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 № 9**

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

**Тема: «Классы. Наследование при изначальной разработке программы»**

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

Проверил: к.т.н., доц. Семёнов А. А.

Москва, 2025г

**Классы. Наследование при изначальной разработке программы**

Лабораторная работа посвящена изучению классов в ООП на 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.

Используя известный по лекциям класс Tovar, в примере создаётся два производных от него класса: 1) TovarProd, добавляющий возможность хранить информацию о сроке хранения и температуре хранения продуктовых товаров; 2) TovarProm, позволяющий хранить информацию в соответствии с полями базового класса. Созданное визуальное приложение (Windows VCL Application) должно позволять: вводить информацию либо о продуктовых товарах, либо о промышленных товарах; выводить общую стоимость товаров, имеющихся на складе

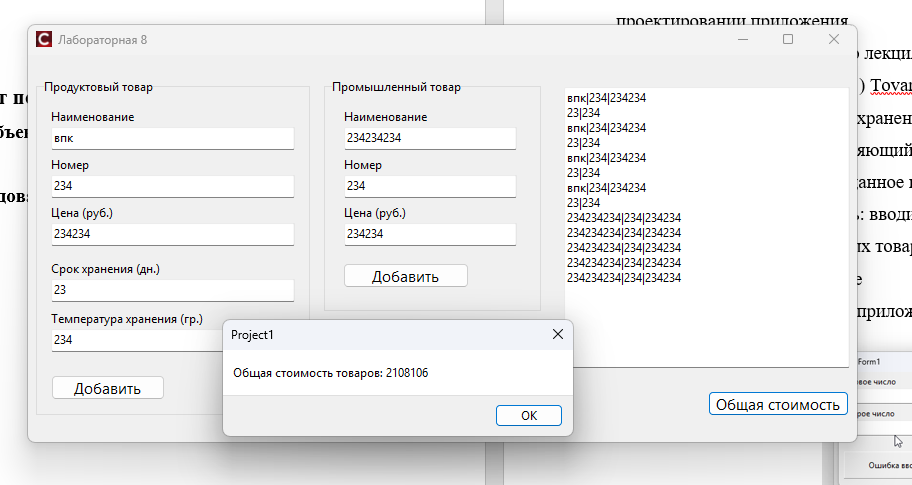
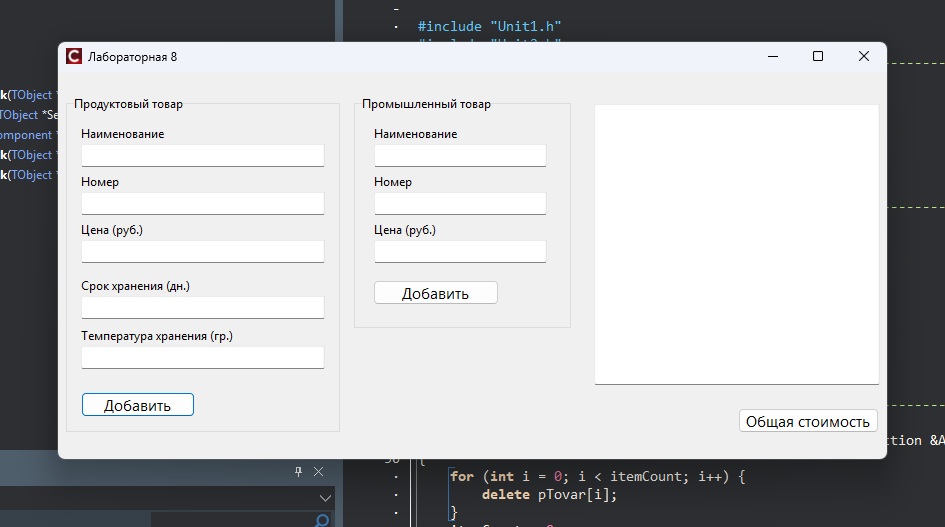


Рисунок 1. Работа программы

  
Рисунок 2. Начальное окно программы

**Листинг кода**

**Unit1.cpp**

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

#include <vcl.h>

#pragma hdrstop

#include "Unit1.h"

#include "Unit2.h"

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

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm1 \*Form1;

const int K = 200;

Tovar\* pTovar [K];

int itemCount = 0;

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

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

{

if (itemCount >= K) return;

ProductType = 1;

pTovar[itemCount] = new TovarProm();

pTovar[itemCount]->add\_rec();

pTovar[itemCount]->show();

itemCount++;

}

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

void \_\_fastcall TForm1::FormClose(TObject \*Sender, TCloseAction &Action)

{

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

delete pTovar[i];

}

itemCount = 0;

}

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

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

: TForm(Owner)

{

}

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

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

{

if (itemCount >= K) return;

ProductType = 0;

pTovar[itemCount] = new TovarProd();

pTovar[itemCount]->add\_rec();

pTovar[itemCount]->show();

itemCount++;

}

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

{

if (itemCount == 0) {

ShowMessage(L"Нет товаров!");

} else {

float sum = 0.0;

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

sum += pTovar[i]->get\_price();

}

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

}

}

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

**Unit1.h**

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

#ifndef Unit1H

#define Unit1H

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

#include <System.Classes.hpp>

#include <Vcl.Controls.hpp>

#include <Vcl.StdCtrls.hpp>

#include <Vcl.Forms.hpp>

#include <Vcl.ExtCtrls.hpp>

#include <Vcl.Mask.hpp>

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

class TForm1 : public TForm

{

\_\_published: // IDE-managed Components

TGroupBox \*GroupBox1;

TGroupBox \*GroupBox2;

TMemo \*Memo1;

TButton \*Button1;

TLabeledEdit \*LabeledEdit1;

TLabeledEdit \*LabeledEdit2;

TLabeledEdit \*LabeledEdit3;

TLabeledEdit \*LabeledEdit4;

TLabeledEdit \*LabeledEdit5;

TLabeledEdit \*LabeledEdit6;

TLabeledEdit \*LabeledEdit7;

TLabeledEdit \*LabeledEdit8;

TButton \*Button2;

TButton \*Button3;

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

void \_\_fastcall FormClose(TObject \*Sender, TCloseAction &Action);

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

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

private: // User declarations

public: // User declarations

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

};

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

extern PACKAGE TForm1 \*Form1;

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

#endif

**Unit2.cpp**

/ #include <vcl.h>

#pragma hdrstop

#include "Unit2.h"

#include "Unit1.h"

#pragma package(smart\_init)

int ProductType = 0;

float Tovar::get\_price() {

return price;

}

void Tovar::add\_rec() {

if (ProductType == 0) {

strcpy(name, AnsiString(Form1->LabeledEdit1->Text).c\_str());

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

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

}

if (ProductType == 1) {

strcpy(name, AnsiString(Form1->LabeledEdit6->Text).c\_str());

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

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

}

}

void Tovar::show() {

AnsiString s = AnsiString(name) + "|" + IntToStr(number) + "|" + FloatToStr(price);

Form1->Memo1->Lines->Add(s);

}

void TovarProd::add\_rec() {

Tovar::add\_rec();

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

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

}

void TovarProd::show() {

Tovar::show();

AnsiString s2 = IntToStr(term) + "|" + IntToStr(temp);

Form1->Memo1->Lines->Add(s2);

}

**Unit2.h**

/#ifndef Unit2\_H

#define Unit2\_H

const int N = 256;

extern int ProductType;

class Tovar {

protected:

char name[N];

int number;

float price;

public:

virtual void add\_rec();

virtual void show();

float get\_price();

virtual ~Tovar() {}

};

class TovarProd : public Tovar {

private:

int term;

int temp;

public:

void add\_rec();

void show();

~TovarProd () {};

};

class TovarProm : public Tovar {

public:

~TovarProm () {};

};

#endif

**Вывод:** В процессе выполнения лабораторной работы было разработано приложение, демонстрирующее применение принципа наследования в объектно-ориентированном программировании на C++. На основе базового класса Tovar созданы два производных класса - TovarProd и TovarProm, реализующие расширенную функциональность для различных категорий товаров. Разработанное приложение успешно демонстрирует преимущества наследования, позволяя эффективно работать с разными типами данных через общий интерфейс базового класса. Реализованная функциональность ввода информации о товарах разных типов и расчета общей стоимости товаров показывает практическое применение наследования в решении реальных задач. Применение принципов ООП, включая наследование, инкапсуляцию и полиморфизм, позволило создать гибкую, расширяемую и легко поддерживаемую архитектуру программы.