МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Российский государственный университет им. А.Н. Косыгина   
(Технологии. Дизайн. Искусство)»**

Институт ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ЦИФРОВЫЕ ТРАНСФОРМАЦИИ

**Отчет по лабораторной работе № 10**

**по дисциплине «Объектно-ориентированное программирование»**

**Тема: «Классы. Наследование и отработка взаимодействия с компонентом StringGrid»**

Выполнил: Сидоров Д. С., группа ИТС-123

Проверил: к.т.н., доц. Семёнов А. А.

Москва, 2025г

**Классы. Наследование и отработка взаимодействия с компонентом StringGrid**

Лабораторная работа посвящена изучению классов в C++. Перед выполнением лабораторной работы рекомендуется ознакомиться с лекционным материалом по общему и частному наследованию в классах.

Задание:

Изучить и повторить проиллюстрированный ниже пример, демонстрирующий применение технологии наследования при изначальном проектировании приложения.

Взяв за основу проект из лабораторной работы 9, мы создадим его улучшенную версию, которая позволит добавлять товары не в компонент Memo, а в компонент StringGrid. Также реализуем сохранение данных в текстовый файл и считывание их из него в компонент StringGrid.

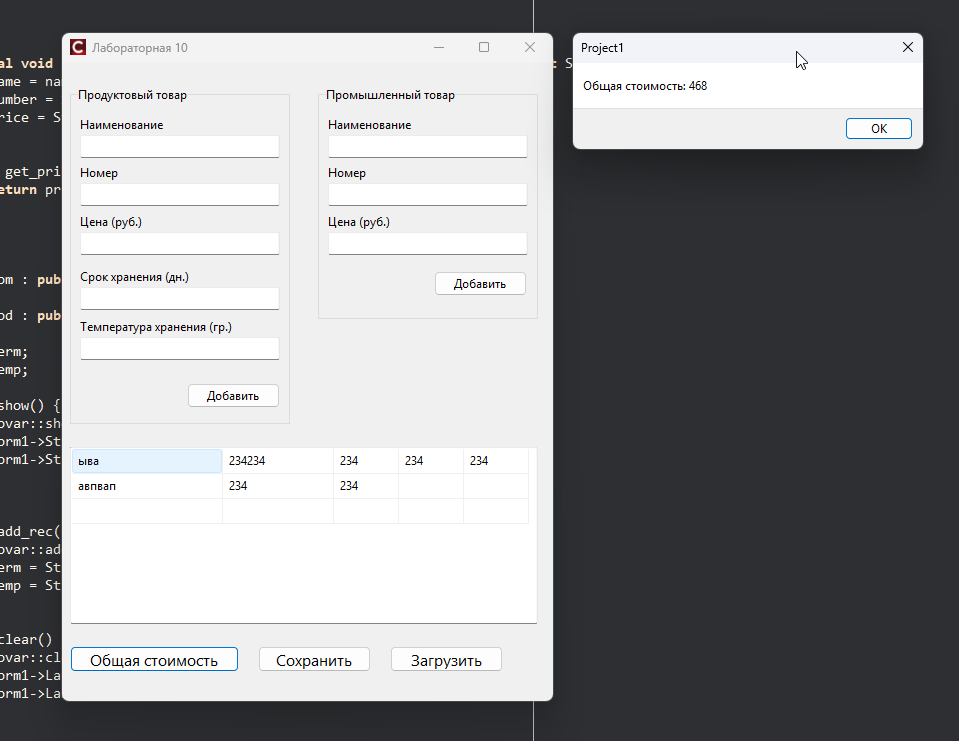


Рисунок 1. Работающий вариант программы

**Листинг кода**

**Unit1.cpp**

/////---------------------------------------------------------------------------

#include <vcl.h>

#include <fstream.h>

#include <System.StrUtils.hpp>

#pragma hdrstop

#include "Unit1.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm1 \*Form1;

int ProductType = 0;

int i = 0, j = 0;

class Tovar {

private:

String name;

int number;

float price;

public:

Tovar() : name(""), number(0), price(0) {};

virtual ~Tovar() {};

void add\_rec() {

if(ProductType == 0) {

name = Form1->LabeledEdit1->Text;

number = StrToInt(Form1->LabeledEdit2->Text);

price = StrToFloat(Form1->LabeledEdit3->Text);

}

if(ProductType == 1) {

name = Form1->LabeledEdit6->Text;

number = StrToInt(Form1->LabeledEdit7->Text);

price = StrToFloat(Form1->LabeledEdit8->Text);

}

}

void show() {

Form1->StringGrid1->Cells[0][i+j] = name;

Form1->StringGrid1->Cells[1][i+j] = number;

Form1->StringGrid1->Cells[2][i+j] = price;

}

void clear() {

if(ProductType == 0) {

Form1->LabeledEdit1->Clear();

Form1->LabeledEdit2->Clear();

Form1->LabeledEdit3->Clear();

}

if(ProductType == 1) {

Form1->LabeledEdit6->Clear();

Form1->LabeledEdit7->Clear();

Form1->LabeledEdit8->Clear();

}

}

virtual void add\_grid(const String& nameStr, const String& numberStr, const String& priceStr) {

name = nameStr;

number = StrToInt(numberStr);

price = StrToFloat(priceStr);

}

float get\_price() {

return price;

}

};

class TovarProm : public Tovar {};

class TovarProd : public Tovar {

private:

int term;

int temp;

public:

void show() {

Tovar::show();

Form1->StringGrid1->Cells[3][i+j] = term;

Form1->StringGrid1->Cells[4][i+j] = temp;

}

void add\_rec() {

Tovar::add\_rec();

term = StrToInt(Form1->LabeledEdit4->Text);

temp = StrToInt(Form1->LabeledEdit5->Text);

}

void clear() {

Tovar::clear();

Form1->LabeledEdit4->Clear();

Form1->LabeledEdit5->Clear();

}

void add\_grid(const String& nameStr, const String& numberStr, const String& priceStr, const String& termStr, const String& tempStr) {

Tovar::add\_grid(nameStr, numberStr, priceStr);

term = StrToInt(termStr);

temp = StrToInt(tempStr);

}

};

const int K = 100;

TovarProd prod[K];

TovarProm prom[K];

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button5Click(TObject \*Sender)

{

if(OpenDialog1->Execute())

{

String filename = OpenDialog1->FileName;

TStringList \*list = new TStringList();

try

{

try {

list->LoadFromFile(filename, TEncoding::UTF8);

}

catch (...) {

list->LoadFromFile(filename, TEncoding::GetEncoding(1251));

}

StringGrid1->RowCount = (StringGrid1->FixedRows > 0) ? 1 : 0;

i = 0; j = 0;

for(int k = 0; k < list->Count; k++)

{

String line = list->Strings[k].Trim();

if (line.IsEmpty()) continue;

TStringList \*parts = new TStringList();

parts->Delimiter = L' ';

parts->DelimitedText = line;

if(parts->Count == 3 && j < K)

{

prom[j].add\_grid(parts->Strings[0], parts->Strings[1], parts->Strings[2]);

prom[j].show();

j++;

StringGrid1->RowCount++;

}

else if(parts->Count == 5 && i < K)

{

prod[i].add\_grid(parts->Strings[0], parts->Strings[1], parts->Strings[2],

parts->Strings[3], parts->Strings[4]);

prod[i].show();

i++;

StringGrid1->RowCount++;

}

delete parts;

}

}

\_\_finally

{

delete list;

}

}

}

//---------------------------------------------------------------------------

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button2Click(TObject \*Sender)

{

ProductType = 1;

prom[j].add\_rec();

prom[j].show();

j++;

prom[j].clear();

StringGrid1->RowCount++;

}

//---------------------------------------------------------------------------

\_\_fastcall TForm1::TForm1(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button1Click(TObject \*Sender)

{

ProductType = 0;

prod[i].add\_rec();

prod[i].show();

i++;

prod[i].clear();

StringGrid1->RowCount++;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button3Click(TObject \*Sender)

{

if(i == 0 && j == 0) {

ShowMessage(L"Нет товаров!");

} else {

float sum = 0.0;

for (int e = 0; e < i; e++) {

sum += prod[e].get\_price();

}

for (int e = 0; e < j; e++) {

sum += prom[e].get\_price();

}

ShowMessage(L"Общая стоимость: " + FloatToStr(sum));

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::FormCreate(TObject \*Sender)

{

StringGrid1->FixedCols = 0;

StringGrid1->FixedRows = 0;

StringGrid1->ColCount = 5;

StringGrid1->RowCount = 0;

StringGrid1->ColWidths[0] = 150;

StringGrid1->ColWidths[1] = 110;

SaveDialog1->Filter = L"Текстовые файлы (\*.txt)|\*.txt|Все файлы (\*.\*)|\*.\*";

OpenDialog1->Filter = L"Текстовые файлы (\*.txt)|\*.txt|Все файлы (\*.\*)|\*.\*";

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button4Click(TObject \*Sender)

{

if(SaveDialog1->Execute())

{

String filename = SaveDialog1->FileName;

TStringList \*list = new TStringList();

for (int i = 0; i < StringGrid1->RowCount; i++)

{

StringGrid1->Rows[i]->Delimiter = L' ';

String line = StringGrid1->Rows[i]->DelimitedText;

list->Add(line);

}

list->SaveToFile(filename, TEncoding::UTF8);

delete list;

}

}

**Unit1.h**

////---------------------------------------------------------------------------

#ifndef Unit1H

#define Unit1H

//---------------------------------------------------------------------------

#include <System.Classes.hpp>

#include <Vcl.Controls.hpp>

#include <Vcl.StdCtrls.hpp>

#include <Vcl.Forms.hpp>

#include <Vcl.Dialogs.hpp>

#include <Vcl.ExtCtrls.hpp>

#include <Vcl.Grids.hpp>

#include <Vcl.Mask.hpp>

//---------------------------------------------------------------------------

class TForm1 : public TForm

{

\_\_published: // IDE-managed Components

TButton \*Button1;

TButton \*Button2;

TGroupBox \*GroupBox1;

TGroupBox \*GroupBox2;

TStringGrid \*StringGrid1;

TButton \*Button3;

TButton \*Button4;

TButton \*Button5;

TLabeledEdit \*LabeledEdit3;

TLabeledEdit \*LabeledEdit4;

TLabeledEdit \*LabeledEdit5;

TOpenDialog \*OpenDialog1;

TSaveDialog \*SaveDialog1;

TLabeledEdit \*LabeledEdit1;

TLabeledEdit \*LabeledEdit2;

TLabeledEdit \*LabeledEdit6;

TLabeledEdit \*LabeledEdit7;

TLabeledEdit \*LabeledEdit8;

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

void \_\_fastcall Button4Click(TObject \*Sender);

void \_\_fastcall Button5Click(TObject \*Sender);

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

void \_\_fastcall Button2Click(TObject \*Sender);

void \_\_fastcall Button3Click(TObject \*Sender);

private: // User declarations

public: // User declarations

\_\_fastcall TForm1(TComponent\* Owner);

};

//---------------------------------------------------------------------------

extern PACKAGE TForm1 \*Form1;

//---------------------------------------------------------------------------

**Вывод:** В рамках выполнения данной лабораторной работы было создано улучшенное приложение на основе предыдущего проекта, демонстрирующее эффективное применение принципов наследования в ООП и взаимодействие с компонентом StringGrid. Замена компонента Memo на StringGrid позволила более структурированно представлять данные о товарах, что улучшило пользовательский интерфейс и визуальное восприятие информации. Реализация механизмов сохранения данных в текстовый файл и их последующего считывания в StringGrid существенно расширила функциональность приложения, обеспечив сохранность информации между сеансами работы. Интеграция иерархии классов с компонентом StringGrid продемонстрировала эффективное взаимодействие объектно-ориентированной архитектуры с элементами пользовательского интерфейса VCL. Данная лабораторная работа способствовала углублению понимания принципов наследования в ООП и развитию практических навыков работы с компонентами визуального интерфейса в C++ Builder.