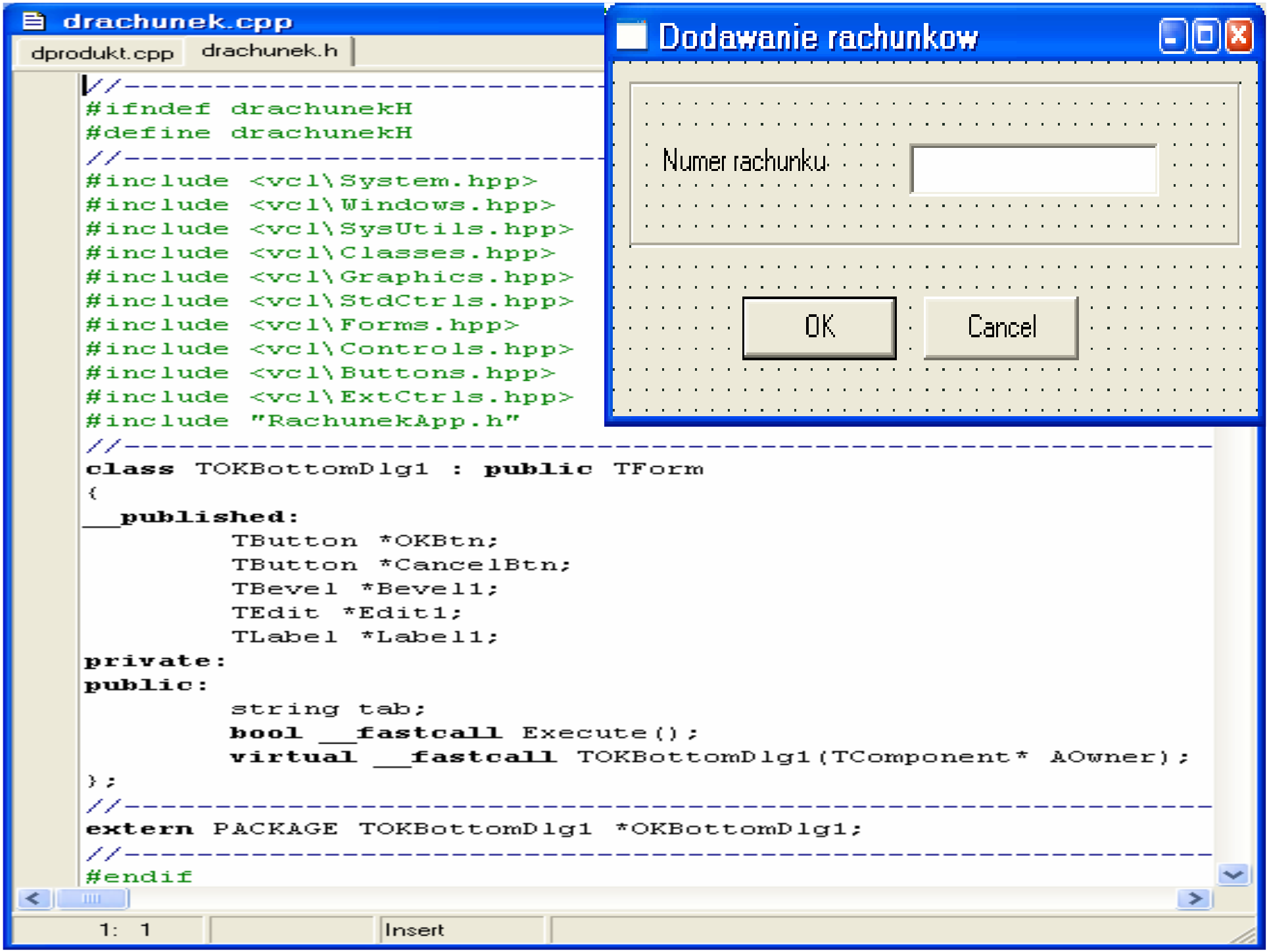
RachunekApp.h | Rachunki.cpp

```
class TForm1 : public TForm
{
__published:    // IDE-managed Components
        TMainMenu *MainMenu1;
        TMenuItem *Pliki1;
        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);
        void __fastcall Dodajrachunek1Click(TObject *Sender);
        void __fastcall Wyswietlrachunek2Click(TObject *Sender);
        void __fastcall Dodajzakup1Click(TObject *Sender);
        void __fastcall Oprogramie1Click(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
//---------------------------------------------------------------------------
#endif
```

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

1: 1 | Insert


View Form
OKBottomDlg1
OK
Cancel
Help
Form1
OKBottomDlg
OKBottomDlg1
OKBottomDlg2
OKBottomDlg3
OKBottomDlg4
View Unit
drachunek
OK
Cancel
Help
dprodukt
drachunek
dzakup
RachunekApp
Rachunki
TAPLIKACJA
TPRODUKT1
TPRODUKT2
TPRODUKT3
TPRODUKT4
TRACHUNEK
TZAKUP
wprodukty
wrachunki
Wstawione okno dialogowe do wprowadzania rachunków




```
//---------------------------
#ifndef drachunekH
#define drachunekH
//---------------------------
#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 TOKBottomDlg1 : public TForm
{
__published:
        TButton *OKBtn;
        TButton *CancelBtn;
        TBevel *Bevel1;
        TEdit *Edit1;
        TLabel *Label1;
private:
public:
        string tab;
        bool __fastcall Execute();
        virtual __fastcall TOKBottomDlg1(TComponent* AOwner);
};
//-------------------------------------------------------------------
extern PACKAGE TOKBottomDlg1 *OKBottomDlg1;
//-------------------------------------------------------------------
#endif
```

drachunek.cpp

dprodukt.cpp | drachunek.cpp

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

```
#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;
}
```

20: 65 | Modified | Insert

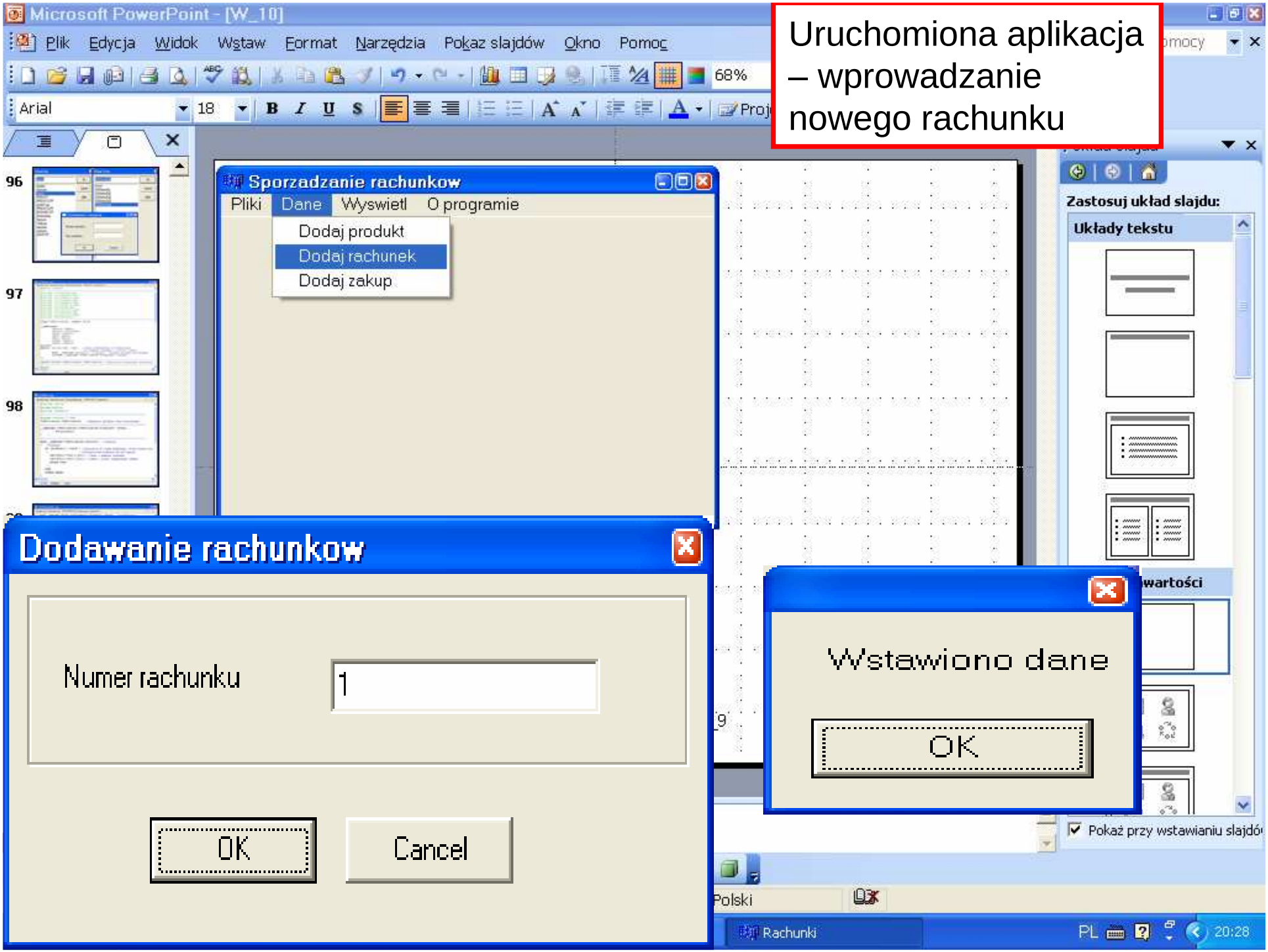
RachunekApp.cpp

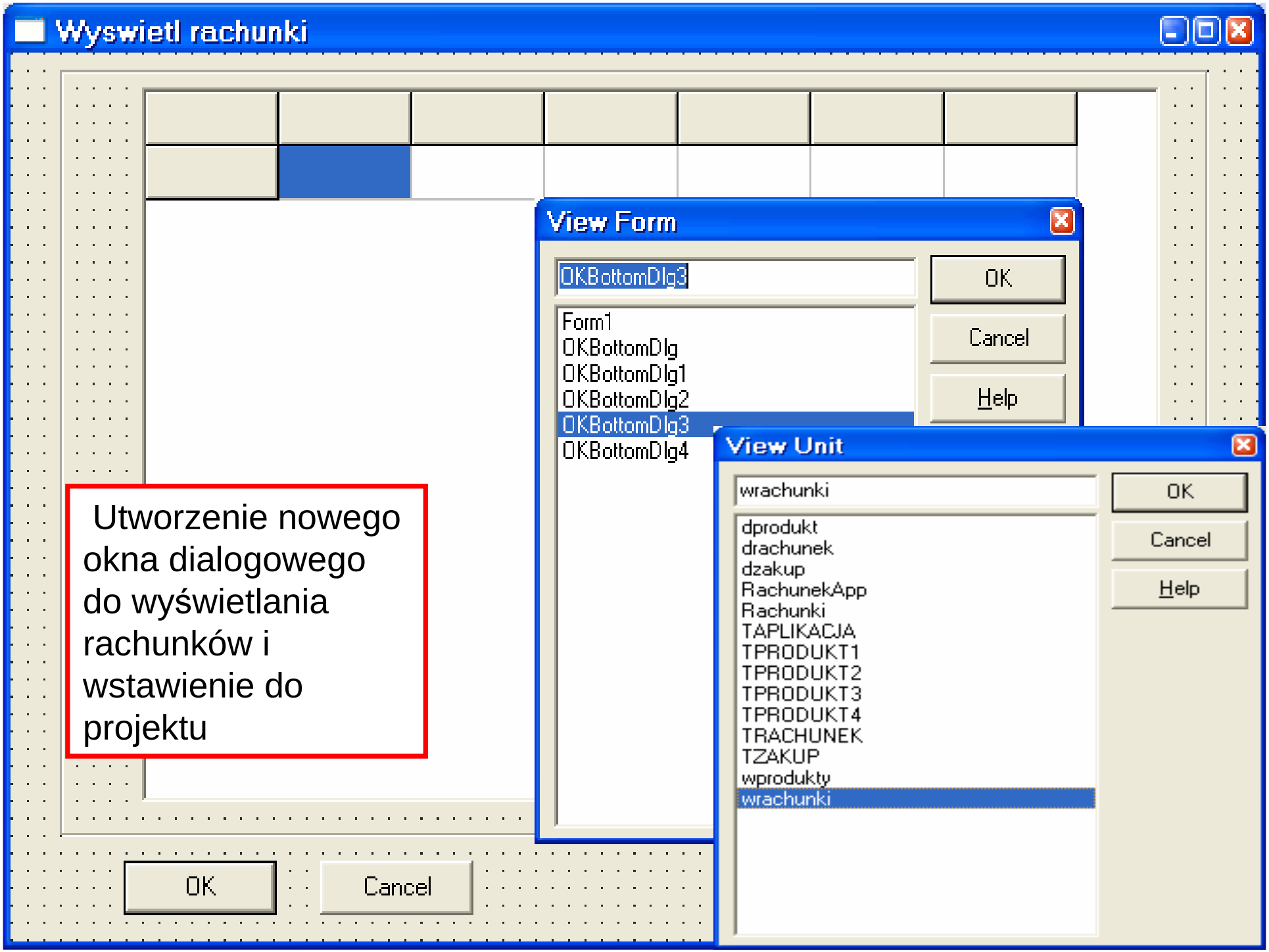
dprodukt.cpp | drachunek.cpp | RachunekApp.cpp

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
Microsoft PowerPoint - [W_10]
Plik Edycja Widok Wstaw Format Narzędzia Pokaz slajdów Okno Pomoc
Sporzadzanie rachunkow
Pliki Dane Wyswietl O programie
Dodaj produkt
Dodaj rachunek
Dodaj zakup
Zastosuj układ slajdu:
Układy tekstu
Dodawanie rachunkow
Numer rachunku
1
OK
Cancel
Wstawiono dane
OK
Rachunki


Wyswietl rachunki
View Form
OKBottomDlg3
OK
Cancel
Help
Form1
OKBottomDlg
OKBottomDlg1
OKBottomDlg2
OKBottomDlg3
OKBottomDlg4
View Unit
wrachunki
OK
Cancel
Help
dprodukt
drachunek
dzakup
RachunekApp
Rachunki
TAPLIKACJA
TPRODUKT1
TPRODUKT2
TPRODUKT3
TPRODUKT4
TRACHUNEK
TZAKUP
wprodukty
wrachunki
Utworzenie nowego okna dialogowego do wyświetlania rachunków i wstawienie do projektu
OK
Cancel

wrachunki.cpp

dprodukt.cpp | drachunek.cpp | RachunekApp.cpp | wrachunki.h

```
#ifndef wrachunkiH
#define wrachunkiH
//---------------------------------------------------------------------------
#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 <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 - obsluga okienka dialogowego
        virtual __fastcall TOKBottomDlg3(TComponent* AOwner);
};
```

Plik nagłówkowy okna dialogowego do wyświetlania rachunków

25: 62 | Insert

wrachunki.cpp

dprodukt.cpp | drachunek.cpp | RachunekApp.cpp | wrachunki.cpp

```
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 wykorystanych
    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
```

7: 25 | Insert

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

wrachunki.cpp

dprodukt.cpp | drachunek.cpp | RachunekApp.cpp | wrachunki.cpp

```
    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);
       pom=produkt->Podaj_promocje();
       if (pom==-1)    //jesli brak promocji wyswietla sie pusty lancuch
         StringGrid1->Cells[4][i]="";
       else
         StringGrid1->Cells[4][i]=gcvt(produkt->Podaj_promocje(),3,tab);
       StringGrid1->Cells[5][i]=itoa(zakup->Podaj_ilosc(),tab,10);
       StringGrid1->Cells[6][i]=gcvt(zakup->Podaj_wartosc(),3,tab);
    }
```

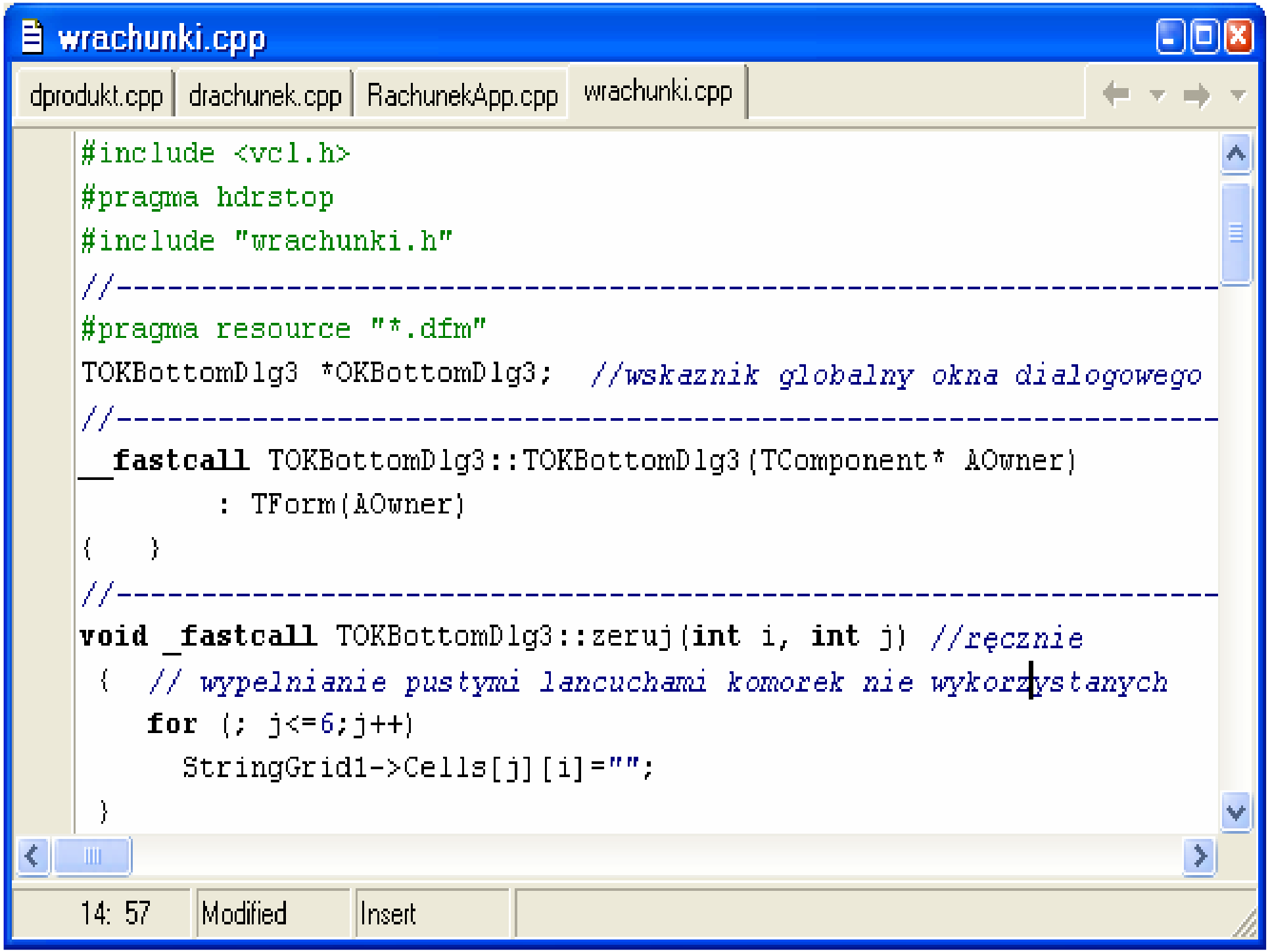
7: 25 | Insert

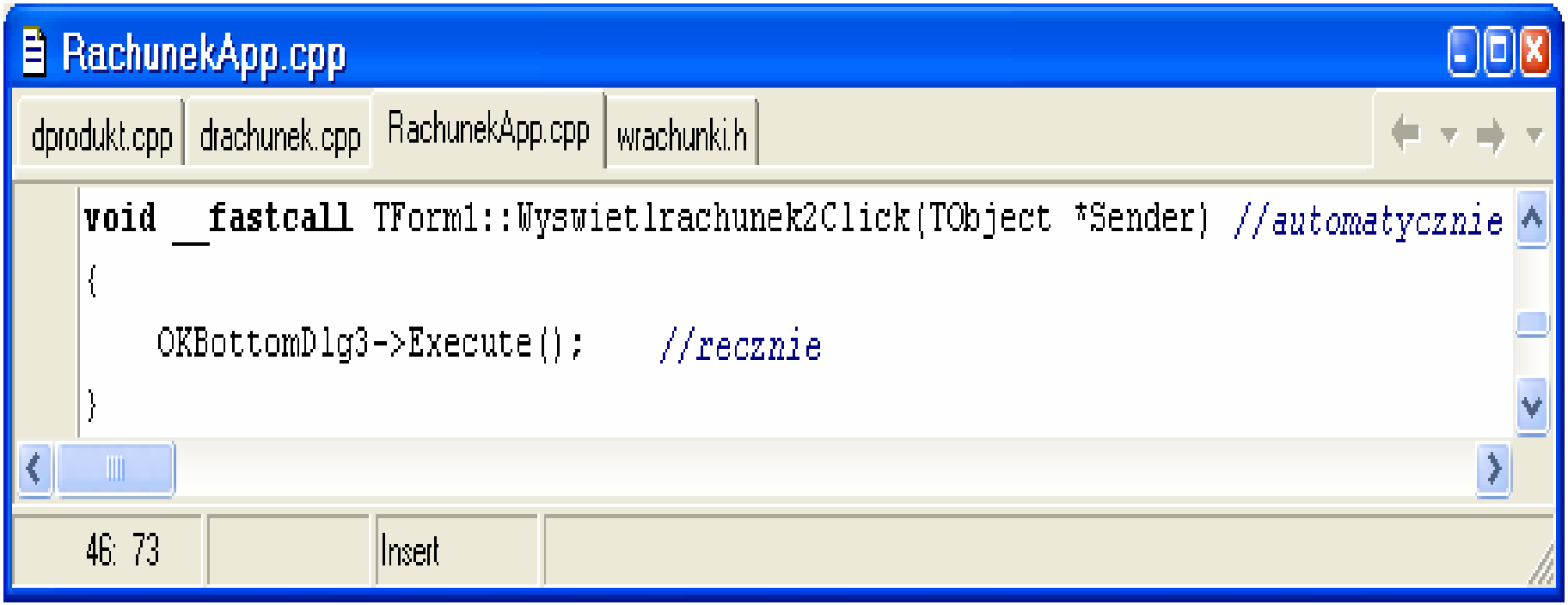
wrachunki.cpp

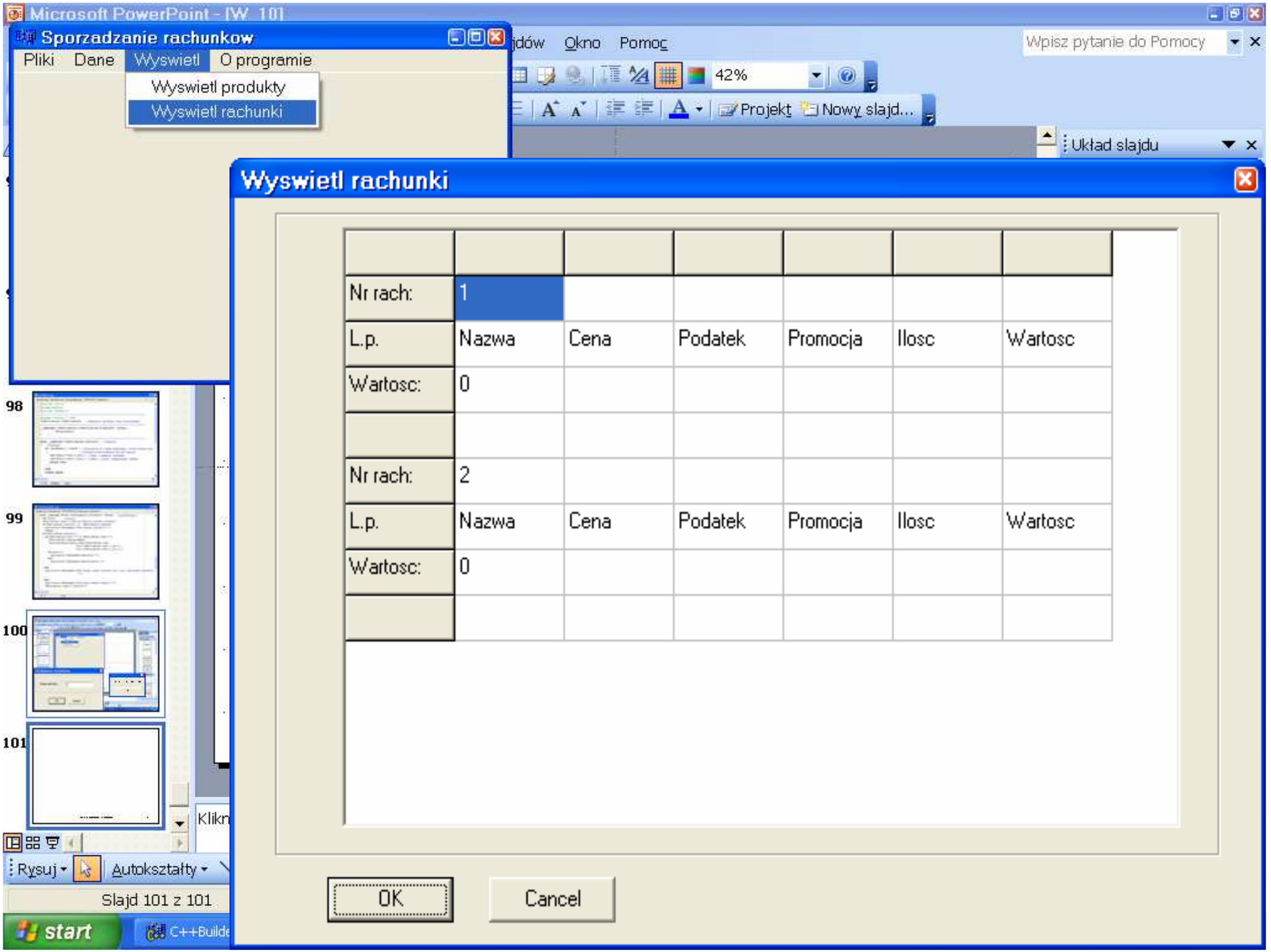
dprodukt.cpp | drachunek.cpp | RachunekApp.cpp | wrachunki.cpp

```
    StringGrid1->RowCount=i+1;  //dodanie nowego rowka
    StringGrid1->Cells[0][i]="Wartosc: ";
    StringGrid1->Cells[1][i]= gcvt(rachunek->Podaj_wartosc(),3,tab);
    zeruj(i,2);  //wypełnianie pustymi lancuchami komorek nie wykorystanych
    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, ktory konczy sie nacisnieciem
return true;    //klawisza OK lub Cancel;
}
```

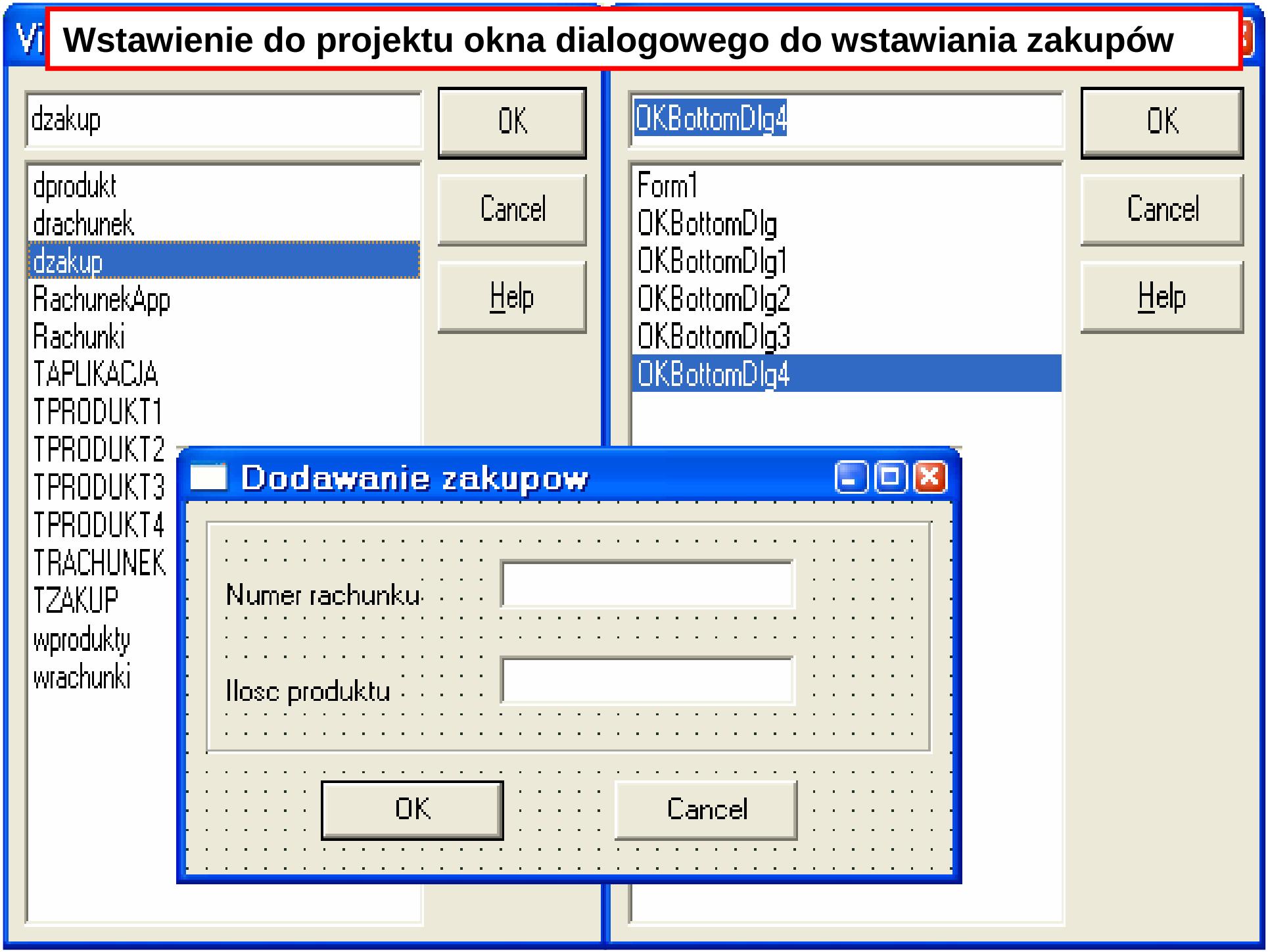
83: 18 | Modified | Insert


wrachunki.cpp
dprodukt.cpp
drachunek.cpp
RachunekApp.cpp
wrachunki.cpp
#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]="";
 }
14: 57
Modified
Insert


RachunekApp.cpp
dprodukt.cpp
drachunek.cpp
RachunekApp.cpp
wrachunki.h
void __fastcall TForm1::Wyswietlrachunek2Click(TObject *Sender) //automatycznie
{
OKBottomDlg3->Execute(); //recznie
}
46: 73
Insert


Microsoft PowerPoint - [W_10]
Sporzadzanie rachunkow
Pliki
Dane
Wyswietl
O programie
Wyswietl produkty
Wyswietl rachunki
Okno
Pomoc
Wpisz pytanie do Pomocy
42%
Projekt
Nowy slajd...
Układ slajdu
Wyswietl rachunki
Nr rach: 1
L.p. Nazwa Cena Podatek Promocja Ilosc Wartosc
Wartosc: 0
Nr rach: 2
L.p. Nazwa Cena Podatek Promocja Ilosc Wartosc
Wartosc: 0
OK
Cancel
98
99
100
101
Rysuj
Autokształty
Slajd 101 z 101
start

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
```

1: 1 | Insert

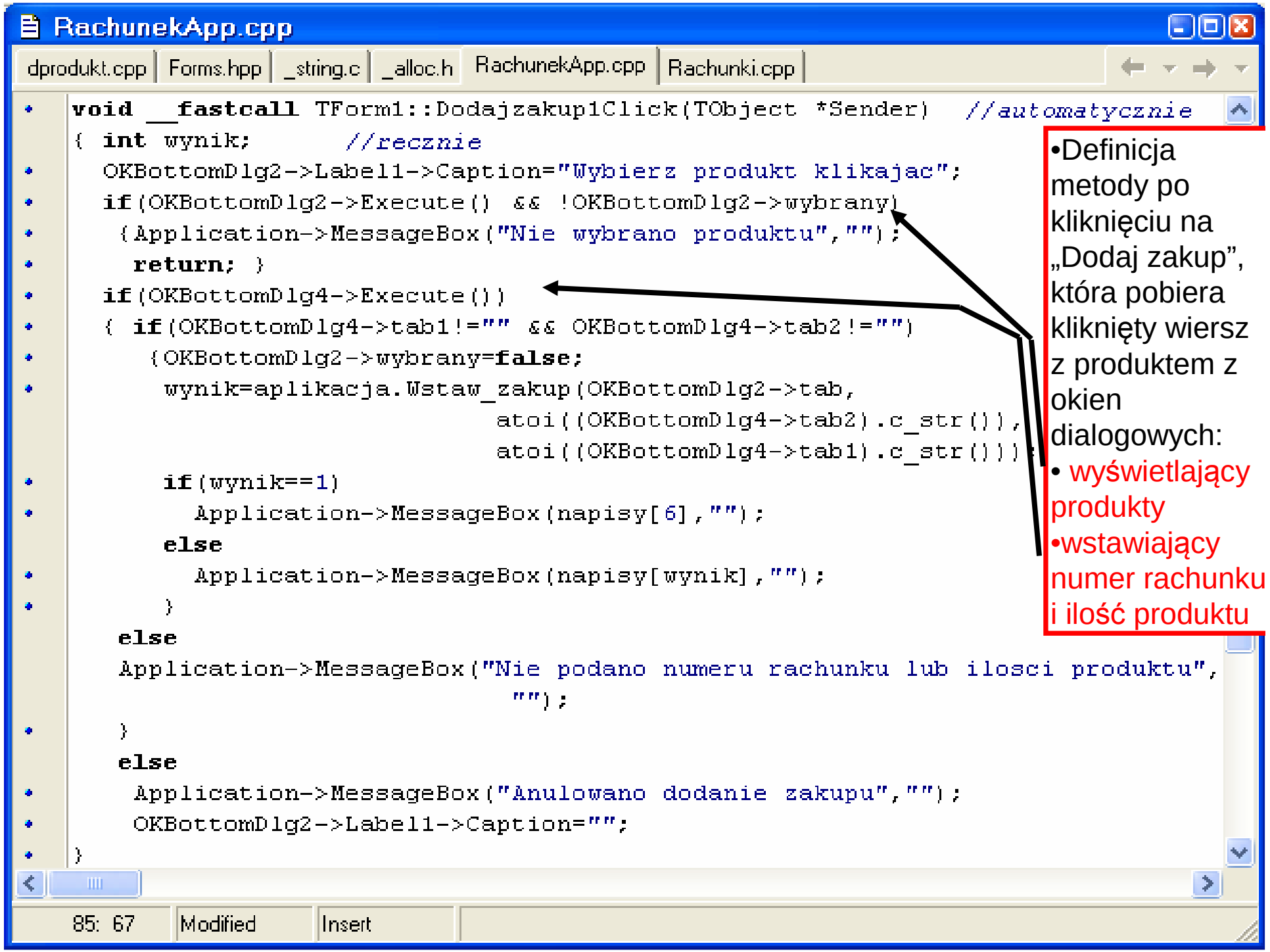
dzakup.cpp

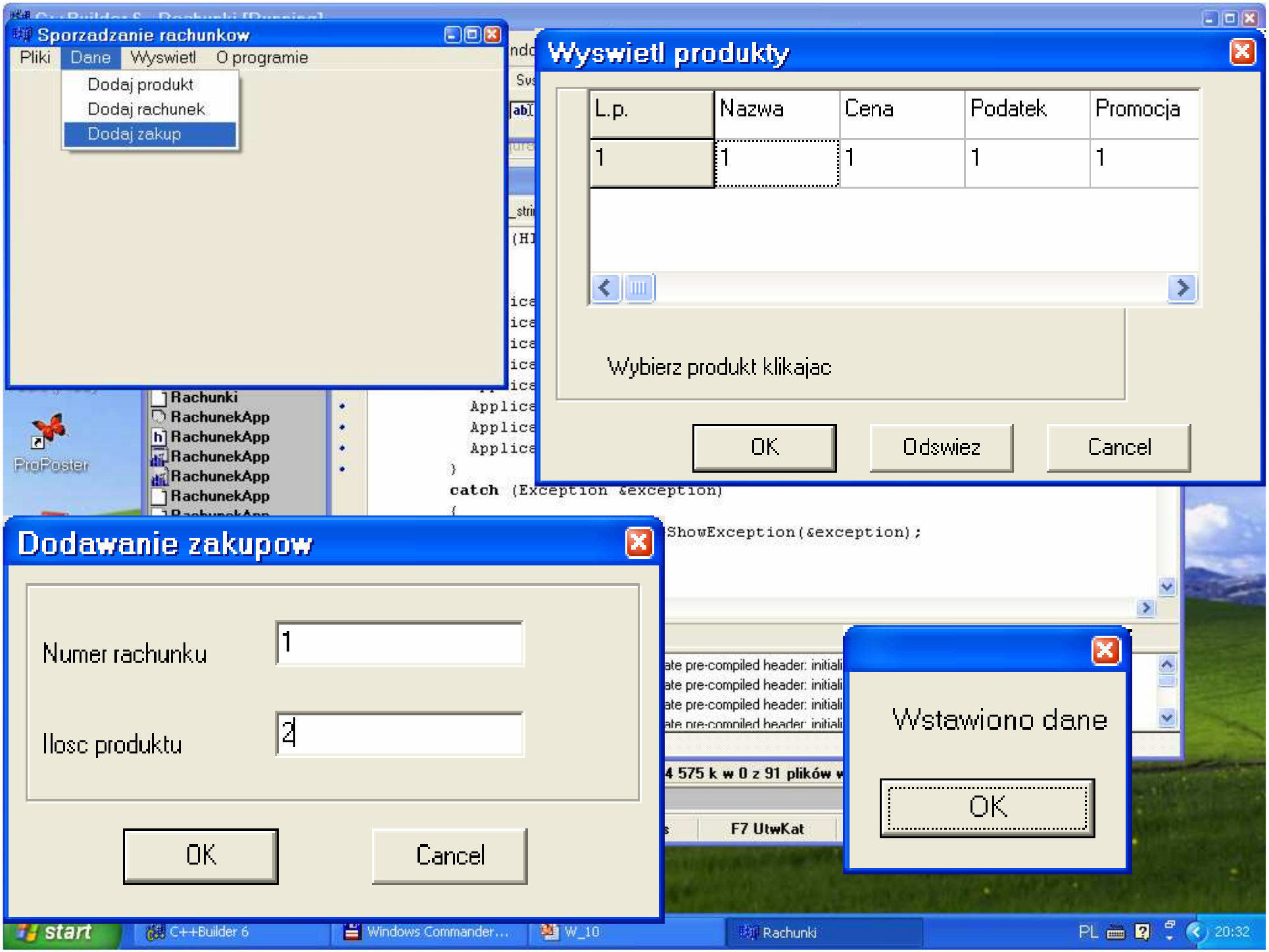
dprodukt.cpp | drachunek.cpp | RachunekApp.cpp | dzakup.cpp | wrachunki.h

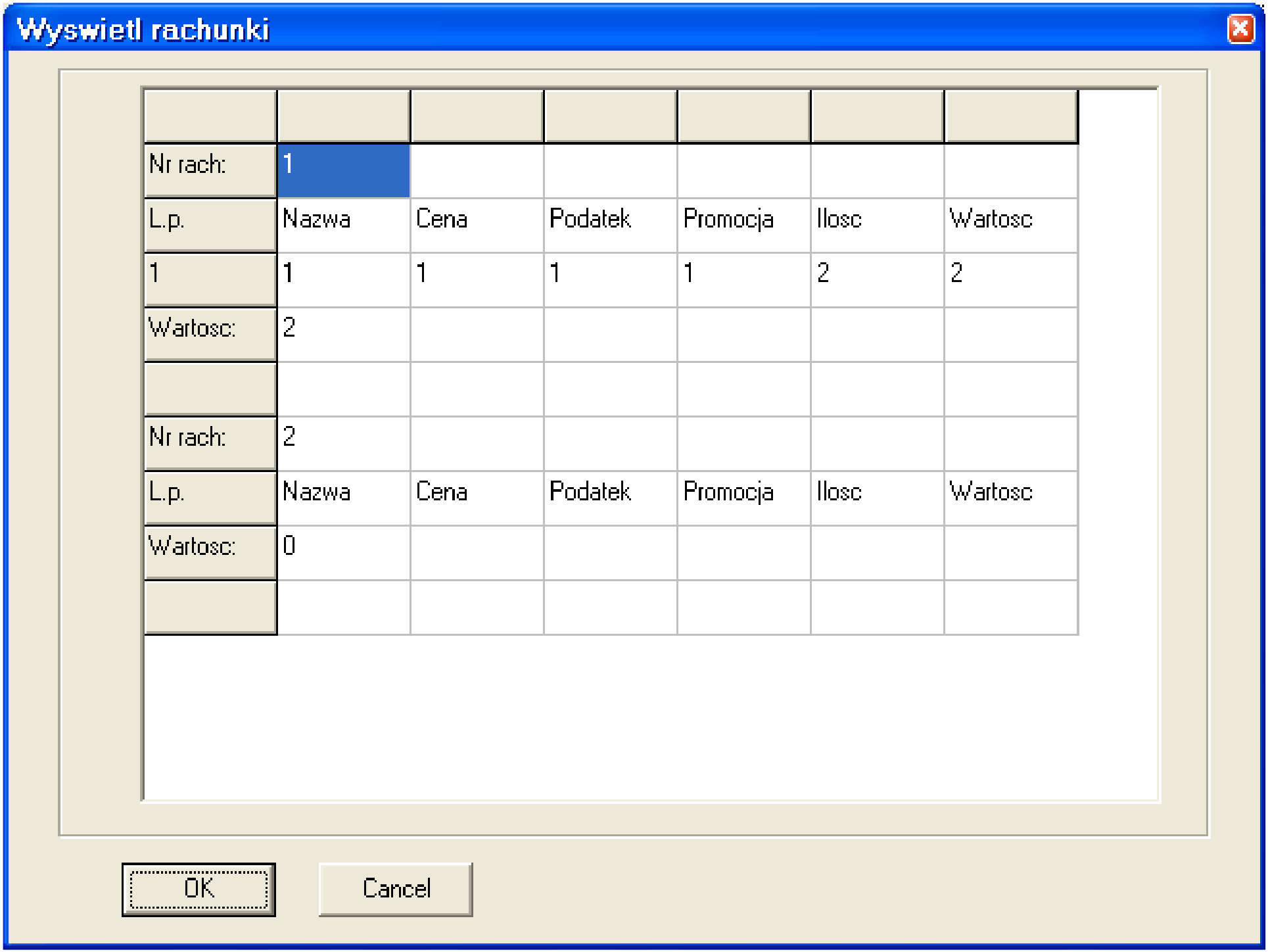
Definicja metody Execute() pobierającej numer rachunku oraz ilość zakupionego produktu

```
#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, ktory 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;
}
```

22: 66 | Modified | Insert


RachunekApp.cpp
dprodukt.cpp
Forms.hpp
_string.c
_alloc.h
RachunekApp.cpp
Rachunki.cpp
void __fastcall TForm1::Dodajzakup1Click(TObject *Sender) //automatycznie
{ 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="";
}
85: 67
Modified
Insert
•Definicja metody po kliknięciu na „Dodaj zakup”, która pobiera kliknięty wiersz z produktem z okien dialogowych:
• wyświetlający produkty
•wstawiający numer rachunku i ilość produktu


Sporzadzanie rachunkow
Pliki
Dane
Wyswietl
O programie
Dodaj produkt
Dodaj rachunek
Dodaj zakup
Wyswietl produkty
L.p.
Nazwa
Cena
Podatek
Promocja
1
1
1
1
1
Wybierz produkt klikajac
OK
Odswiez
Cancel
Dodawanie zakupow
Numer rachunku
1
Ilosc produktu
2
OK
Cancel
Wstawiono dane
OK
Rachunki
RachunekApp
RachunekApp
RachunekApp
RachunekApp
RachunekApp
ProPoster
catch (Exception &exception)
ShowException(&exception);
F7 UtwKat
start
C++Builder 6
Windows Commander...
W_10
Rachunki
PL
20:32


Wyswietl rachunki
Nr rach: 1
L.p. Nazwa Cena Podatek Promocja Ilosc Wartosc
1 1 1 1 1 2 2
Wartosc: 2
Nr rach: 2
L.p. Nazwa Cena Podatek Promocja Ilosc Wartosc
Wartosc: 0
OK
Cancel