

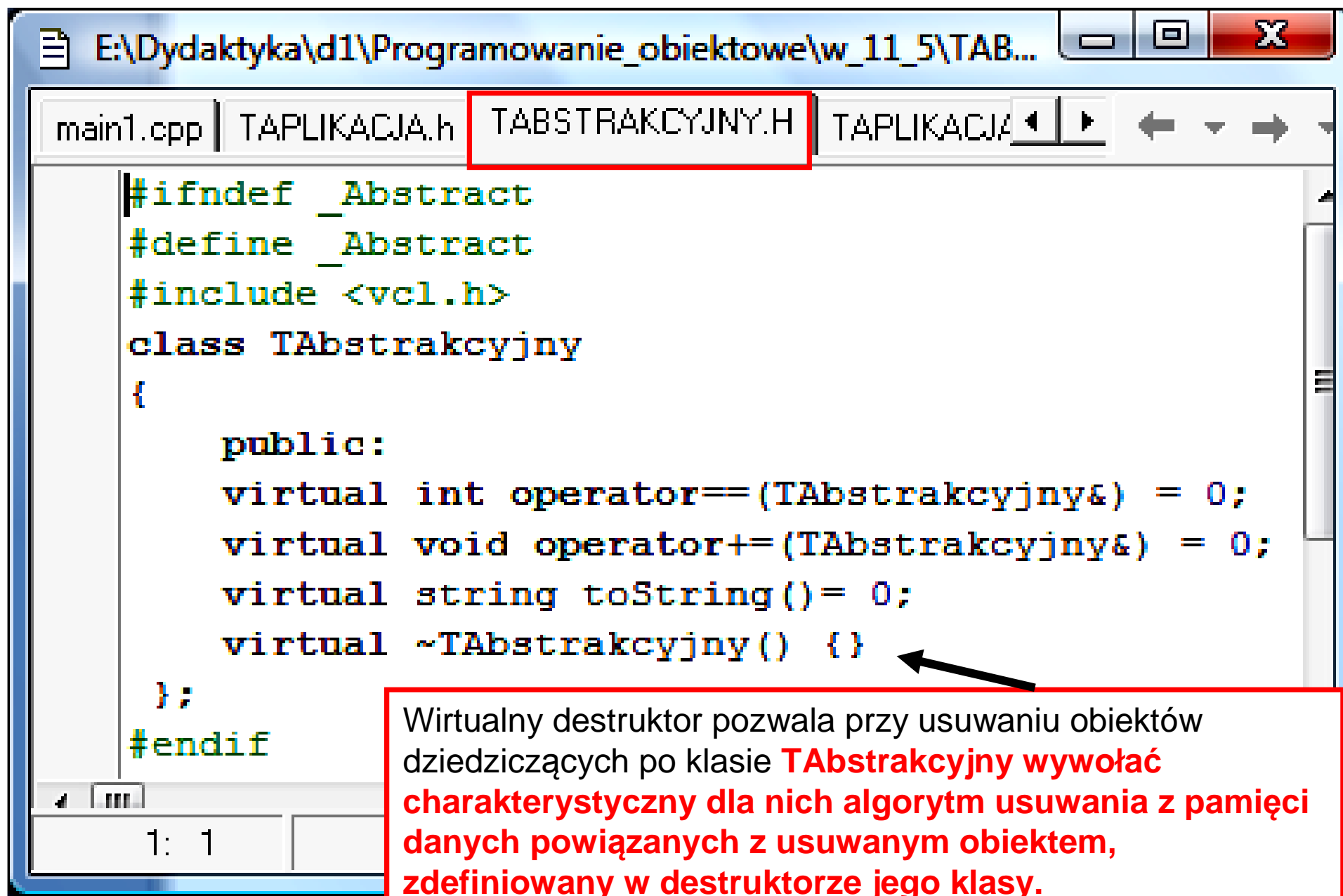
Budowa aplikacji z graficznym interfejsem użytkownika - GUI (Graphic User Interface)

1. **Wzorzec fasady – klasa *TAplikacja***
hermetyzująca dostęp do kolekcji produktów
i rachunków. **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**

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

1. **Wzorzec fasady – klasa *TAplikacja***
hermetyzująca dostęp do kolekcji produktów i
rachunków

Klasa abstrakcyjna TAbstrakcyjny – przypomnienie definicji



```
#ifndef _Abstract
#define _Abstract
#include <vcl.h>
class TAbstrakcyjny
{
public:
virtual int operator==(TAbstrakcyjny&) = 0;
virtual void operator+=(TAbstrakcyjny&) = 0;
virtual string toString() = 0;
virtual ~TAbstrakcyjny() {}
};
#endif
```

Wirtualny destruktor pozwala przy usuwaniu obiektów dziedziczących po klasie **TAbstrakcyjny** wywołać charakterystyczny dla nich algorytm usuwania z pamięci danych powiązanych z usuwanym obiektem, zdefiniowany w destruktorze jego klasy.

```

#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;}
    virtual float Podaj_cene();
    virtual float Podaj_podatek() { return -1; }
    void operator+=(TABstrakcyjny&){}
    int operator==(TABstrakcyjny&);
    string toString();
    friend ostream& operator<<(ostream&, TProdukt1&);
};
#endif

```

Metody wirtualne przydatne do tworzenia pozycji rachunku (obiekty *TZakup*) z różnymi produktami i jednolitym sposobie używania tych obiektów

dla GUI

Metody wirtualne przydatne do tworzenia kolekcji zawierającej różne typy obiektów – elementów kolekcji

```
#include "TPRODUKT1.H"
TProdukt1::TProdukt1(string nazwa_,float cena_)
{ nazwa=nazwa_;
  cena=cena_; }
TProdukt1::TProdukt1(TProdukt1& p)
{ nazwa=p.nazwa;
  cena=p.cena; }
TProdukt1::~~TProdukt1()
{ }
float TProdukt1::Podaj_cene()
{ return cena; }
int TProdukt1::operator==(TAbstrakcyjny& p_)
{ TProdukt1& p = (TProdukt1&)p_;
  float a= Podaj_cene(), b= p.Podaj_cene();
  return nazwa==p.nazwa &&a==b&& Podaj_podatek()==p.Podaj_podatek(); }
string TProdukt1::toString()
{ char tab[10];
  return " Nazwa: "+nazwa+", Cena detaliczna: "+gcvt(Podaj_cene(),3,tab); }
ostream& operator<<(ostream& wy, TProdukt1& p)
{ return wy<<p.toString()<<endl; }
```

```
#ifndef _Produkt2
#define _Produkt2
#include "TPRODUKT1.h"
class TProdukt2: 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
```

```
#include "TPRODUKT2.H"
TProdukt2::TProdukt2(string nazwa_, float cena_, float podatek_):
    TProdukt1(nazwa_, cena_), podatek(podatek_)
{ }
TProdukt2::TProdukt2(TProdukt2& p):TProdukt1(p), podatek(p.podatek)
{ }
TProdukt2::~~TProdukt2()
{ }
float TProdukt2::Podaj_podatek()
{ return podatek; }
float TProdukt2::Czesc_brutto()
{ return cena*podatek/100; }
float TProdukt2::Podaj_cene()
{ return TProdukt1::Podaj_cene() + Czesc_brutto();}
string TProdukt2::toString()
{ char t[10];
  return TProdukt1::toString()+" , Podatek: "+itoa(Podaj_podatek(),t,10);}
ostream& operator<<(ostream& wy, TProdukt2& p)
{ wy<<"Operator TProdukt2"<<endl;
  return wy<<p.toString()<<endl; }
```

TRACHUNEK.CPP | TRACHUNEK.h | TZAKUP.cpp | TZAKUP.h

```
#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;}
    void operator+=(TAbstrakcyjny&);
    int operator==(TAbstrakcyjny&);
    string toString();
    friend ostream& operator<<(ostream&, TZakup&);
};
#endif
```

dla GUI

Metody wirtualne przydatne do tworzenia kolekcji zawierającej różne typy obiektów – elementów kolekcji


```
#include "TZAKUP.H"
TZakup::TZakup(TProdukt1*produkt_,float ilosc_)
{ produkt=produkt_;
  ilosc=ilosc_;}
TZakup::TZakup(TZakup& z)
{ produkt=z.produkt;
  ilosc=z.ilosc; }
TZakup::~TZakup()
{ }
float TZakup::Podaj_wartosc()
{ return ilosc*produkt->Podaj_cene(); }
float TZakup::Podaj_ilosc()
{ return ilosc; }
void TZakup::operator+=(TAbstrakcyjny& z)
{ ilosc+=((TZakup&)z).ilosc; }
int TZakup::operator==(TAbstrakcyjny& zakup_)
{ TZakup& zakup = (TZakup&)(zakup_);
  return *produkt==*zakup.produkt; }
string TZakup::toString()
{ char t[10], p[10];
  return produkt->toString()+"\n"
         " Ilosc produktu: "+itoa(ilosc,p,10)+
         ", Wartosc zakupu: "+gcvt(Podaj_wartosc(),3,t); }
ostream& operator<<(ostream& wy, TZakup& zakup)
{ return wy<<zakup.toString()<<endl; }
```

Powtórzenie informacji o klasie *TRachunek*

TRACHUNEK.CPP

TRACHUNEK.h

TZAKUP.cpp

TZAKUP.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()
    int Podaj_numer()
    void operator=(TRachunek&);
    float Podaj_wartosc();
    int Dodaj_zakup(TZakup*);
    void operator+=(TAbstrakcyjny&) {}
    int operator==(TAbstrakcyjny&);
    string toString();
    friend ostream& operator<<(ostream&, TRachunek&);
};
#endif
```

dla
GUI

Metody wirtualne przydatne do wstawiania do kolekcji zawierającej różne typy obiektów – elementów kolekcji

```
TRACHUNEK.CPP
TRACHUNEK.CPP | TRACHUNEK.h | TZAKUP.cpp | TZAKUP.h |
#include "TRACHUNEK.h"
TRachunek::TRachunek(int nr)
{ numer=nr;}
void TRachunek::kopia(TRachunek& r)
{ TZakup* z,*pom;
  numer= r.numer;
  zakupy=r.zakupy;
  r.zakupy.Zeruj();
  zakupy.Zeruj();
  while (!r.zakupy.Koniec())
  { z=r.zakupy.Podaj_nast();
    pom =new TZakup(*z);
    zakupy.Podaj_nast()=pom; }
}
TRachunek::TRachunek(TRachunek& r)
{ kopia(r); }

TRachunek::~~TRachunek()
{ zakupy.Usun_kolekcje();}

void TRachunek::operator=(TRachunek& r)
{ if (this==&r) return;
  zakupy.Usun_kolekcje();
  kopia(r);
}
int TRachunek::Dodaj_zakup(TZakup* wstawianyzakup)
{ return zakupy.Wstaw(wstawianyzakup); }
```

26: 1 Modified Insert

```
float TRachunek::Podaj_wartosc()
{ float suma=0;
  zakupy.Zeruj();
  while(!zakupy.Koniec())
  { TZakup* nastepnyzakup=zakupy.Podaj_nast();
    suma+=nastepnyzakup->Podaj_wartosc();
  }
  return suma;
}

int TRachunek::operator==(TAbstrakcyjny& r)
{ int bStatus = 1;
  if ( numer!= ((TRachunek&)r).numer ) bStatus = 0;
  return bStatus; }

string TRachunek::toString()
{ char p[10];
  string napis= " Rachunek : ";
  napis+=itoa(numer,p,10);
  napis+="\n";
  napis+=zakupy.toString();
  napis+=" Wartosc rachunku: ";
  napis+=gcvt(Podaj_wartosc(),3,p);
  napis+="\n";
  return napis; }

ostream& operator<<(ostream& wy, TRachunek& r)
{ return wy<<r.toString(); }
```

Powtórzenie informacji o klasie *TKol2*

```
TKOL2.h
TRACHUNEK.CPP | TKOL2.h | TRACHUNEK.h | TZAKUP.cpp | TZAKUP.h
#ifndef _TKOL2
#define _TKOL2
#include <string.h>
#include "TABSTRAKCYJNY.h"
const int N=5;
template <class T>
class TKol2
{protected:
    T* kolekcja[N];
    int ile;
    int biezacy;
public:
    TKol2(int a=0)
        {ile = a;
         biezacy = 0; }
    ~TKol2()          { }
    int Pusta()       { return ile==0;}
    void Zeruj()      { biezacy=0;}
    int Koniec()      { return biezacy==ile; }
    T*& Podaj_nast()  { return kolekcja[biezacy++]; }
    void Usun_kolekcje();
    int Wstaw(T* dane);
    T* Podaj(T* dane);
    string toString();
    friend ostream& operator<<(ostream& wy, TKol2<T>& kol)
        { return wy<<kol.toString()<<endl; }
};
#endif
```

```
E:\Dydaktyka\d1\Programowanie_obiektowe\w_11_5\TKOL2.h
TABSTRAKCYJNY.H | TPRODUKT1.h | TKOL2.h | TZAKUP.cpp | TZA
template <class T> void TKol2<T>::Usun_kolekcje()
{
    for (int i=0; i<ile;i++)
        delete kolekcja[i];
    ile=0;
}
template <class T> int TKol2<T>::Wstaw(T* dane)
{
    for (int i=0;i<ile;i++)
        if(*dane == *kolekcja[i])
            { *(kolekcja[i])+=*dane;
              delete dane; return 1;}
    if(ile==N)
        { delete dane; return 0; }
    kolekcja[ile++]=dane;
    return 2;
}
template <class T> T* TKol2<T>::Podaj(T* dane)
{ T* pom=NULL;
  for (int i=0;i<ile;i++)
      if(*dane == *kolekcja[i])
          { pom= kolekcja[i];
            break; }
  return pom;
}
template <class T> string TKol2<T>::toString()
{ string s;
  for (int i=0; i<ile;i++)
      s+=kolekcja[i]->toString()+"\n";
  return s; }
43: 8 | Insert
Build
```

```
#ifndef _Aplikacja
#define _Aplikacja
#include "TPRODUKT2.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
```

Metody klasy **TAplikacja**
hermetyzują operacje na kolekcji
produktów i kolekcji zakupów

```
TAPLIKACJA.h | TABSTRAKCYJNY.H | TAPLIKACJA.cpp | TPRODUKT1.CPP | TPRODUKT1.h | TPRODUKT2.CPP
#include "TAPLIKACJA.H"
int TApplikacja::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;
}
62: 33 | Insert
```



```
int TApplikacja::Wstaw_produkt(string* wstawianyprodukt)
{ TProdukt1* pom= Wykonaj_produkt(wstawianyprodukt);
  int wynik=3;
  if (pom!=NULL)
    wynik=produkty.Wstaw(pom);
  return wynik;
}

int TApplikacja::Wstaw_rachunek(int nr)
{ TRachunek* r=new TRachunek(nr);
  int wynik=3;
  if(r!=NULL)
    wynik=rachunki.Wstaw(r);
  return wynik;
}

TKol2<TProdukt1>& TApplikacja::Podaj_produkty()
{ return produkty; }

TKol2<TRachunek>& TApplikacja::Podaj_rachunki()
{ return rachunki; }

TRachunek* TApplikacja::Podaj_rachunek(int nr)
{ TRachunek r(nr);
  return rachunki.Podaj(&r);
}
```

The image shows a screenshot of a C++ IDE window titled "TAPLIKACJA.cpp". The window contains a code editor with the following C++ code:

```
TProdukt1* TApplikacja::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()));
    return produkt;
}
```

The code editor also shows a status bar at the bottom with the text "62: 33" and "Insert".

main1.cpp

main1.cpp

TAPLIKACJA.h

TABSTRAKCYJNY.H

TAPLIKAC

```
#include "TAPLIKACJA.h"
void main()
{
    TApplikacja aplikacja;
    string tablica1[3]= {"1", "zeszyt", "2.0"};
    string tablica2[4]= {"2", "pioro", "4.80", "20"};
    string tablica3[4]= {"2", "pioro", "4.80", "10"};
    aplikacja.Wstaw_produkt(tablica1);
    aplikacja.Wstaw_produkt(tablica2);
    aplikacja.Wstaw_produkt(tablica2);
    aplikacja.Wstaw_produkt(tablica3);

    cout<<aplikacja.Podaj_produkty();

    aplikacja.Wstaw_rachunek(1);
    aplikacja.Wstaw_rachunek(1);
    aplikacja.Wstaw_rachunek(2);
    cout<<aplikacja.Podaj_rachunki();
}
```

1: 1

Insert

```
main1.cpp | TAPLIKACJA.h | TABSTRAKCYJN...  
  
    aplikacja.Wstaw_zakup(tablica1,1,1);  
    aplikacja.Wstaw_zakup(tablica1,2,1);  
    aplikacja.Wstaw_zakup(tablica2,5,1);  
    aplikacja.Wstaw_zakup(tablica2,6,1);  
    aplikacja.Wstaw_zakup(tablica2,5,1);  
    aplikacja.Wstaw_zakup(tablica3,6,1);  
  
    aplikacja.Wstaw_zakup(tablica1,1,2);  
    aplikacja.Wstaw_zakup(tablica2,2,2);  
    aplikacja.Wstaw_zakup(tablica3,6,2);  
  
    cout<<aplikacja.Podaj_rachunki();  
    cin.get();  
}
```

1: 1 | Insert

E:\Dydaktyka\d1\Programowanie_obiektowe\w_11_5\Apli...

```
Nazwa: zeszyt, Cena detaliczna: 2  
Nazwa: pioro, Cena detaliczna: 5.76, Podatek: 20  
Nazwa: pioro, Cena detaliczna: 5.28, Podatek: 10
```

```
Rachunek : 1  
Wartosc rachunku: 0
```

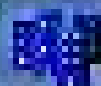
```
Rachunek : 2  
Wartosc rachunku: 0
```

```
Rachunek : 1  
Nazwa: zeszyt, Cena detaliczna: 2  
Ilosc produktu: 3, Wartosc zakupu: 6  
Nazwa: pioro, Cena detaliczna: 5.76, Podatek: 20  
Ilosc produktu: 16, Wartosc zakupu: 92.2  
Nazwa: pioro, Cena detaliczna: 5.28, Podatek: 10  
Ilosc produktu: 6, Wartosc zakupu: 31.7  
Wartosc rachunku: 130
```

```
Rachunek : 2  
Nazwa: zeszyt, Cena detaliczna: 2  
Ilosc produktu: 1, Wartosc zakupu: 2  
Nazwa: pioro, Cena detaliczna: 5.76, Podatek: 20  
Ilosc produktu: 2, Wartosc zakupu: 11.5  
Nazwa: pioro, Cena detaliczna: 5.28, Podatek: 10  
Ilosc produktu: 6, Wartosc zakupu: 31.7  
Wartosc rachunku: 45.2
```

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

1. **Wzorzec fasady – klasa TAplikacja**
hermetyzująca dostęp do kolekcji produktów i rachunków
2. **Budowa głównego formularza GUI**



Sporządzanie rachunkow



Pliki

Dane

Wyświetl

O programie

Wyświetl produkty

Wyświetl rachunki

Gotowa aplikacja

Zakładanie projektu

The screenshot shows the C++Builder 6 IDE interface. The 'New' menu is open, with 'Application' selected. The code editor displays the following C++ code:

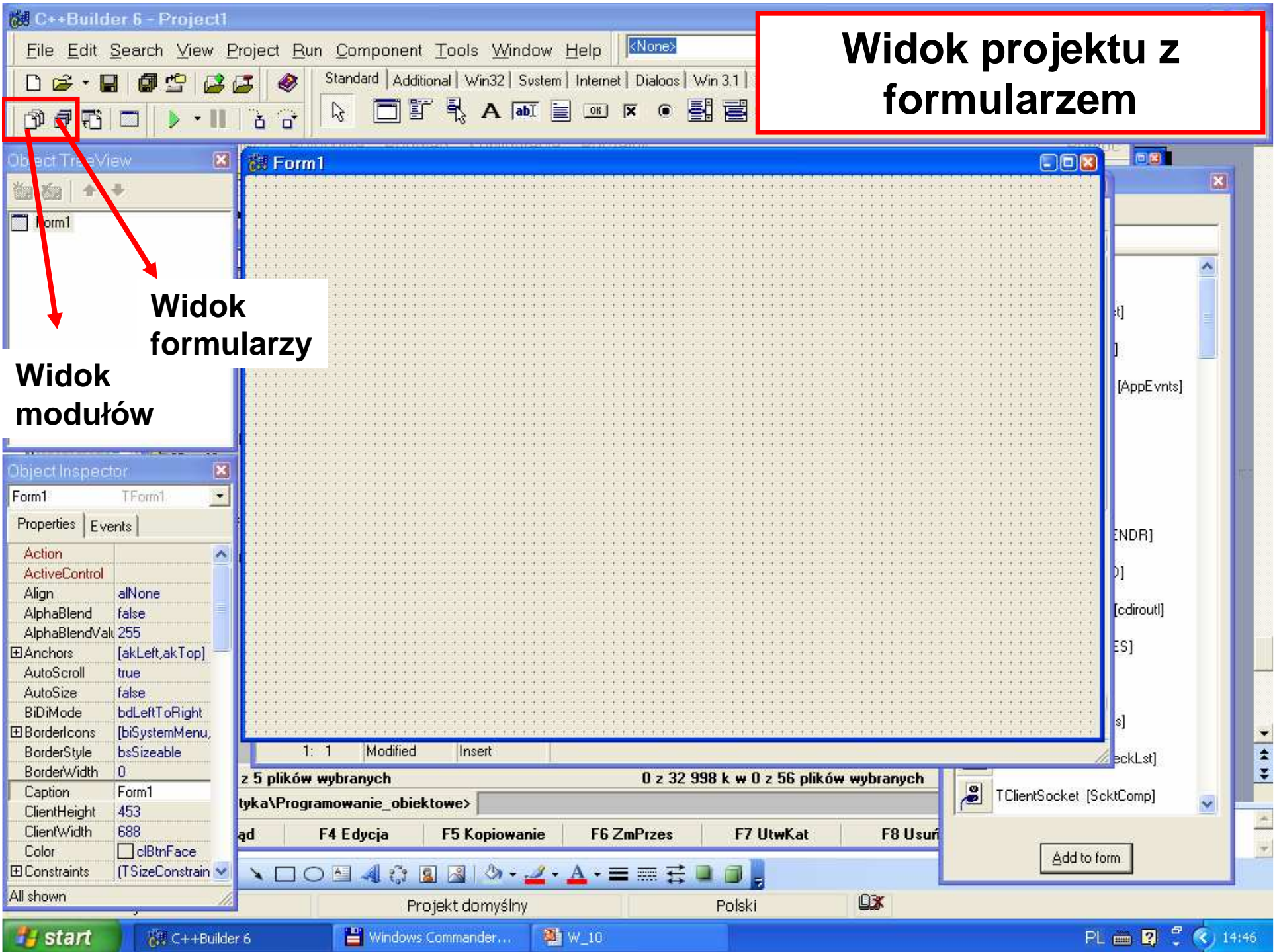
```
int TApplikacja::Wstaw_zakup(string* produkt, int ilosc, int numer)
(
    int wynik;
    TRachunek* poszukiwanyrachunek=new TRachunek(numer);
    TProdukt1* poszukiwanyprodukt=Wykonaj_produkt(produkt);
    if (poszukiwanyrachunek!=NULL && poszukiwanyprodukt!=NULL)
    {
        TRachunek* znalezionyrachunek;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

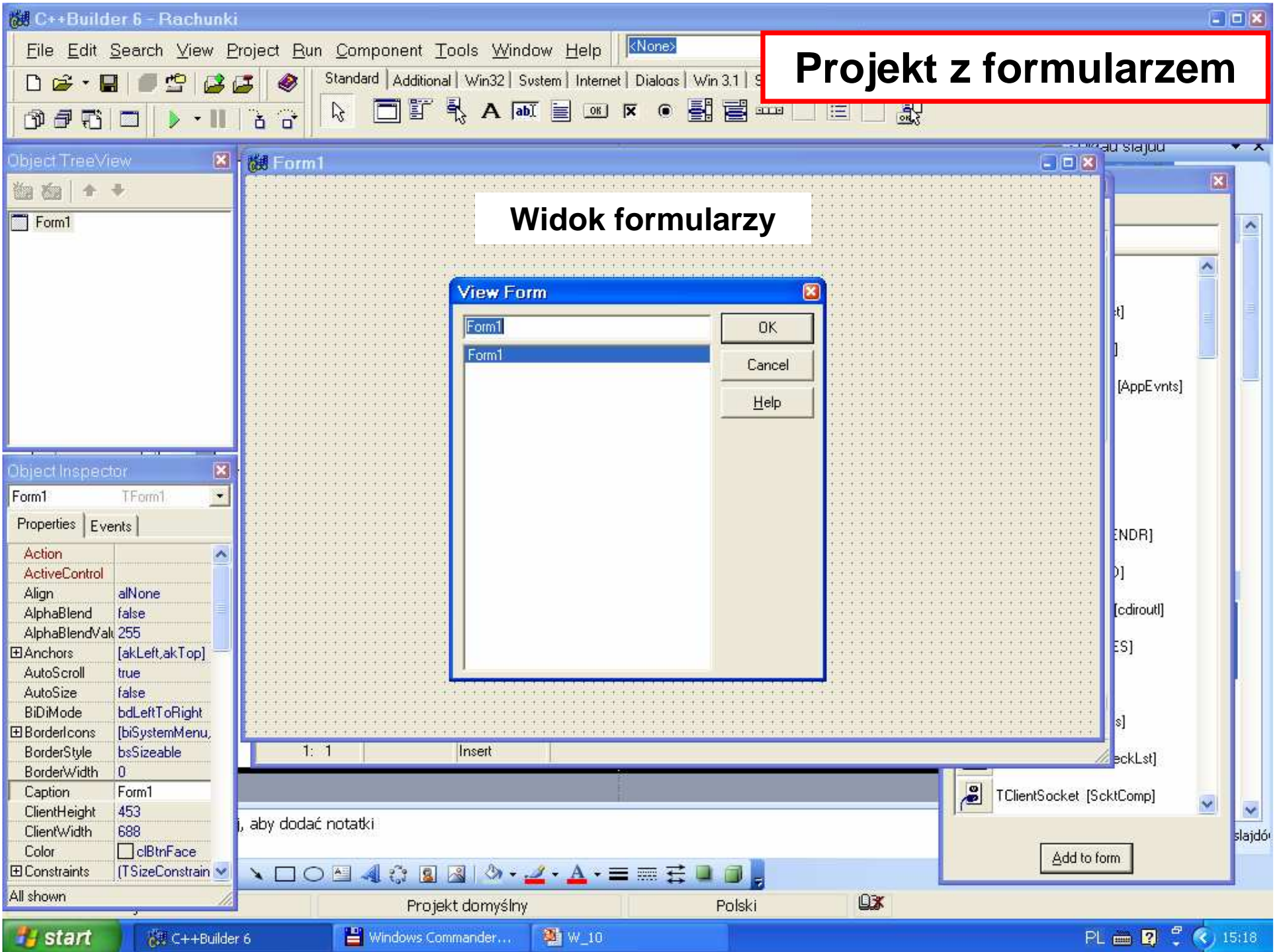
The Object Inspector shows the properties of the selected component, OKBottomDlg2. The 'Caption' property is set to 'swietl produkty'. The 'Color' property is set to 'clBtnFace'. The 'Constraints' property is set to '(TSizeConstrain)'. The 'Build' window shows two warnings:

```
[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
```

The status bar at the bottom shows 'Projekt domyślny' and 'Polski'.

Widok projektu z formularzem

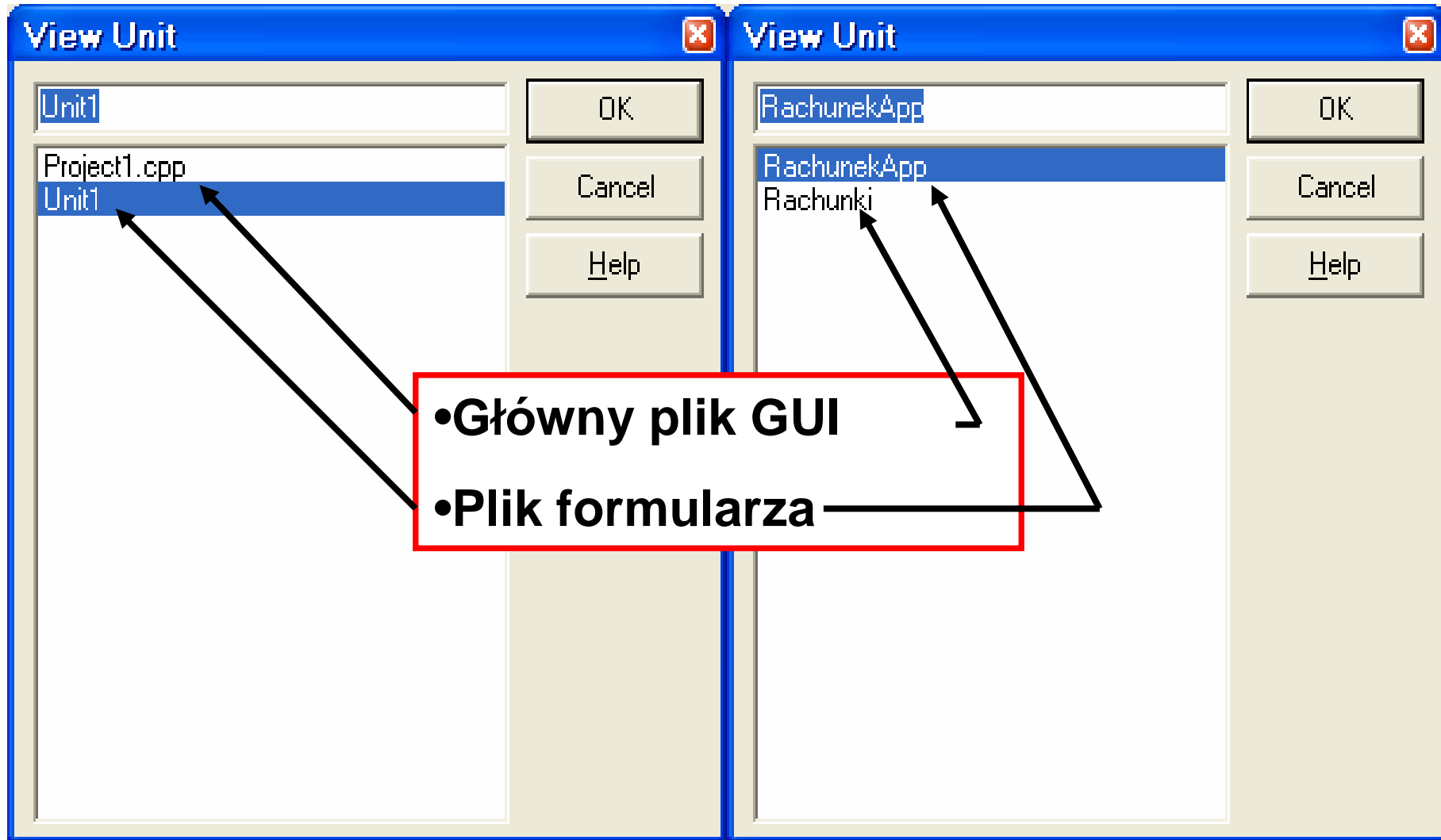


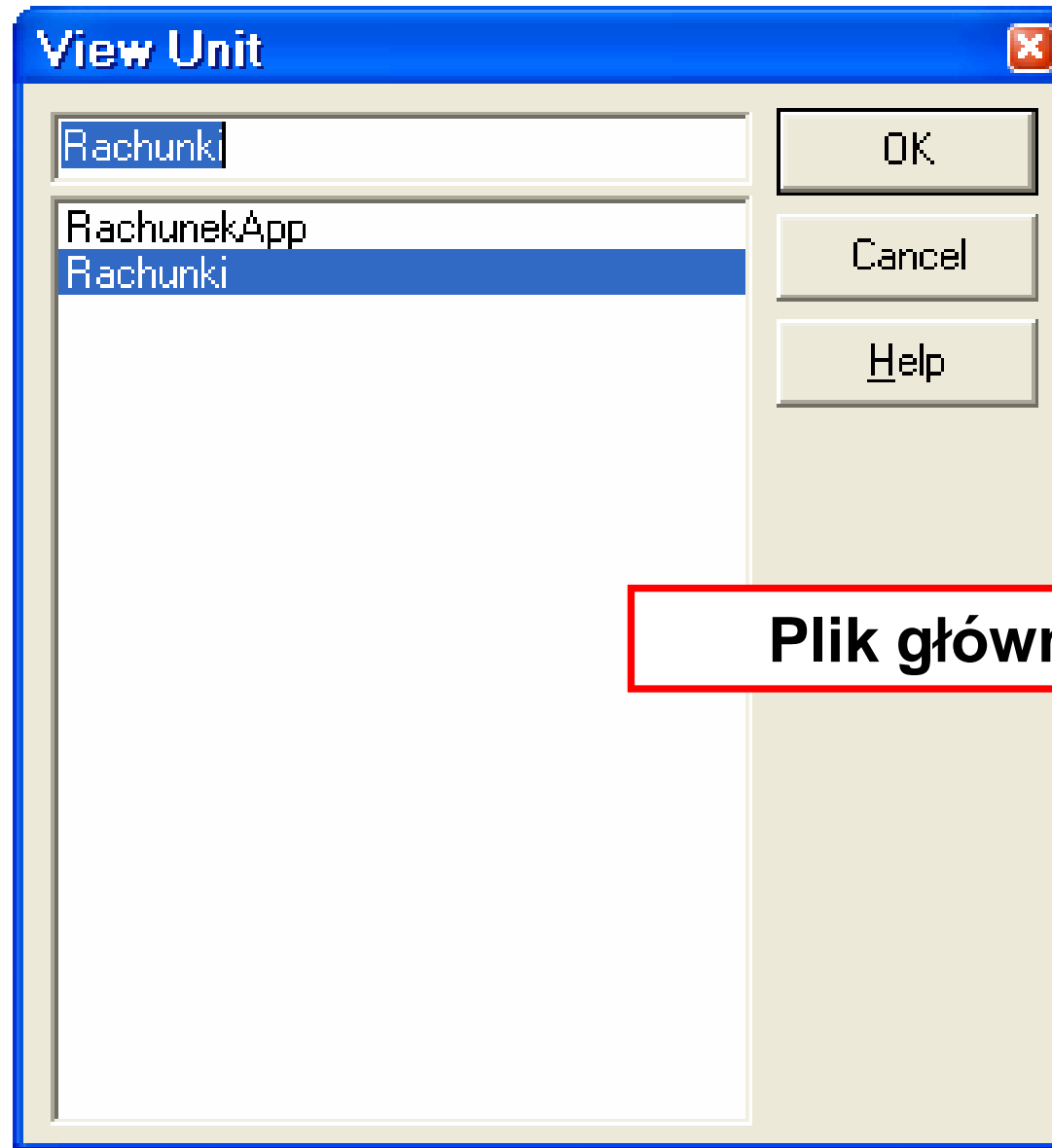


Widok modułów

Przed **Save Project As**

Po **Save Project As**





```
Rachunki.cpp
RachunekApp.h  Rachunki.cpp
//-----
#include <vc1.h>
#pragma hdrstop
//-----
USEFORM("RachunekApp.cpp", Form1);
//-----
WINAPI WinMain(HINSTANCE, HINSTANCE, LPST
{
    try
    {
        Application->Initialize();
        Application->CreateForm(__classid(TForm1), &Form1);
        Application->Run();
    }
    catch (Exception &exception)
    {
        Application->ShowException(&exception);
    }
    catch (...)
    {
        try
        {
            throw Exception("");
        }
        catch (Exception &exception)
        {
            Application->ShowException(&exception);
        }
    }
    return 0;
}
//-----
```

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

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
```

Plik formularza – plik
modułowy

RachunekApp.cpp

```
//-----  
  
#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

The image displays the C++Builder 6 development environment. The main window shows a form titled 'Form1' in the center. To the left is the 'Object TreeView' showing the form's structure. Below it is the 'Object Inspector' for 'Form1', showing various properties like 'Align', 'AlphaBlend', and 'Caption'. On the right side, there are two 'Components' palettes. The top one is a standard list of components, and the bottom one is a more detailed view with a search bar and an 'Add to form' button. The text 'Paleta komponentów wizualnych interfejsu użytkownika' is overlaid on the center of the image.

Object TreeView

Form1

Object Inspector

Form1 TForm1

| Property | Value |
|---------------|------------------------------------|
| 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 | 294 |
| Color | <input type="checkbox"/> clBtnFace |
| Constraints | (TSizeConstrain |

Projekt domyślny

Components

Search by name:

TBitBtn [Buttons]

- TToolBar [ComCtrls]
- TSpinButton [CSPIN]
- TSpinEdit [CSPIN]
- TDateTimePicker [ComCtrls]
- TDdeClientConv [DdeMan]
- TDdeClientItem [DdeMan]
- TDdeServerConv [DdeMan]
- TDdeServerItem [DdeMan]
- TDirectoryListBox [FileCtrl]
- TDrawGrid [Grids]
- TDriveComboBox [FileCtrl]
- TEdit [StdCtrls]
- TF1Book [VCF1]
- TFileListBox [FileCtrl]
- TFilterComboBox [FileCtrl]
- TFindDialog [Dialogs]
- TFontDialog [Dialogs]
- TGroupBox [StdCtrls]
- THeader [ExtCtrls]
- THeaderControl [ComCtrls]

Components

Search by name:

TBitBtn [Buttons]

- Frames
- TActionList [ActnList]
- TAnimate [ComCtrls]
- TApplicationEvents [AppEvnts]
- TBevel [ExtCtrls]
- TBitBtn [Buttons]
- TButton [StdCtrls]
- TCalendar [CCAENDR]
- TColorGrid [CGRID]
- TDirectoryOutline [cdirout]
- TCGauge [CGAUGES]
- TChartfx [Chartfx3]
- TCheckBox [StdCtrls]
- TCheckBoxList [CheckLst]
- TClientSocket [ScktComp]
- TColorBox [ExtCtrls]
- TColorDialog [Dialogs]
- TComboBox [StdCtrls]
- TComboBoxEx [ComCtrls]
- TControlBar [ExtCtrls]

Add to form

Add to form

start C++Builder 6 Windows Commander... W_11 15:21

The image displays two side-by-side screenshots of the Delphi Components palette. The left window, titled 'Components', shows a search for 'TCheckBox [CheckLst]'. The 'TButton [StdCtrls]' component is highlighted with a red box and the number '(1)'. The right window, also titled 'Components', shows a search for 'TChartfx [Chartfx3]'. The 'TCheckBox [StdCtrls]' component is highlighted with a red box and the number '(1)'. Both windows feature an 'Add to form' button at the bottom.

**Wybrane
komponenty w
aplikacji
(1) przycisk**

Components

Search by name:

TDateTimePicker [ComCtrls]

- TDdeClientConv [DdeMan]
- TDdeClientItem [DdeMan]
- TDdeServerConv [DdeMan]
- TDdeServerItem [DdeMan]
- TDirectoryListBox [FileCtrl]
- TDrawGrid [Grids]
- TDriveComboBox [FileCtrl]
- TEdit [StdCtrls] (2)**
- TF1Book [VCF1]
- TFileListBox [FileCtrl]
- TFilterComboBox [FileCtrl]
- TFindDialog [Dialogs]

Add to form

Components

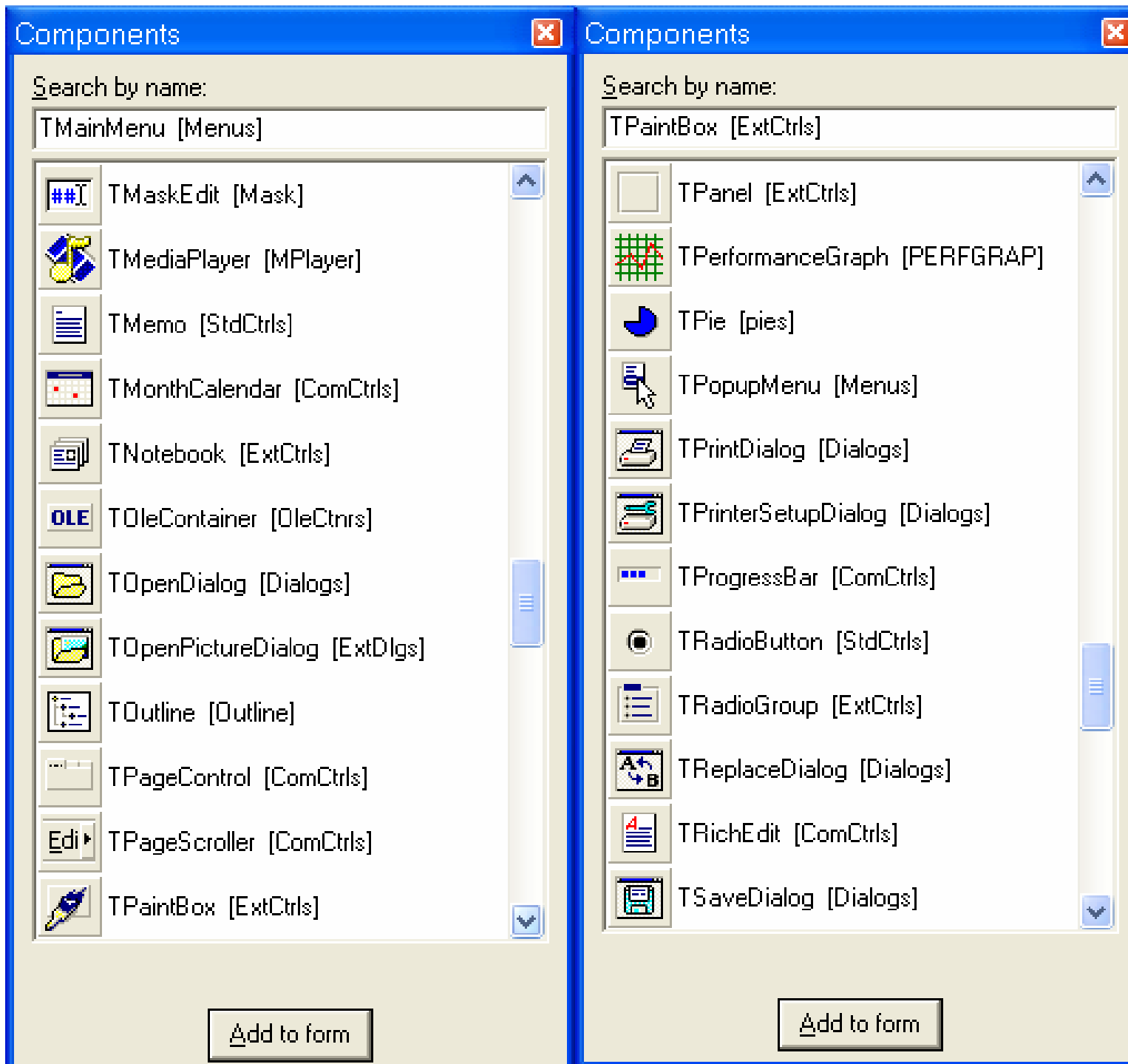
Search by name:

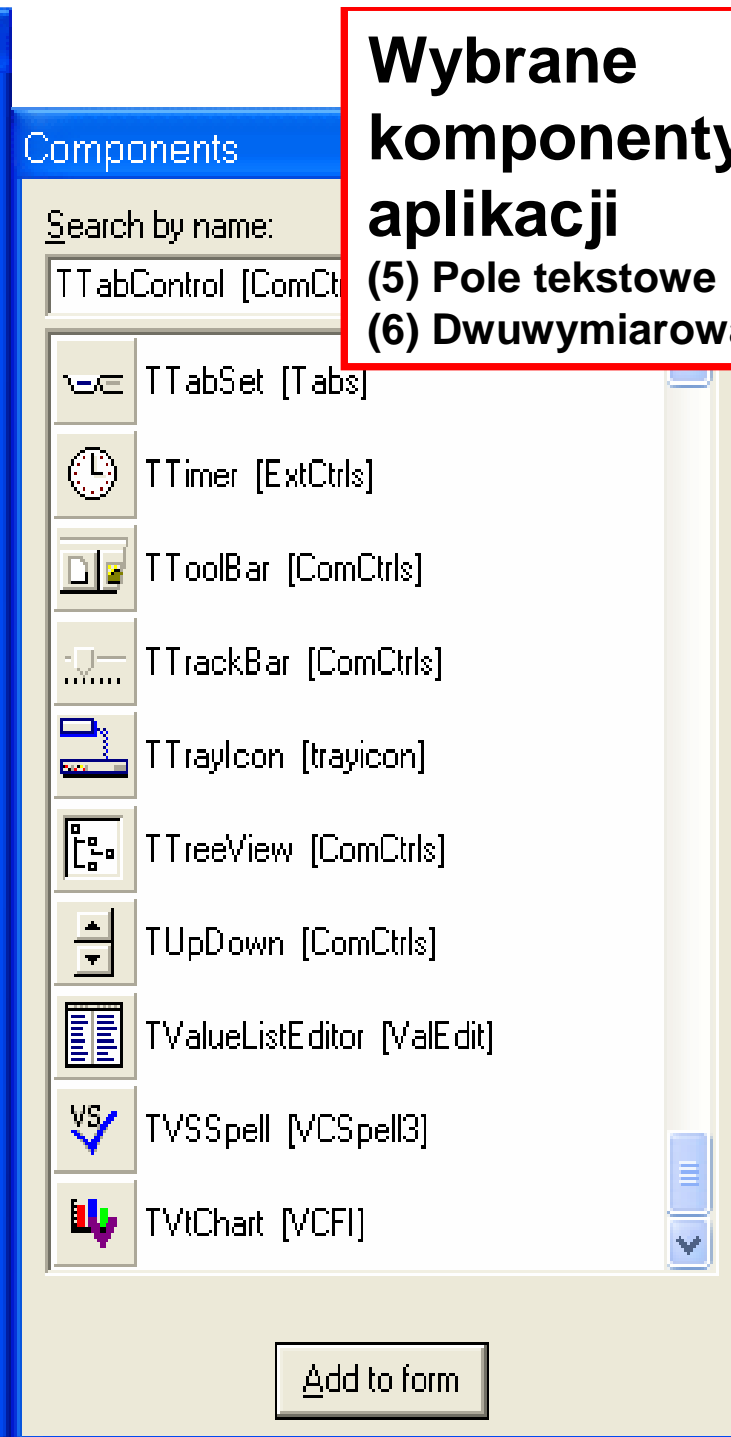
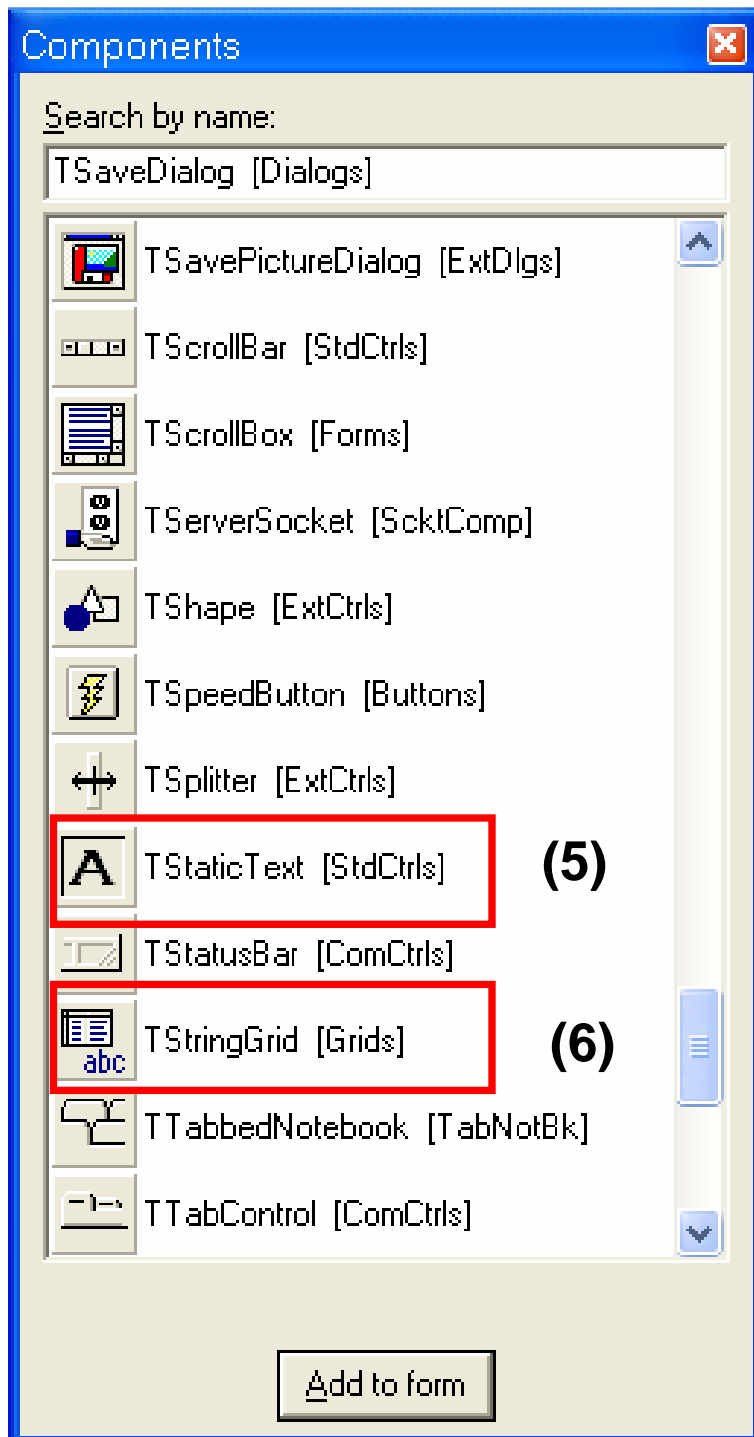
TFindDialog [Dialogs]

- TFontDialog [Dialogs]
- TGroupBox [StdCtrls]
- THeader [ExtCtrls]
- THeaderControl [ComCtrls]
- THotKey [ComCtrls]
- TImage [ExtCtrls]
- TImageList [Controls]
- TLabel [StdCtrls] (3)**
- TLabelEdit [ExtCtrls]
- TListBox [StdCtrls]
- TListView [ComCtrls]
- TMainMenu [Menus] (4)**

Add to form

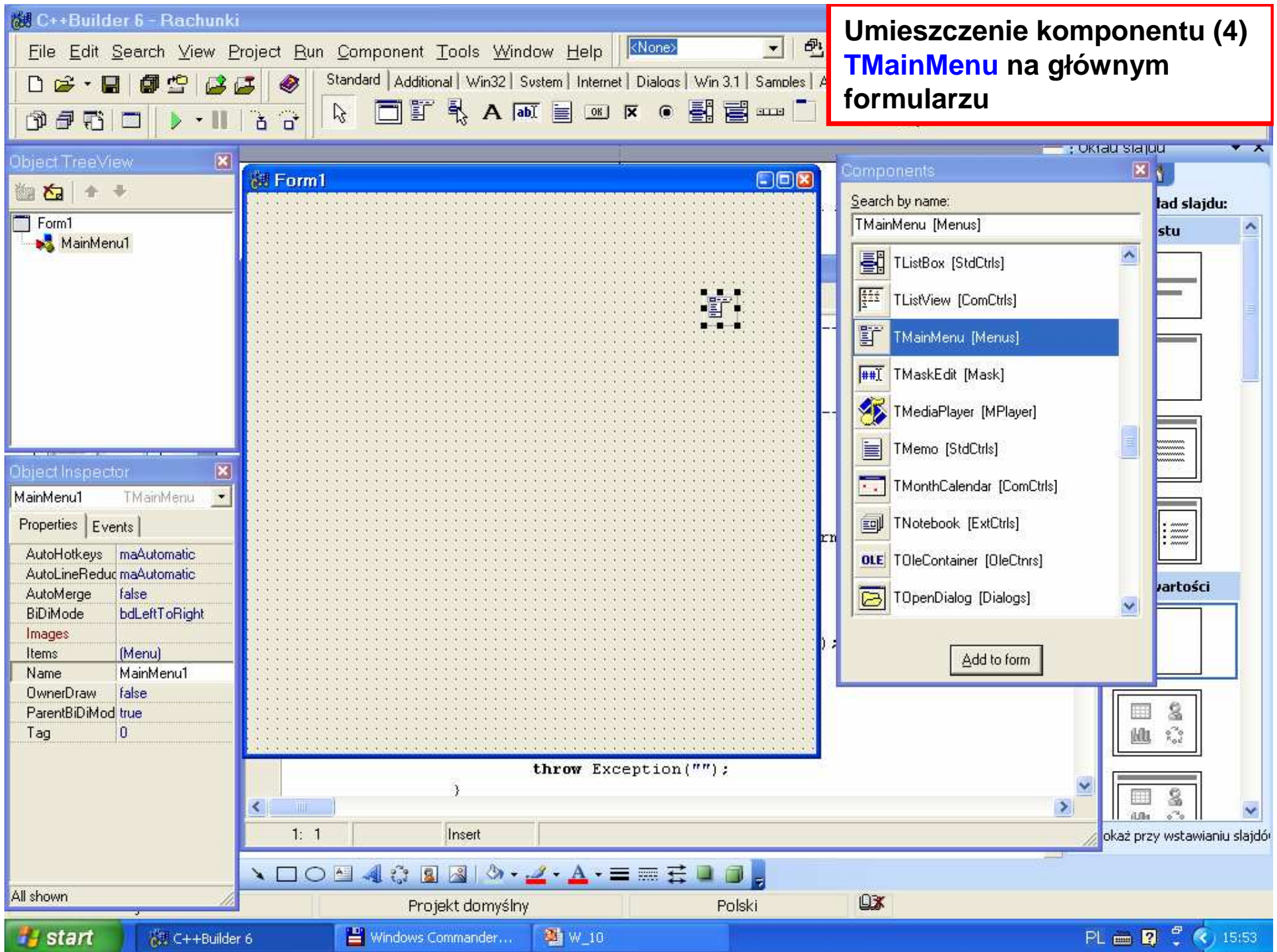
Wybrane komponenty w aplikacji
(2) Pole do wprowadzania danych
(3) Etykieta
(4) Menu głównego formularza



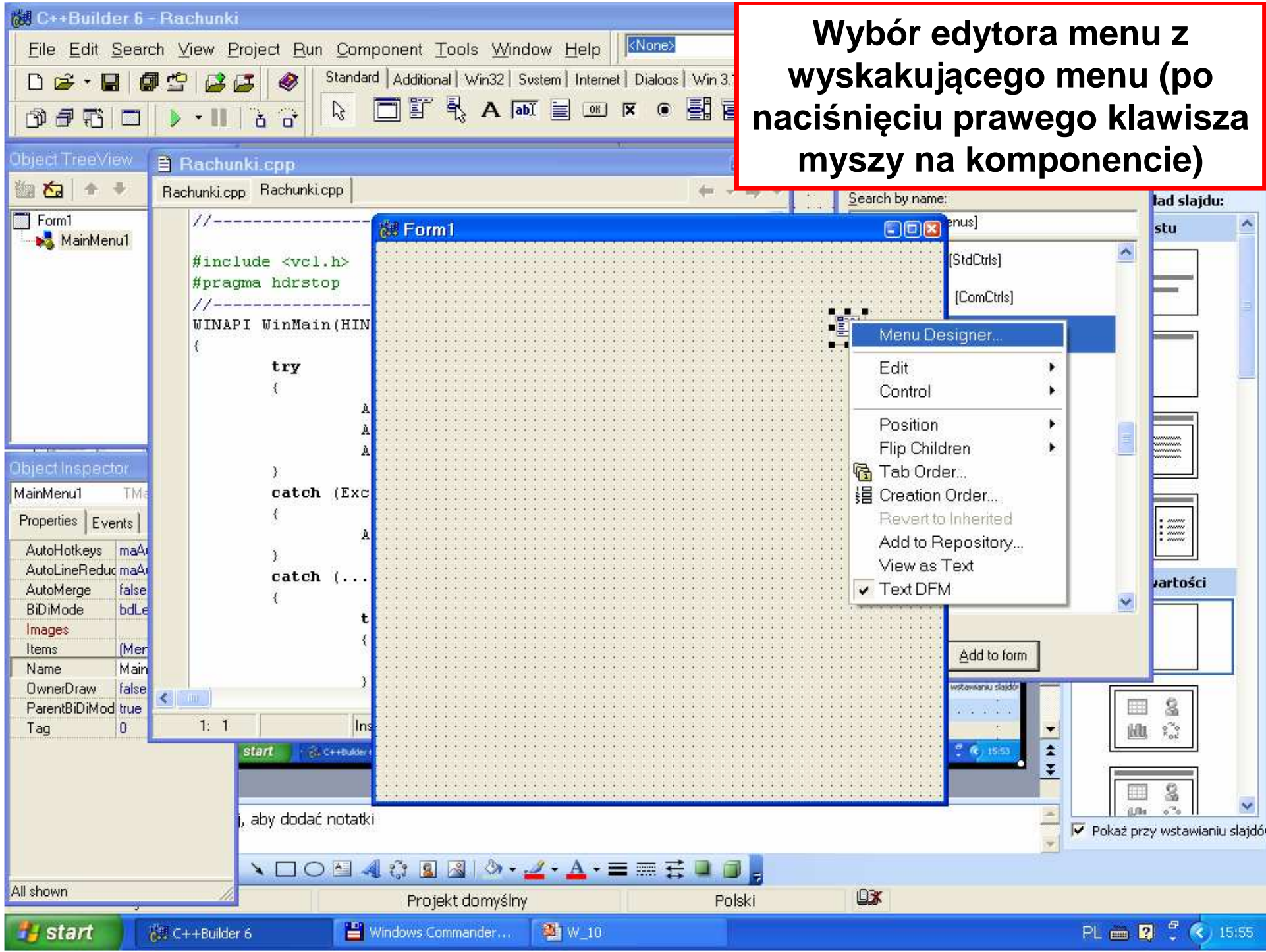


Wybrane komponenty w aplikacji
(5) Pole tekstowe
(6) Dwuwymiarowa tablica

**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

The screenshot displays the C++Builder 6 IDE interface. The main window shows a form titled 'Form1' with a menu bar 'Form1->MainMenu1' being edited. A context menu is open over the menu bar, listing options such as 'Insert', 'Delete', 'Create Submenu', 'Select Menu...', 'Save As Template...', 'Insert From Template...', 'Delete Templates...', and 'Insert From Resource...'. The 'Insert' option is selected. The 'Components' palette on the right shows various components, with 'Menu' selected. The 'Object Inspector' on the left shows the properties of the selected component. The 'Object TreeView' on the top left shows the project structure, including 'Form1' and 'MainMenu1'. The 'Code Editor' in the center shows the source code for 'Rachunki.cpp', which includes standard C++ headers and a main function. The 'Object Inspector' shows the properties of the selected component, including 'Action', 'AutoCheck', 'AutoHotkeys', 'AutoLineReduction', 'Bitmap', 'Break', 'Caption', 'Checked', 'Default', 'Enabled', 'GroupIndex', 'HelpContext', 'Hint', 'ImageIndex', 'Name', 'RadioItem', and 'Shortcut'. The 'Components' palette shows a list of components, including 'Menu', 'Mask', 'MPlayer', 'StdCtrls', 'Calendar', 'ExtCtrls', 'Dialogs', and 'Dialogs'. The 'Object TreeView' shows the project structure, including 'Form1' and 'MainMenu1'. The 'Code Editor' shows the source code for 'Rachunki.cpp', which includes standard C++ headers and a main function. The 'Object Inspector' shows the properties of the selected component, including 'Action', 'AutoCheck', 'AutoHotkeys', 'AutoLineReduction', 'Bitmap', 'Break', 'Caption', 'Checked', 'Default', 'Enabled', 'GroupIndex', 'HelpContext', 'Hint', 'ImageIndex', 'Name', 'RadioItem', and 'Shortcut'. The 'Components' palette shows a list of components, including 'Menu', 'Mask', 'MPlayer', 'StdCtrls', 'Calendar', 'ExtCtrls', 'Dialogs', and 'Dialogs'. The 'Object TreeView' shows the project structure, including 'Form1' and 'MainMenu1'. The 'Code Editor' shows the source code for 'Rachunki.cpp', which includes standard C++ headers and a main function.

```
//-----  
#include <vcl.h>  
#pragma hdrstop  
//-----  
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)  
{  
    try  
    {  
        Application->MainForm = Form1;  
        Application->MainForm->MainMenu1;  
    }  
    catch (Exception &E)  
    {  
        ShowMessage(E.Message);  
    }  
    catch (...)  
    {  
    }  
}
```

Object Inspector
Properties Events
Action
AutoCheck false
AutoHotkeys maP
AutoLineReduction maP
Bitmap (None)
Break mbN
Caption
Checked false
Default false
Enabled true
GroupIndex 0
HelpContext 0
Hint
ImageIndex -1
Name
RadioItem false
Shortcut (None)
All shown

Components
Search by name:
Menu [Menus]
Mask [Mask]
MPlayer [MPlayer]
StdCtrls
Calendar [ComCtrls]
ExtCtrls
Dialogs [Dialogs]
Dialogs

Object TreeView
Rachunki.cpp
Rachunki.cpp
Form1
MainMenu1

Code Editor
Rachunki.cpp
1: 1
Insert

Project: Projekt domyślny
Language: Polski

Taskbar: start, C++Builder 6, Windows Commander..., W_10, 15:56

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

The screenshot shows the C++Builder 6 IDE interface. The main window displays the 'Form1' design view, where a menu bar is being edited. The 'Object Inspector' is open, showing the properties of the selected menu item, with the 'Caption' property set to 'Plik'. The 'Component Palette' on the right shows the 'Menu' component selected. The 'Object TreeView' on the left shows the project structure, including 'Form1' and 'MainMenu1'. The 'Code Editor' shows the source code for 'Rachunki.cpp', which includes standard headers and a main function.

```
//-----  
#include <vcl.h>  
#pragma hdrstop  
//-----  
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)  
{  
    // TODO: Your code here  
}
```

26

27

28

| Object Inspector | |
|---------------------|----------|
| <Unnamed> TMenuItem | |
| Properties | |
| Action | |
| AutoCheck | false |
| AutoHotkeys | maParent |
| AutoLineReduction | maParent |
| Bitmap | (None) |
| Break | mbNone |
| Caption | Plik |
| Checked | false |
| Default | false |
| Enabled | true |
| GroupIndex | 0 |
| HelpContext | 0 |
| Hint | |
| ImageIndex | -1 |
| Name | |
| RadioItem | false |
| Shortcut | (None) |
| All shown | |

Kliknij, aby dodać notatki

Rysuj Autokształty

Slajd 27 z 28

Projekt domyślny

Polski

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

Object Inspector

| Property | Value |
|-------------------|--------------|
| Action | |
| AutoCheck | false |
| AutoHotkeys | maParent |
| AutoLineReduction | maParent |
| Bitmap | (None) |
| Break | mbNone |
| Caption | Dtworz &plik |
| Checked | false |
| Default | false |
| Enabled | true |
| GroupIndex | 0 |
| HelpContext | 0 |
| Hint | |
| ImageIndex | -1 |
| Name | |
| RadioItem | false |
| ShortCut | (None) |
| All shown | |

Object TreeView

- Form1
 - MainMenu1
 - Pliki (Pliki)

Form1 -> MainMenu1

Pliki

Component Palette

- [StdCtrls]
- [ComCtrls]
- [Menus]
- [Mask]
- [MPlayer]
- [StdCtrls]
- [ComCtrls]
- [ExtCtrls]
- [OleCtrls]
- [Dialogs]

Wartości

Slajd 28 z 29 Projekt domyślny Polski 15:59

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

The screenshot displays the C++Builder 6 IDE interface. The main window shows a form with a menu bar. The 'Object Inspector' is open, showing the 'Events' tab for a 'TMenuItem' component. The 'Action' property is set to 'false'. The 'Caption' property is set to 'Pliki'. The 'ShortCut' property is set to '(None)'. The 'Object TreeView' shows the 'MainMenu1' component with a sub-component 'Pliki (Plik1)'. The 'Form1' window shows the 'Pliki' menu item with a horizontal line next to it. The 'Component Palette' is open, showing the 'Menu' component.

| Property | Value |
|-------------------|----------|
| Action | false |
| AutoCheck | false |
| AutoHotkeys | maParent |
| AutoLineReduction | maParent |
| Bitmap | (None) |
| Break | mbNone |
| Caption | Pliki |
| Checked | false |
| Default | false |
| Enabled | true |
| GroupIndex | 0 |
| HelpContext | 0 |
| Hint | |
| ImageIndex | -1 |
| Name | |
| RadioItem | false |
| Shortcut | (None) |

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

The screenshot displays the C++Builder 6 IDE interface. The main window shows a form titled 'Form1' with a menu bar containing a 'Pliki' menu item. A context menu is open over the 'Pliki' item, showing 'Otworz plik' and 'Zapisz plik'. The 'Object Inspector' is open, showing the properties of the selected 'TMenuItem' object. The 'Properties' tab is active, and the 'Caption' property is set to 'Koniec'. The 'Object TreeView' on the left shows the project structure, including 'MainMenu1' and 'Pliki (Plik1)'. The 'Components' palette on the right is open, showing various components like 'Menu', 'MenuItem', and 'PopupMenu'. The 'Object TreeView' shows the following structure:

```
form1
├── MainMenu1
│   ├── Pliki (Plik1)
│   │   ├── Otworz &plik
│   │   └── &Zapisz plik
│   └── ...
└── ...
```

The 'Object Inspector' shows the following properties for the selected 'TMenuItem' object:

| Property | Value |
|-------------------|----------|
| Action | |
| AutoCheck | false |
| AutoHotkeys | maParent |
| AutoLineReduction | maParent |
| Bitmap | (None) |
| Break | mbNone |
| Caption | Koniec |
| Checked | false |
| Default | false |
| Enabled | true |
| GroupIndex | 0 |
| HelpContext | 0 |
| Hint | |
| ImageIndex | -1 |
| Name | |
| RadioItem | false |
| Shortcut | (None) |

The 'Components' palette shows the following components:

- Menu
- MenuItem
- PopupMenu
- StdCtrls
- ComCtrls
- ExtCtrls
- Dialogs

The 'Object TreeView' shows the following structure:

```
form1
├── MainMenu1
│   ├── Pliki (Plik1)
│   │   ├── Otworz &plik
│   │   └── &Zapisz plik
│   └── ...
└── ...
```

The 'Object Inspector' shows the following properties for the selected 'TMenuItem' object:

| Property | Value |
|-------------------|----------|
| Action | |
| AutoCheck | false |
| AutoHotkeys | maParent |
| AutoLineReduction | maParent |
| Bitmap | (None) |
| Break | mbNone |
| Caption | Koniec |
| Checked | false |
| Default | false |
| Enabled | true |
| GroupIndex | 0 |
| HelpContext | 0 |
| Hint | |
| ImageIndex | -1 |
| Name | |
| RadioItem | false |
| Shortcut | (None) |

The 'Components' palette shows the following components:

- Menu
- MenuItem
- PopupMenu
- StdCtrls
- ComCtrls
- ExtCtrls
- Dialogs

The 'Object TreeView' shows the following structure:

```
form1
├── MainMenu1
│   ├── Pliki (Plik1)
│   │   ├── Otworz &plik
│   │   └── &Zapisz plik
│   └── ...
└── ...
```

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

The screenshot displays the C++Builder 6 IDE interface. The main window shows a form titled "Sporządzanie rachunków" with a menu bar containing "Pliki", "Edycja", "Formatowanie", "Wstawianie", "Okno", and "Pomoc". The "Pliki" menu is expanded, showing a list of actions: "Otworz plik", "Zapisz plik", and "Koniec". The "Object Inspector" is open on the left, showing properties for the form. The "Object TreeView" on the far left shows the project structure, including "Form1" and "MainMenu1". The "Components" palette is open on the right, showing various standard components like "StdCtrls", "ComCtrls", and "Menus". The status bar at the bottom indicates "Slajd 31 z 32", "Projekt domyślny", and "Polski".

Object TreeView

Rachunki.cpp

```
//-----  
#include <vcl.h>  
#pragma hdrstop  
//-----  
WINAPI WinMain(HIN
```

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,biMinimize |
| BorderStyle | bsSizeable |
| BorderWidth | 0 |
| Caption | Sporządzanie rachunków |
| ClientHeight | 434 |
| ClientWidth | 459 |
| Color | <input type="checkbox"/> clBtnFace |
| Constraints | (TSizeConstraints) |
| All shown | |

Components

Search by name:

- StdCtrls
- ComCtrls
- Menus
- Mask
- MPlayer
- StdCtrls
- Calendar [ComCtrls]
- ExtCtrls
- DlgCtrls
- Dialogs

Add to form

Wstawianie slajdów

Pokaż przy wstawianiu slajdów

Slajd 31 z 32 Projekt domyślny Polski

```
RachunekApp.cpp
RachunekApp.h | Rachunki.cpp |
//-----
#ifndef RachunekAppH
#define RachunekAppH
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
//-----
class TForm1 : public TForm
{
    __published:      // IDE-managed Components
        TMainMenu *MainMenu1;
        TMenuItem *Plik1;
        TMenuItem *Otworzplik1;
        TMenuItem *Zapiszplik1;
        TMenuItem *N1;
        TMenuItem *Koniec1;
    private:          // User declarations
    public:           // User declarations
        __fastcall TForm1(TComponent* Owner);
};
//-----
extern PACKAGE TForm1 *Form1;
//-----
#endif
```

Atrybuty komponentu **TMainMenu** wstawione automatycznie do pliku nagłówkowego formularza głównego



Sporządzanie rachunków



Pliki Dane Wyświetl O programie

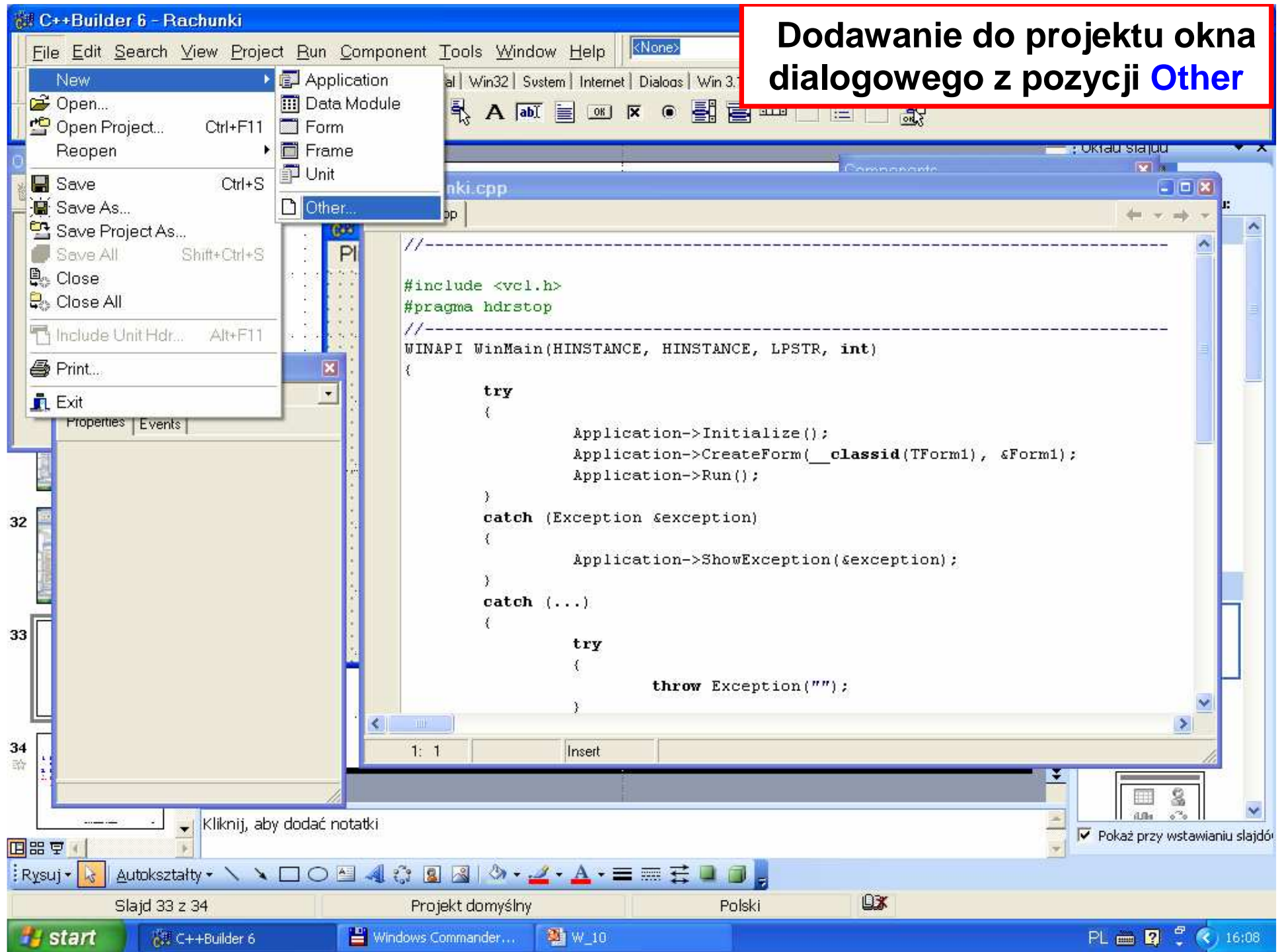


Gotowe Menu aplikacji

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

- 1. Wzorzec fasady – klasa TAplikacja
hermetyzująca dostęp do kolekcji produktów i
rachunków**
- 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



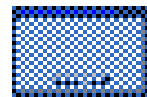
Dialog with
Help (Ho...



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**

The screenshot displays the C++Builder 6 IDE interface. The main window shows a form with a grid. A dialog box titled "Dodawanie produktów" is open over the form. The Object TreeView on the left shows a tree structure for "OKBottomDlg" containing "Bevel1", "CancelBtn", and "OKBtn". The Object Inspector on the bottom left shows the properties for "TOKBottomDlg", including "Caption" set to "Dodawanie produktów". The code editor in the foreground shows the implementation of the dialog class in "Unit1.cpp":

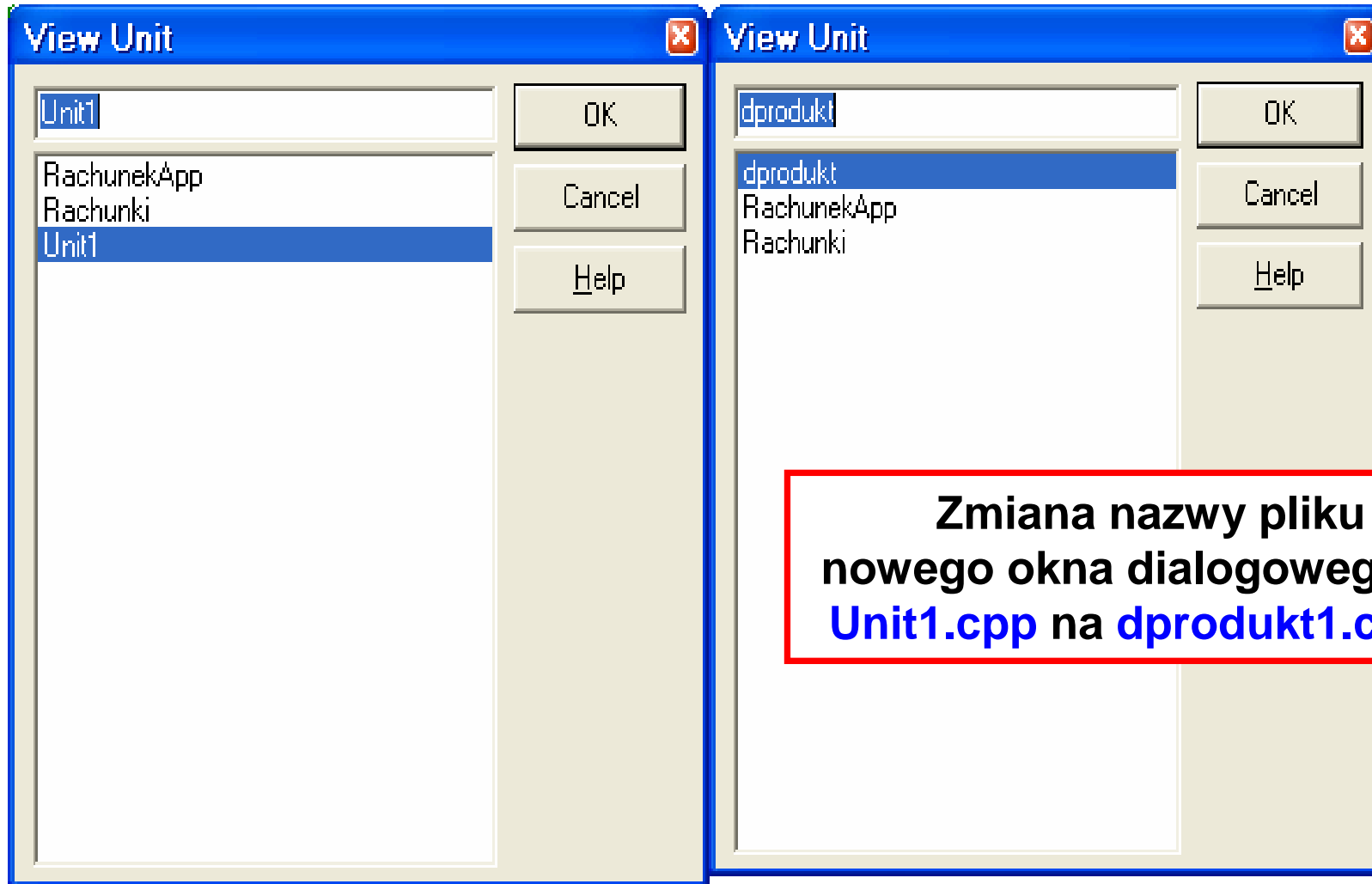
```
//-----  
#include <vcl.h>  
#pragma hdrstop  
  
#include "Unit1.h"  
//-----  
#pragma resource "*.dfm"  
TOKBottomDlg *OKBottomDlg;  
//-----  
__fastcall TOKBottomDlg::TOKBottomDlg(TComponent* AOwner)  
    : TForm(AOwner)  
{  
}  
//-----
```

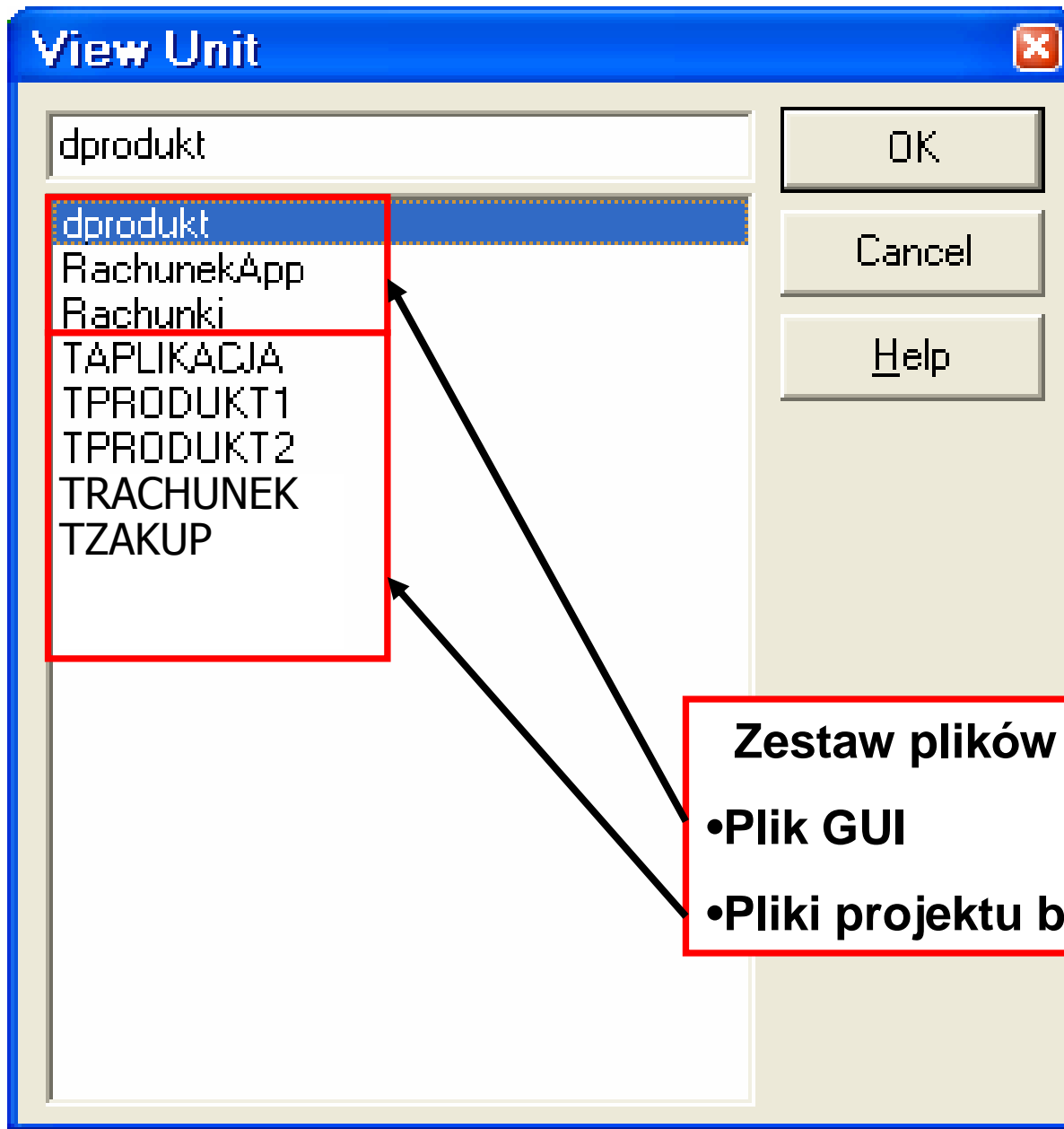
The status bar at the bottom indicates "Slajd 37 z 37", "Projekt domyślny", and "Polski".

Widok modułów

Przed Save As

Po Save As





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

dprodukt.cpp

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

Podstawowe elementy standardowego okna dialogowego

Object Inspector

| Properties | |
|---------------|------------------------------------|
| 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 | <input type="checkbox"/> clBtnFace |
| ⊕ Constraints | (TSizeConstraints) |
| All shown | |

notatki

Slajd 38 z 38 Projekt domyślny Polski

start C++Builder 6 Windows Commander... W_10 PL 16:18

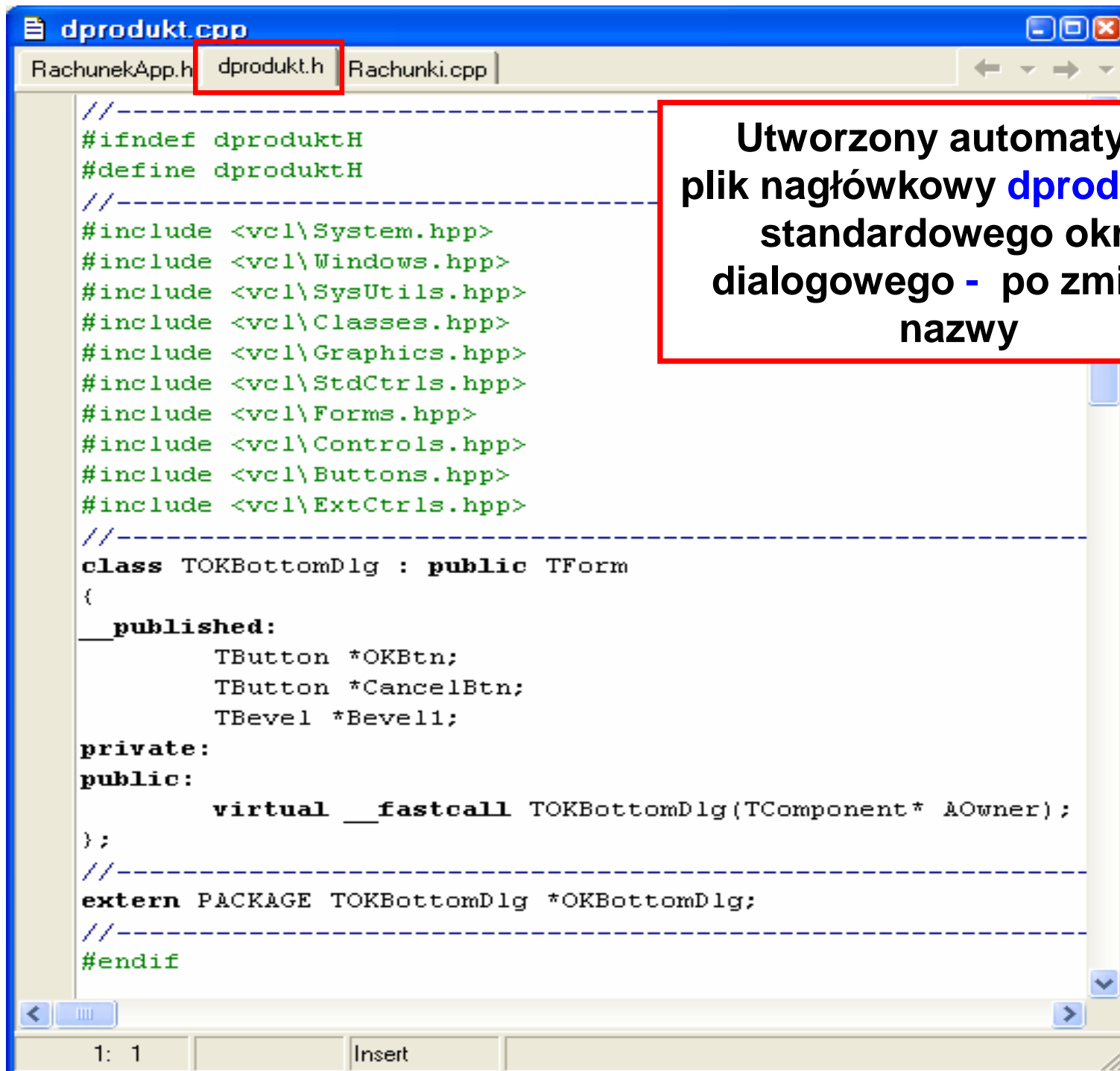
```
dprodukt.cpp
Rachunki.cpp dprodukt.cpp

//-----
#include <vcl.h>
#pragma hdrstop

#include "dprodukt.h"
//-----
#pragma resource "*.dfm"
TOKBottomDlg *OKBottomDlg;
//-----
__fastcall TOKBottomDlg::TOKBottomDlg(TComponent* AOwner)
    : TForm(AOwner)
{
}
//-----
```

3: 16 Insert

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

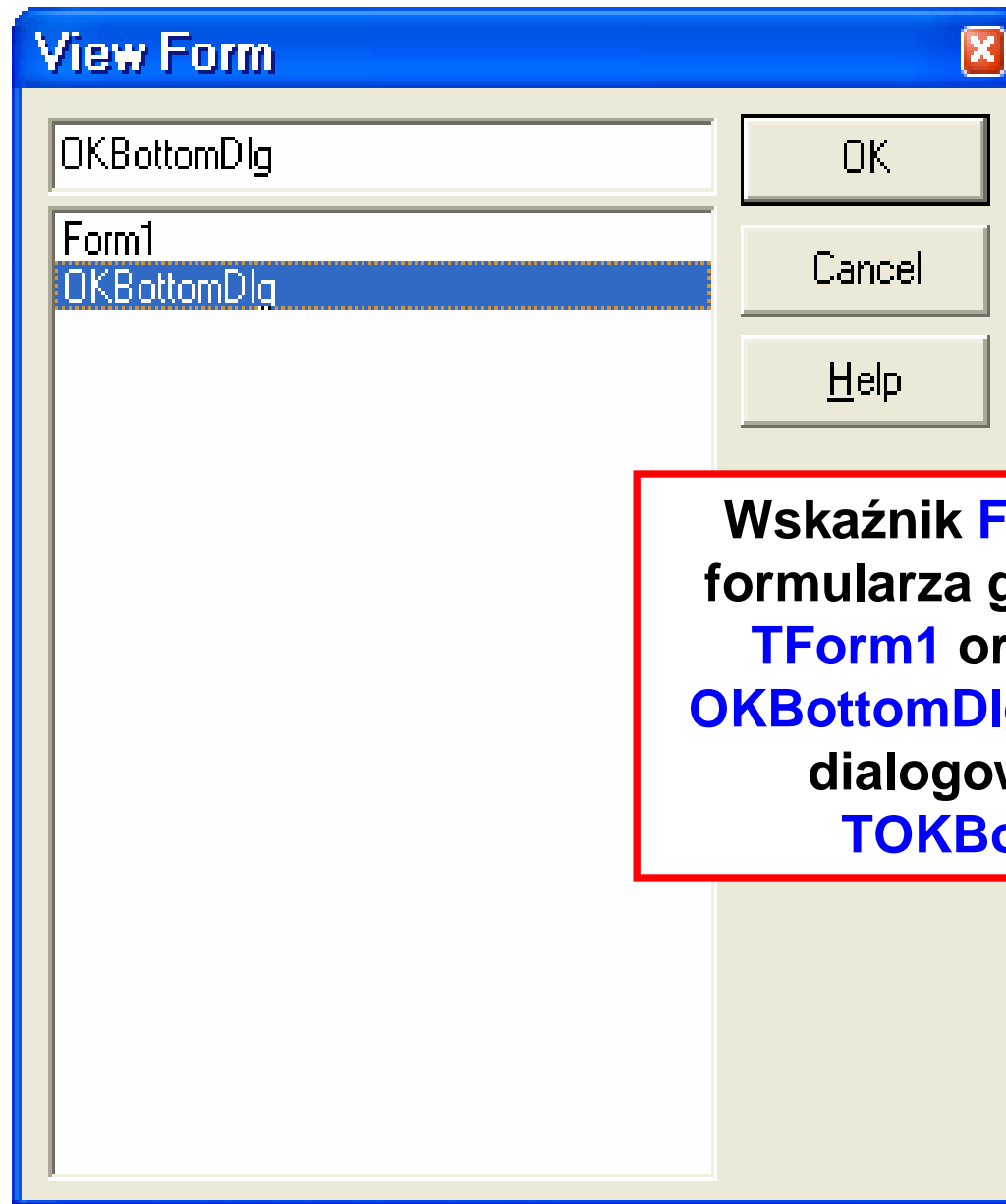


```
//-----  
#ifndef dproduktH  
#define dproduktH  
//-----  
#include <vcl\System.hpp>  
#include <vcl\Windows.hpp>  
#include <vcl\SysUtils.hpp>  
#include <vcl\Classes.hpp>  
#include <vcl\Graphics.hpp>  
#include <vcl\StdCtrls.hpp>  
#include <vcl\Forms.hpp>  
#include <vcl\Controls.hpp>  
#include <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
```

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

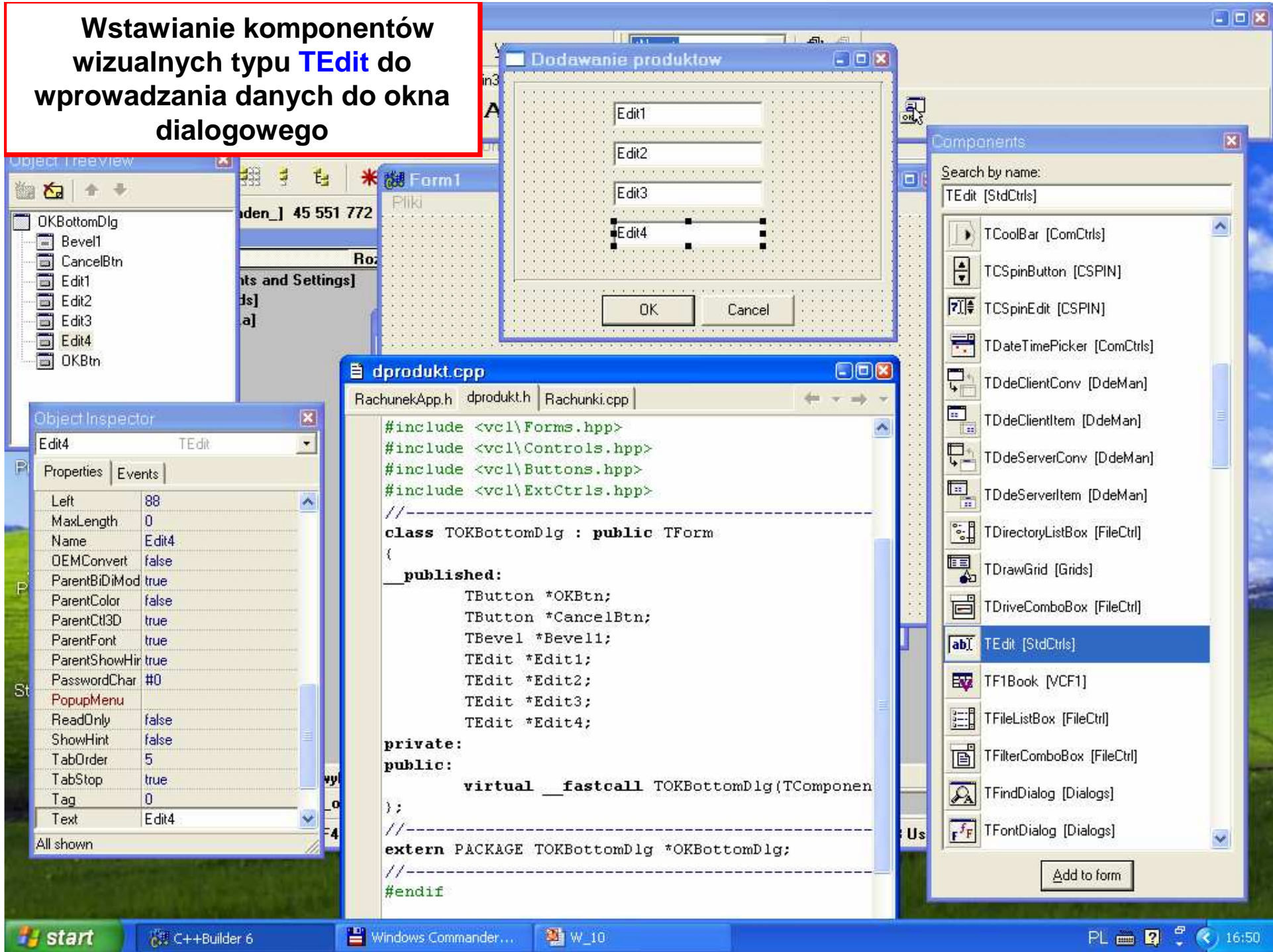
```
Rachunki.cpp
RachunekApp.h dprodukt.h Rachunki.cpp
//-----
#include <vc1.h>
#pragma hdrstop
//-----
USEFORM("RachunekApp.cpp", Form1);
USEFORM("dprodukt.cpp", OKBottomDlg);
//-----
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR,
{
    try
    {
        Application->Initialize();
        Application->CreateForm(__classid(TForm1), &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;
}
```

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



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

#include <vc1\Forms.hpp>
#include <vc1\Controls.hpp>
#include <vc1\Buttons.hpp>
#include <vc1\ExtCtrls.hpp>
//-----
class TOKBottomDlg : public TForm
{
    __published:
        TButton *OKBtn;
        TButton *CancelBtn;
        TBevel *Bevel1;
        TEdit *Edit1;
        TEdit *Edit2;
        TEdit *Edit3;
        TEdit *Edit4;
    private:
    public:
        virtual __fastcall TOKBottomDlg(TComponent
};
//-----
extern PACKAGE TOKBottomDlg *OKBottomDlg;
//-----
#endif
```

| Object Inspector | |
|------------------|--------|
| Edit4 TEdit | |
| Properties | Events |
| Left | 88 |
| MaxLength | 0 |
| Name | Edit4 |
| OLEMConvert | false |
| ParentBiDiMod | true |
| ParentColor | false |
| ParentCtl3D | true |
| ParentFont | true |
| ParentShowHint | true |
| PasswordChar | #0 |
| PopupMenu | |
| ReadOnly | false |
| ShowHint | false |
| TabOrder | 5 |
| TabStop | true |
| Tag | 0 |
| Text | Edit4 |
| All shown | |

- Components
- Search by name:
- TEdit [StdCtrls]
 - TCoolBar [ComCtrls]
 - TCSpinButton [CSPIN]
 - TCSpinEdit [CSPIN]
 - TDateTimePicker [ComCtrls]
 - TDdeClientConv [DdeMan]
 - TDdeClientItem [DdeMan]
 - TDdeServerConv [DdeMan]
 - TDdeServerItem [DdeMan]
 - TDirectoryListBox [FileCtrl]
 - TDrawGrid [Grids]
 - TDriveComboBox [FileCtrl]
 - TEdit [StdCtrls]
 - TF1Book [VCF1]
 - TFileListBox [FileCtrl]
 - TFilterComboBox [FileCtrl]
 - TFindDialog [Dialogs]
 - TFontDialog [Dialogs]
- Add to form

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

The screenshot displays the Delphi IDE interface with several windows open:

- Object Treeview:** Shows the component hierarchy for the dialog box, including OKBottomDlg, Bevel1, CancelBtn, Edit1, Edit2, Edit3, Edit4, Label1, Label2, Label3, and Label4.
- Object Inspector:** Shows the properties for the selected TLabel component (Label4), such as Align, Alignment, Anchors, AutoSize, BiDiMode, Caption, Color, Constraints, Cursor, DragCursor, DragKind, DragMode, Enabled, FocusControl, Font, Height, and HelpContext.
- Code Editor (dprodukt.cpp):** Shows the C++ code for the TOKBottomDlg class, including header files and member variables for buttons, bevels, edit boxes, and labels.
- Components Palette:** Shows the list of available components, with TLabel [StdCtrls] selected.
- Dialog Box (Dodawanie produktow):** Shows the visual design of the dialog box with four edit boxes (Edit1-4) and their corresponding labels (Label1-4).

```
#include <vc1\Forms.hpp>
#include <vc1\Controls.hpp>
#include <vc1\Buttons.hpp>
#include <vc1\ExtCtrls.hpp>
//-----
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:
        virtual __fastcall TOKBottomDlg (TComponent
```

Dodawanie produktów

Nazwa

Cena

Podatek

Promocja

OK Cancel

**Gotowy
projekt okna
dialogowego**

Dodawanie produktów X

| | |
|----------|--------------------------------|
| Nazwa | <input type="text" value="1"/> |
| Cena | <input type="text" value="1"/> |
| Podatek | <input type="text" value="1"/> |
| Promocja | <input type="text" value="1"/> |

Uruchomione okno dialogowe

```
dprodukt.cpp
RachunekApp.cpp dprodukt.h

#ifndef dproduktH
#define dproduktH
//-----
#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 "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

Definicja metody `Execute()` do wprowadzania danych w oknie dialogowym

```
bool __fastcall TOKBottomDlg::Execute()
{
    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
        //oraz wyznaczenie typu produktu w tab[0] w zaleznosci, czy wprowadzono
        //podatek      if(tab[3]!="")
            if(tab[3]!="")
                tab[0]="2"; //jest podatek - typ TProdukt2
            else
                tab[0]="1"; //brak podatku i promocji - typ TProdukt1
        return true;}
    else
        return false;
}
```

```
// -----  
#ifndef RachunekApp1H  
#define RachunekApp1H  
// -----  
#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"
```

14: 26 Modified Insert

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


```
C:\Settings\dvdaktyka\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 *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);
private:      // User declarations
public:      // User declarations
    TAplikacja aplikacja; //połączenie z obiektem uzytkownika
    __fastcall TForm1(TComponent* Owner);
};
//-----
extern PACKAGE TForm1 *Form1; //deklaracja globalnego wskaznika
//-----
#endif
```

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

Połączenie GUI z obiektami aplikacji za pomocą obiektu hermetyzującego aplikacja typu **TAplikacja**

The image shows a screenshot of the C++Builder 6 IDE. The main window displays the source code for `RachunekApp.cpp`. The code defines a `TForm1` class that inherits from `TForm`. It lists several menu items in the `published` section, including `*Dodajprodukt1`. A method `void __fastcall Dodajprodukt1Click(TObject *Sender);` is also visible. The `Object TreeView` on the left shows the form structure with `MainMenu1`. The `Object Inspector` at the bottom left shows properties for `Form1`. A dialog box titled "Sporządzanie rachunków" is open, showing a menu with options: "Dodaj produkt", "Dodaj rachunek", and "Dodaj zakup". A red box highlights a text block on the right side of the image, which explains the connection between the menu item and the code method.

```
class TForm1 : public TForm
(
    published:    // IDE-managed Components
    TMainMenu *MainMenu;
    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);
);
```

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

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", "");
}
```

```
RachunekApp.cpp
dprodukt.cpp drachunek.cpp RachunekApp.cpp
#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",
               "Ustawiono dane",
               "Brak pamieci",
               "Brak rachunku",
               "Brak produktu",
               "Wybrany produkt kupiono wczesniej - powiekszo jego ilosc"};
//-----
__fastcall TForm1::TForm1(TComponent* Owner) //automatycznie
    : TForm(Owner)
{
}
```

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

The image shows a screenshot of a Microsoft PowerPoint presentation with a blue title bar. The main window is titled "Microsoft PowerPoint - [W_10]". The menu bar includes "Plik", "Edycja", "Widok", "Wstaw", "Format", "Narzędzia", and "Pokaż slajdów". The slide navigation pane on the left shows slides 56 through 59. A "Sporządzanie rachunków" menu is open, with "Dodaj produkt" selected. A "Dodawanie produktów" dialog box is displayed in the foreground, containing four input fields: "Nazwa" (1), "Cena" (1), "Podatek" (1), and "Promocja" (empty). Below the fields are "OK" and "Cancel" buttons. A "Wstawiono dane" message box with an "OK" button is also visible. A red-bordered text box in the lower-left corner contains the text: "Uruchomienie aplikacji z oknem dialogowym do wprowadzania danych – przypadek poprawnego wprowadzania danych". The taskbar at the bottom shows the Start button, taskbar buttons for "C++Builder 6", "Windows Commander...", "W_10", and "Rachunki", and the system tray with "PL", a help icon, a refresh icon, and the time "18:26".

Dodawanie produktów

Nazwa: 1
Cena: 1
Podatek: 1
Promocja:

OK Cancel

Wstawiono dane

OK

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

Sporządzanie rachunków

Plik Dane Wyświetl O programie

Dodaj produkt
Dodaj rachunek
Dodaj zakup

Układy zawartości

Pokaż przy wstawianiu slajdów

Slajd 59 z 71 Projekt domyślny Polski

start C++Builder 6 Windows Commander... W_10 Rachunki PL 18:26

The image shows a screenshot of a Microsoft PowerPoint presentation titled "Sporządzanie rachunków" (Preparing invoices). The menu bar includes "Pliki", "Edycja", "Widok", "Wstaw", "Format", "Narzędzia", and "Pokaż slajdy". The "Wstaw" menu is open, showing options: "Dodaj produkt", "Dodaj rachunek", and "Dodaj zakup".

A dialog box titled "Dodawanie produktów" (Adding products) is displayed, with the following fields and values:

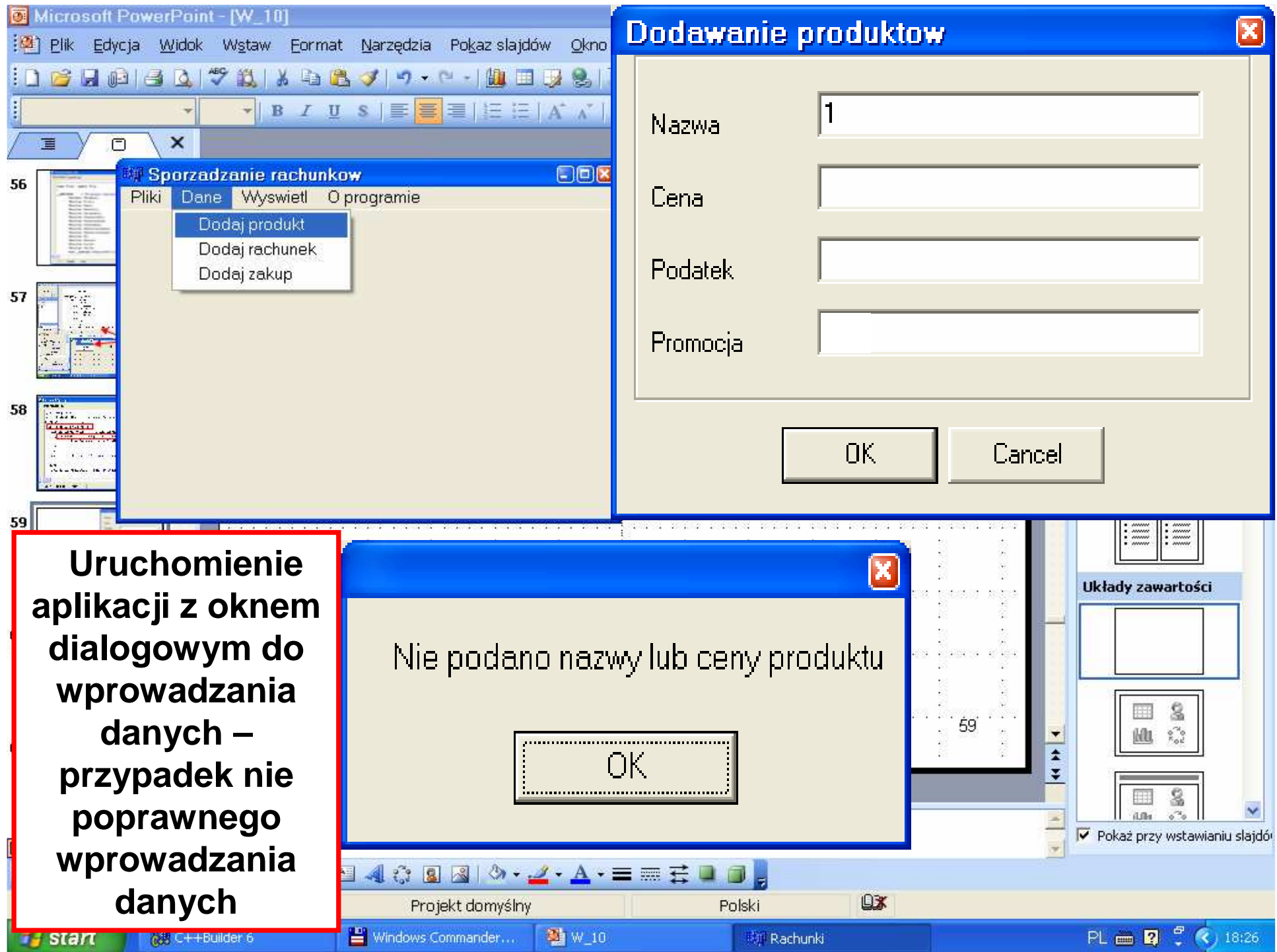
| Field | Value |
|----------|-------|
| Nazwa | 1 |
| Cena | 1 |
| Podatek | 1 |
| Promocja | |

Buttons "OK" and "Cancel" are visible at the bottom of the dialog box.

An error message dialog box is shown with the text: "Proba wstawienia tej samej danej" (Attempt to insert the same data) and an "OK" button.

A red-bordered text box contains the following text: "Uruchomienie aplikacji z oknem dialogowym do wprowadzania danych – przypadek ponownego wprowadzenia tej samej danej" (Running the application with a dialog box for data entry – case of re-adding the same data).

The bottom of the screen shows the PowerPoint status bar with "Slajd 59 z 71", "Projekt domyślny", "Polski", and the Windows taskbar with the Start button and open applications: "C++Builder 6", "Windows Commander...", "W_10", and "Rachunki". The system tray shows "PL", a help icon, a refresh icon, and the time "18:25".



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

Nie podano nazwy lub ceny produktu

OK