Министерство науки и высшего образования РФ

ФГБОУ ВО «Байкальский государственный университет»

Институт культуры, социальных коммуникаций и информационных технологий

Кафедра математических методов и цифровых технологий

УТВЕРЖДАЮ

Зав.кафедрой ММ д.э.н.,

Профессор\_\_\_\_\_С.С.Ованесян «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

**ПРОГРАММА ДЛЯ УПРАВЛЕНИЯ ИНФОРМАЦИЕЙ ПАРФЮМЕРНОГО МАГАЗИНА**

**Текст программы и результаты выполнения**

Руководитель, к.т.н., доцент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Т.И. Ведерникова

Исполнитель, студент гр.ИС-20-2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.С. Апасова

Иркутск – 2022

Оглавление

[Текст документируемой программы 3](#_Toc100921061)

[Результаты работы программы 19](#_Toc100921062)

[Описание процесса (методов и результатов) тестирования 22](#_Toc100921063)

[Обоснование и описание изменений проекта 25](#_Toc100921064)

# Текст документируемой программы

Классы:

Product.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

namespace ПИ

{

public class Product

{

public int id;

public string title;

public string type;

public string manufacturer;

public DateTime dateIn;

public decimal price;

public int amount;

public string edIzm;

public int timeRealize;

public int sales;

public Product(int id)

{

this.id = id;

}

public Product(string q, string w, string e, string r, string t, string y, string u, string i, string o, string p)

{

this.id = Convert.ToInt16(q);

this.title = w;

this.type = e;

this.manufacturer = r;

this.dateIn = Convert.ToDateTime(t);

this.price = Convert.ToDecimal(y);

this.amount = Convert.ToInt16(u);

this.edIzm = i;

this.timeRealize = Convert.ToInt16(o);

this.sales = Convert.ToInt16(p);

}

}

public class Products

{

public Dictionary<int,Product> allProduct;

public Products()

{

allProduct = new Dictionary<int, Product>();

allProduct = readProduct();

}

private Dictionary<int, Product> readProduct()

{

foreach (string line in File.ReadLines("base.txt", Encoding.UTF8))

{

string[] p = line.Split(new string[] { ";" }, StringSplitOptions.None);

Product pr = new Product(p[0], p[1], p[2], p[3], p[4], p[5], p[6], p[7], p[8], p[9]);

allProduct.Add(pr.id, pr);

}

return allProduct;

}

public void refreshProduct()

{

Product del=new Product(0);

foreach(Product p in allProduct.Values)

{

if(p.amount==p.sales)

{

del = p;

}

}

delProduct(del);

saveProduct();

}

public void addProduct(Product a)

{

allProduct.Add(a.id, a);

}

public void delProduct(Product a)

{

allProduct.Remove(a.id);

}

public void saveProduct()

{

StreamWriter f = new StreamWriter("base.txt", false, Encoding.UTF8);

foreach (Product p in allProduct.Values)

{

f.WriteLine($"{p.id};{p.title};{p.type};{p.manufacturer};{p.dateIn.ToShortDateString()};{p.price};{p.amount};{p.edIzm};{p.timeRealize};{p.sales}");

}

f.Close();

}

public int getMaxId()

{

int max = 0;

foreach(Product p in allProduct.Values)

{

if (max < p.id) max = p.id;

}

return max;

}

public Dictionary<int, Product> getList()

{

return allProduct;

}

public void updateList()

{

allProduct.Clear();

allProduct = readProduct();

}

}

}

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ПИ

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new MainForm());

}

}

}

Код форм:

MainForm.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

namespace ПИ

{

public partial class MainForm : Form

{

public MainForm()

{

InitializeComponent();

loadProduct();

}

Products products;

Dictionary<int, Product> list;

int idchange;

private void loadProduct()

{

dataGridView1.ReadOnly = true;

dataGridView2.ReadOnly = true;

toolStripStatusLabel2.Text= "Список товаров успешно загружен";

comboBox1.Items.Add("Парфюмерная вода");

comboBox1.Items.Add("Туалетная вода");

products = new Products();

list= products.getList();

int r = 0;

foreach(Product p in list.Values)

{

fill(p, r);

r++;

}

}

private void fill(Product p, int r)

{

dataGridView1.Rows.Add();

dataGridView1[0, r].Value = p.id;

dataGridView1[1, r].Value = p.title;

dataGridView1[2, r].Value = p.type;

dataGridView1[3, r].Value = p.manufacturer;

dataGridView1[4, r].Value = p.dateIn;

dataGridView1[5, r].Value = p.price;

dataGridView1[6, r].Value = p.amount;

dataGridView1[7, r].Value = p.edIzm;

dataGridView1[8, r].Value = p.timeRealize;

dataGridView1[9, r].Value = p.sales;

}

private void fillsalses(Product p, int r)

{

bool find = false;

foreach(DataGridViewRow row in dataGridView2.Rows)

{

if(Convert.ToInt16(row.Cells[0].Value)==idchange)

{

int count = Convert.ToInt16(dataGridView2[3, row.Index].Value);

dataGridView2[3, row.Index].Value = count + 1;

dataGridView2[4, row.Index].Value = Convert.ToInt16(dataGridView2[4, row.Index].Value) - 1;

find = true;

}

}

if (find == false)

{

dataGridView2.Rows.Add();

dataGridView2[0, r].Value = p.id;

dataGridView2[1, r].Value = p.title;

dataGridView2[2, r].Value = p.price;

dataGridView2[3, r].Value = 1;

dataGridView2[4, r].Value = p.amount - p.sales-Convert.ToInt16(dataGridView2[4,r].Value)-1;

}

if (checkBox1.Checked)

{

textBox5.Text = Convert.ToString(Convert.ToInt32(textBox5.Text) + Convert.ToInt16(p.price)\*0.9);

}

else

{

textBox5.Text = Convert.ToString(Convert.ToInt32(textBox5.Text) + p.price);

}

}

private void sort()

{

bool a = (textBox1.Text == string.Empty);

bool b = (textBox2.Text == string.Empty);

bool price1 =(textBox3.Text==string.Empty);

bool price2 = (textBox4.Text==string.Empty);

bool date = (dateTimePicker1.Value.ToShortDateString() == "01.01.2000");

bool type = (comboBox1.SelectedIndex==-1);

bool count1 = (numericUpDown1.Value == 0);

bool count2 = (numericUpDown2.Value == 0);

List<Product> sorted1 =new List<Product>();

List<Product> sorted2 =new List<Product>();

List<Product> sorted3 = new List<Product>();

List<Product> sorted4 = new List<Product>();

if(a==false)

{

dataGridView1.Rows.Clear();

int r = 0;

foreach(Product p in list.Values)

{

if(p.title.ToLower().Contains(textBox1.Text.ToLower()))

{

fill(p,r );

r++;

}

}

b = true; price1 = true; price2 = true; date = true;type = true;count1 = true;count2 = true;

}

if (b==false)

{

dataGridView1.Rows.Clear();

foreach(Product p in list.Values)

{

if(p.manufacturer.ToLower().Contains(textBox2.Text.ToLower()))

{

sorted1.Add(p);

}

}

}

if(price1==false&&price2==false)

{

dataGridView1.Rows.Clear();

foreach(Product p in list.Values)

{

if(p.price>=Convert.ToDecimal(textBox3.Text)&&p.price<=Convert.ToDecimal(textBox4.Text))

{

sorted2.Add(p);

}

}

}

if(price1==false&&price2==true)

{

dataGridView1.Rows.Clear();

foreach(Product p in list.Values)

{

if(p.price>= Convert.ToDecimal(textBox3.Text))

{

sorted2.Add(p);

}

}

}

if(price1==true&&price2==false)

{

dataGridView1.Rows.Clear();

foreach(Product p in list.Values)

{

if(p.price<= Convert.ToDecimal(textBox4.Text))

{

sorted2.Add(p);

}

}

}

if(date==false&&type==false)

{

dataGridView1.Rows.Clear();

foreach(Product p in list.Values)

{

if(p.dateIn.ToShortDateString()==Convert.ToDateTime(dateTimePicker1.Value).ToShortDateString()&&p.type==comboBox1.SelectedItem.ToString())

{

sorted3.Add(p);

}

}

}

if(date==false&&type==true)

{

dataGridView1.Rows.Clear();

foreach(Product p in list.Values)

{

if(p.dateIn.ToShortDateString()==Convert.ToDateTime(dateTimePicker1.Value).ToShortDateString())

{

sorted3.Add(p);

}

}

}

if(date==true&&type==false)

{

dataGridView1.Rows.Clear();

foreach(Product p in list.Values)

{

if(p.type==comboBox1.SelectedItem.ToString())

{

sorted3.Add(p);

}

}

}

if(count1==false&&count2==false)

{

dataGridView1.Rows.Clear();

foreach(Product p in list.Values)

{

if(p.amount>=numericUpDown1.Value&&p.amount<=numericUpDown2.Value)

{

sorted4.Add(p);

}

}

}

if(count1==false&&count2==true)

{

dataGridView1.Rows.Clear();

foreach(Product p in list.Values)

{

if(p.amount>=numericUpDown1.Value)

{

sorted4.Add(p);

}

}

}

if(count1==true&&count2==false)

{

dataGridView1.Rows.Clear();

foreach(Product p in list.Values)

{

if(p.amount<=numericUpDown2.Value)

{

sorted4.Add(p);

}

}

}

List<Product> itog = new List<Product>();

if(sorted1.Count==0)

{

if(sorted2.Count==0&&sorted3.Count==0&&sorted4.Count!=0)

{

itog = sorted4;

}

if(sorted2.Count==0&&sorted3.Count!=0&&sorted4.Count==0)

{

itog = sorted3;

}

if(sorted2.Count!=0&&sorted3.Count==0&&sorted4.Count==0)

{

itog = sorted2;

}

if(sorted2.Count!=0&&sorted3.Count!=0&&sorted4.Count==0)

{

itog = sorted2.Intersect(sorted3).ToList();

}

if(sorted2.Count!=0&sorted3.Count==0&&sorted4.Count!=0)

{

itog = sorted2.Intersect(sorted4).ToList();

}

if(sorted2.Count==0&&sorted3.Count!=0&&sorted4.Count!=0)

{

itog = sorted3.Intersect(sorted4).ToList();

}

if(sorted2.Count!=0&&sorted3.Count!=0&&sorted4.Count!=0)

{

itog = sorted2.Intersect(sorted3).Intersect(sorted4).ToList();

}

}

if(sorted1.Count!=0)

{

if(sorted2.Count==0&&sorted3.Count==0&&sorted4.Count==0)

{

itog = sorted1;

}

if (sorted2.Count == 0 && sorted3.Count == 0 && sorted4.Count != 0)

{

itog =sorted1.Intersect(sorted4).ToList();

}

if (sorted2.Count == 0 && sorted3.Count != 0 && sorted4.Count == 0)

{

itog =sorted1.Intersect(sorted3).ToList();

}

if (sorted2.Count != 0 && sorted3.Count == 0 && sorted4.Count == 0)

{

itog =sorted1.Intersect(sorted2).ToList();

}

if (sorted2.Count != 0 && sorted3.Count != 0 && sorted4.Count == 0)

{

itog =sorted1.Intersect(sorted2).Intersect(sorted3).ToList();

}

if (sorted2.Count != 0 & sorted3.Count == 0 && sorted4.Count != 0)

{

itog =sorted1.Intersect(sorted2).Intersect(sorted4).ToList();

}

if (sorted2.Count == 0 && sorted3.Count != 0 && sorted4.Count != 0)

{

itog =sorted1.Intersect(sorted3).Intersect(sorted4).ToList();

}

if (sorted2.Count != 0 && sorted3.Count != 0 && sorted4.Count != 0)

{

itog =sorted1.Intersect(sorted2).Intersect(sorted3).Intersect(sorted4).ToList();

}

}

int q = 0;

foreach(Product p in itog)

{

fill(p, q);

q++;

}

sorted1.Clear();

sorted2.Clear();

sorted3.Clear();

sorted4.Clear();

toolStripStatusLabel2.Text = "Результат поиска выданы";

}

private void newLoad()

{

products.refreshProduct();

dataGridView1.Rows.Clear();

products.updateList();

list = products.getList();

int r = 0;

foreach (Product p in list.Values)

{

fill(p, r);

r++;

}

toolStripStatusLabel2.Text = "Список продуктов обновлен";

}

private void button2\_Click(object sender, EventArgs e)

{

sort();

}

private void button3\_Click(object sender, EventArgs e)

{

dataGridView1.Rows.Clear();

textBox1.Text = "";

textBox2.Text = "";

textBox3.Text = "";

textBox4.Text ="";

dateTimePicker1.Value= Convert.ToDateTime("01.01.2000");

comboBox1.SelectedIndex = -1;

numericUpDown1.Value =0;

numericUpDown2.Value =0;

int r = 0;

foreach (Product p in list.Values)

{

fill(p, r);

r++;

}

}

private void добавитьToolStripMenuItem\_Click(object sender, EventArgs e)

{

int new\_id = products.getMaxId()+1;

Product p = new Product(new\_id);

AddOrChangeProduct form = new AddOrChangeProduct(new\_id, "add", products, p);

form.StartPosition = FormStartPosition.CenterScreen;

form.Text = "Новый продукт";

form.FormClosing += new FormClosingEventHandler(formclose);

form.ShowDialog();

}

void formclose( object sender , FormClosingEventArgs e)

{

newLoad();

}

private void изменитьToolStripMenuItem\_Click(object sender, EventArgs e)

{

Product p = list.FirstOrDefault(t => t.Value.id == idchange).Value;

AddOrChangeProduct form = new AddOrChangeProduct(idchange, "change", products,p);

form.StartPosition = FormStartPosition.CenterScreen;

form.Text = $"Изменение продукта № {idchange}";

form.FormClosing += new FormClosingEventHandler(formclose);

form.ShowDialog();

}

private void addinsale()

{

int k = dataGridView2.RowCount-1;

Product p = list[idchange];

fillsalses(p, k);

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

if (e.ColumnIndex == 10)

{

int t = e.RowIndex;

idchange = Convert.ToInt16(dataGridView1[0,t].Value);

addinsale();

toolStripStatusLabel2.Text = "Товар добавлен в чек";

}

else

{

int t = e.RowIndex;

idchange = Convert.ToInt16(dataGridView1[0, t].Value);

toolStripStatusLabel2.Text = "Товар для изменения выбран";

}

}

private void ПоДатеСначалаToolStripMenuItem\_Click(object sender, EventArgs e)

{

List<Product> sorted = list.Values.ToList();

sorted = sorted.OrderBy(x => x.dateIn).ToList();

dataGridView1.Rows.Clear();

int r = 0;

foreach(Product p in sorted)

{

fill(p, r);

r++;

}

toolStripStatusLabel2.Text = "Товары успешно отсортированы";

}

private void отПоследнихКПервымToolStripMenuItem\_Click(object sender, EventArgs e)

{

List<Product> sorted = list.Values.ToList();

sorted = sorted.OrderByDescending(x => x.dateIn).ToList();

dataGridView1.Rows.Clear();

int r = 0;

foreach(Product p in sorted)

{

fill(p, r);

r++;

}

toolStripStatusLabel2.Text = "Товары успешно отсортированы";

}

private void отНаименьшейСтоимостиToolStripMenuItem\_Click(object sender, EventArgs e)

{

List<Product> sorted = list.Values.ToList();

sorted = sorted.OrderBy(x => x.price).ToList();

dataGridView1.Rows.Clear();

int r = 0;

foreach(Product p in sorted)

{

fill(p, r);

r++;

}

toolStripStatusLabel2.Text = "Товары успешно отсортированы";

}

private void отНаиболшейСтоимостиToolStripMenuItem\_Click(object sender, EventArgs e)

{

List<Product> sorted = list.Values.ToList();

sorted = sorted.OrderByDescending(x => x.price).ToList();

dataGridView1.Rows.Clear();

int r = 0;

foreach(Product p in sorted)

{

fill(p, r);

r++;

}

toolStripStatusLabel2.Text = "Товары успешно отсортированы";

}

private void сброситьToolStripMenuItem\_Click(object sender, EventArgs e)

{

dataGridView1.Rows.Clear();

int r = 0;

foreach(Product p in list.Values)

{

fill(p, r);

r++;

}

toolStripStatusLabel2.Text = "Сортировки сброшены";

}

private void discount()

{

if(checkBox1.Checked)

{

textBox5.Text = Convert.ToString(Convert.ToInt32(textBox5.Text) \* 0.9);

}

else

{

textBox5.Text = Convert.ToString(Convert.ToInt32(textBox5.Text) / 0.9);

}

}

private void checkBox1\_CheckedChanged(object sender, EventArgs e)

{

discount();

}

private void dataGridView2\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

if(e.ColumnIndex==5)

{

int r = e.RowIndex;int c = e.ColumnIndex;

if(Convert.ToInt16(dataGridView2[3,r].Value)==1)

{

if(checkBox1.Checked)

{

textBox5.Text = Convert.ToString(Convert.ToInt16(textBox5.Text) - Convert.ToInt16(dataGridView2[2, r].Value) \* 0.9);

}

else

{

textBox5.Text = Convert.ToString(Convert.ToInt16(textBox5.Text) - Convert.ToInt16(dataGridView2[2, r].Value));

}

dataGridView2.Rows.RemoveAt(r);

}

/\* if(Convert.ToString(dataGridView2[0, r].Value)==string.Empty)

{

MessageBox.Show("Нельзя удалить несуществующий товар");

}\*/

else

{

dataGridView2[3, r].Value = Convert.ToInt16(dataGridView2[3, r].Value) - 1;

dataGridView2[4, r].Value = Convert.ToInt16(dataGridView2[4, r].Value) + 1;

if (checkBox1.Checked)

{

textBox5.Text = Convert.ToString(Convert.ToInt16(textBox5.Text) - Convert.ToInt16(dataGridView2[2, r].Value) \* 0.9);

}

else

{

textBox5.Text = Convert.ToString(Convert.ToInt16(textBox5.Text) - Convert.ToInt16(dataGridView2[2, r].Value));

}

}

}

}

private void button1\_Click(object sender, EventArgs e)

{

foreach(DataGridViewRow dgvr in dataGridView2.Rows)

{

if(Convert.ToInt16(dgvr.Cells[4].Value)<0)

{

MessageBox.Show("Вы не можете продать больше товара, чем имеется в наличии. Пожалуйста, проверьте чек", "Предупреждение");

}

else

{

Dictionary<int, int> sales = new Dictionary<int, int>();

int k = dataGridView2.Rows.Count - 1;

for (int i = 0; i < k; i++)

{

sales.Add(Convert.ToInt16(dataGridView2[0, i].Value), Convert.ToInt16(dataGridView2[3, i].Value));

}

int u = 0;

foreach (Product p in list.Values)

{

if (u != k)

{

foreach (var v in sales)

{

if (p.id == v.Key)

{

p.sales += v.Value;

u++;

}

}

}

else

{

break;

}

}

if (checkBox2.Checked)

{

int c = dataGridView2.RowCount - 1;

string path = $@"Продажи\{ DateTime.Now.ToShortDateString()}.txt";

FileInfo fileinfo = new FileInfo(path);

if (fileinfo.Exists)

{

StreamWriter file = new StreamWriter(path, true);

string text = $"Время:{DateTime.Now.ToShortTimeString()}";

file.WriteLine(text);

for (int i = 0; i < c; i++)

{

text = $"Номер товара:{dataGridView2[0, i].Value}, наименование:{dataGridView2[1, i].Value} , цена:{dataGridView2[2, i].Value}, количество:{dataGridView2[3, i].Value}";

file.WriteLine(text);

}

if (checkBox1.Checked)

{

text = $"Итоговая сумма чека с учетом скидки:{textBox5.Text}";

file.WriteLine(text);

}

else

{

text = $"Итоговая сумма чека:{textBox5.Text}";

file.WriteLine(text);

}

file.Close();

}

else

{

StreamWriter file = new StreamWriter(path, true);

string text = $"Время:{DateTime.Now.ToShortTimeString()}";

file.WriteLine(text);

for (int i = 0; i < c; i++)

{

text = $"Номер товара:{dataGridView2[0, i].Value}, наименование:{dataGridView2[1, i].Value} , цена:{dataGridView2[2, i].Value}, количество:{dataGridView2[3, i].Value}";

file.WriteLine(text);

}

if (checkBox1.Checked)

{

text = $"Итоговая сумма чека с учетом скидки:{textBox5.Text}";

file.WriteLine(text);

}

else

{

text = $"Итоговая сумма чека:{textBox5.Text}";

file.WriteLine(text);

}

file.Close();

}

}

textBox5.Text = "";

MessageBox.Show("Продажа успешно завершена.");

dataGridView2.Rows.Clear();

products.saveProduct();

toolStripStatusLabel2.Text = "Успешно продано";

newLoad();

}

}

}

private void создатьОтчетToolStripMenuItem\_Click(object sender, EventArgs e)

{

int c = list.Count();

Report form = new Report(c, products);

form.StartPosition = FormStartPosition.CenterScreen;

form.ShowDialog();

}

}

}

Report.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ПИ

{

public partial class Report : Form

{

int count;

Products prod;

List<Product> list = new List<Product>();

public Report(int c, Products p)

{

prod = p;

count = c;

InitializeComponent();

load();

}

private void load()

{

list = prod.getList().Values.ToList();

numericUpDown1.Maximum = count;

comboBox2.Items.Add("Парфюмерная вода");

comboBox2.Items.Add("Туалетная вода");

comboBox4.Items.Add("Парфюмерная вода");

comboBox4.Items.Add("Туалетная вода");

List<string> manuf = new List<string>();

foreach(Product p in list)

{

manuf.Add(p.manufacturer);

}

manuf = manuf.Distinct().ToList();

foreach(string i in manuf)

{

comboBox3.Items.Add(i);

comboBox5.Items.Add(i);

}

}

private void button1\_Click(object sender, EventArgs e)

{

if(radioButton1.Checked==false&&radioButton2.Checked==false&&radioButton3.Checked==false&&radioButton3.Checked==false)

{

MessageBox.Show("Пожалуйста, выберите тип необходимого отчета");

}

else

{

List<Product> report = list;

int y = Convert.ToInt16(numericUpDown1.Value);

if (radioButton1.Checked)//Самые продаваемые товары

{

string text = "";

report = report.OrderByDescending(x => x.sales).ToList();

for (int i=0; i<y; i++)

{

text += $"{report[i].title}\n";

}

MessageBox.Show(text, $"Топ-{y} продаваемых товаров");

}

if(radioButton2.Checked)//самые непопулярные товары

{

string text = "";

report = report.OrderBy(x => x.sales).ToList();

for(int i=0;i<y;i++)

{

text += $"{report[i].title}\n";

}

MessageBox.Show(text, $"Топ-{y}самых непопулярных товаров");

}

if(radioButton3.Checked)//Самые дорогие товары

{

string text = "";

report = report.OrderByDescending(x => x.price).ToList();

for(int i=0;i<y;i++)

{

text += $"{report[i].title}\n";

}

MessageBox.Show(text, $"Топ-{y} самых дорогих товаров");

}

if(radioButton4.Checked)//самые недорогие товары

{

string text = "";

report = report.OrderBy(x => x.price).ToList();

for (int i=0;i<y;i++)

{

text += $"{report[i].title}\n";

}

MessageBox.Show(text, $"Топ-{y} самых недорогих товаров");

}

}

}

private void button4\_Click(object sender, EventArgs e)//по типу товара

{

string type = comboBox2.SelectedItem.ToString();

decimal minprice=list[0].price;

decimal avgprice=0;

int allcount = 0;

foreach(Product p in prod.getList().Values)

{

if (p.type == type)

{

if (p.price < minprice)

{

minprice = p.price;

}

avgprice += p.price;

allcount++;

}

}

textBox6.Text =Convert.ToString(minprice);

textBox7.Text = Convert.ToString(avgprice / allcount);

}

private void button5\_Click(object sender, EventArgs e)

{

string man = comboBox3.SelectedItem.ToString();

decimal avgmonth = 0;

int count = 0;

int allcount = 0;

foreach(Product p in list)

{

if(p.manufacturer==man)

{

avgmonth += p.timeRealize;

count += p.amount - p.sales;

allcount++;

}

}

textBox9.Text = Convert.ToString(avgmonth / allcount);

textBox10.Text =Convert.ToString(count);

}

private void button6\_Click(object sender, EventArgs e)

{

string type = comboBox4.SelectedItem.ToString();

string man = comboBox5.SelectedItem.ToString();

decimal avgprice = 0;

int count = 0;

foreach(Product p in list)

{

if(p.type==type&&p.manufacturer==man)

{

avgprice += p.price;

count++;

}

}

textBox11.Text = Convert.ToString(avgprice / count);

}

}

}

AddOrChangeProduct.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ПИ

{

public partial class AddOrChangeProduct : Form

{

string check;

int newID;

public AddOrChangeProduct(int id, string ch, Products p, Product produ)

{

InitializeComponent();

check = ch;

newID = id;

load(p, produ);

}

Products products;

Product prod;

private void load(Products p, Product produ)

{

products = p;

prod = produ;

comboBox1.Items.Add("Туалетная вода");

comboBox1.Items.Add("Парфюмерная вода");

if (check=="add")

{

dateTimePicker1.Value = DateTime.Now;

}

if(check=="change")

{

dateTimePicker1.Enabled = false;

dateTimePicker1.Value = prod.dateIn;

numericUpDown1.Value = prod.price;

numericUpDown2.Value = prod.amount;

numericUpDown3.Value = prod.timeRealize;

textBox1.Text = prod.title;

textBox2.Text = prod.manufacturer;

comboBox1.SelectedItem = prod.type;

}

}

private void button1\_Click(object sender, EventArgs e)

{

if (textBox1.Text == string.Empty || comboBox1.SelectedIndex == -1 || textBox2.Text == string.Empty || numericUpDown1.Value == 0 || numericUpDown2.Value == 0||numericUpDown3.Value==0)

{

MessageBox.Show("Пожалуйста, заполните все поля для добавления товара");

}

else

{

if (check == "add")

{

prod.title = textBox1.Text;

prod.type = comboBox1.SelectedItem.ToString();

prod.manufacturer = textBox2.Text;

prod.dateIn = dateTimePicker1.Value;

prod.price = numericUpDown1.Value;

prod.amount = Convert.ToInt16(numericUpDown2.Value);

prod.timeRealize = Convert.ToInt16(numericUpDown3.Value);

prod.sales = 0;

prod.edIzm = "шт.";

products.addProduct(prod);

products.saveProduct();

MessageBox.Show("Продукт успешно добавлен");

button2.Enabled = false;

button1.Enabled = false;

}

if(check=="change")

{

prod.title = textBox1.Text;

prod.type = comboBox1.SelectedItem.ToString();

prod.manufacturer = textBox2.Text;

prod.dateIn = dateTimePicker1.Value;

prod.price = numericUpDown1.Value;

prod.amount = Convert.ToInt16(numericUpDown2.Value);

prod.timeRealize = Convert.ToInt16(numericUpDown3.Value);

prod.sales = 0;

prod.edIzm = "шт.";

products.saveProduct();

MessageBox.Show("Продукт успешно изменен");

button2.Enabled = false;

button1.Enabled = false;

}

}

}

private void button2\_Click(object sender, EventArgs e)

{

this.Close();

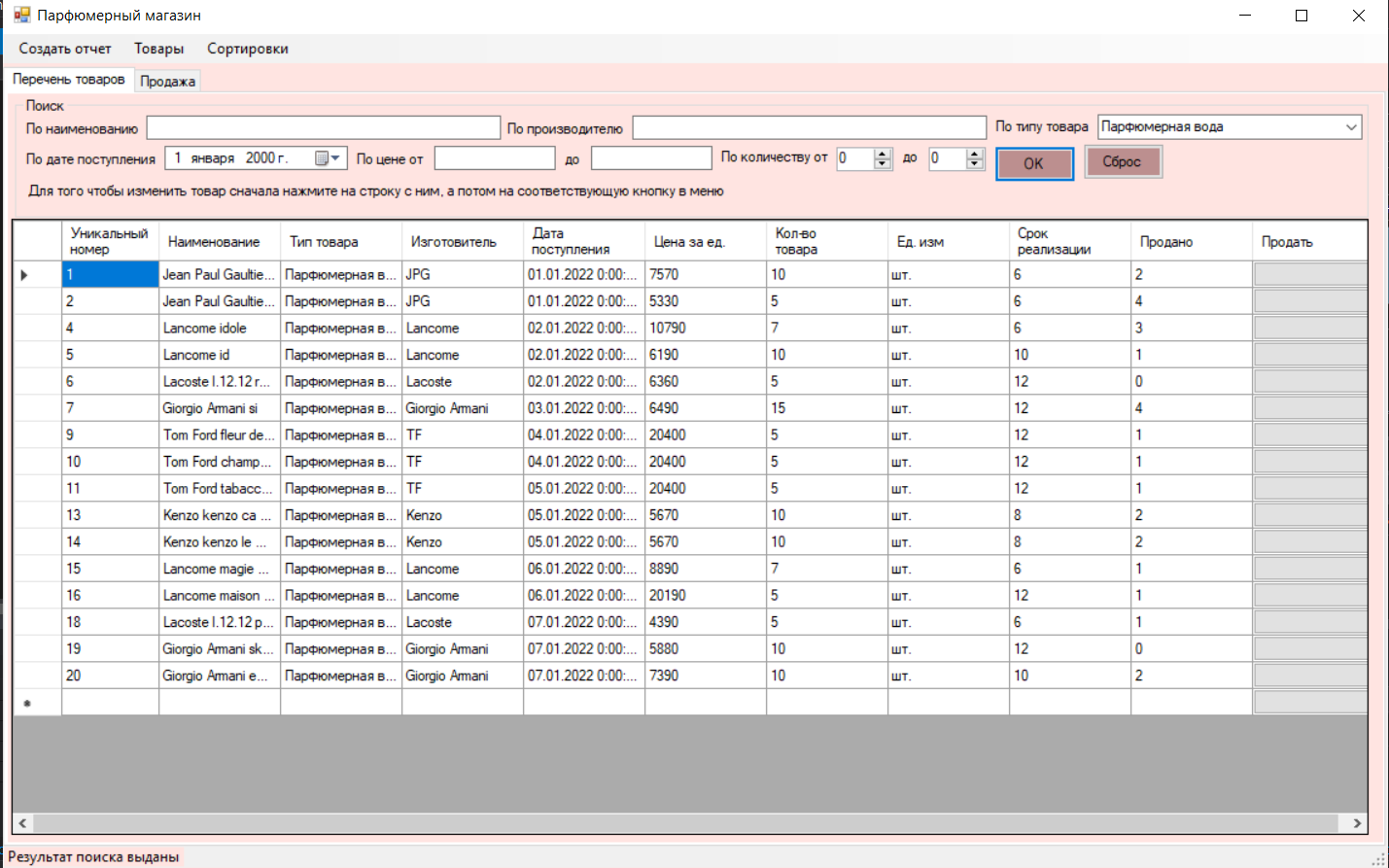
}

}

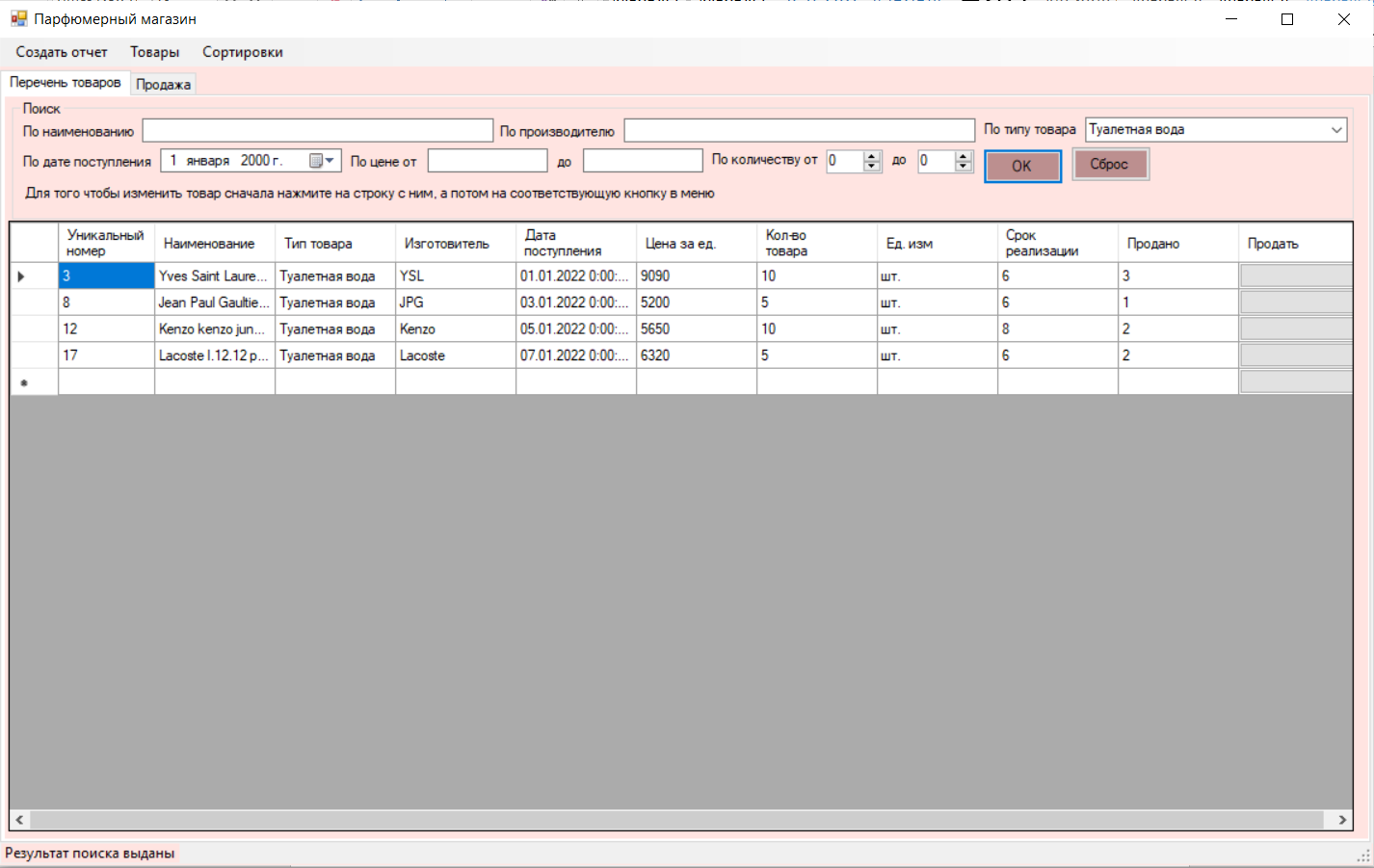
}

# Результаты работы программы

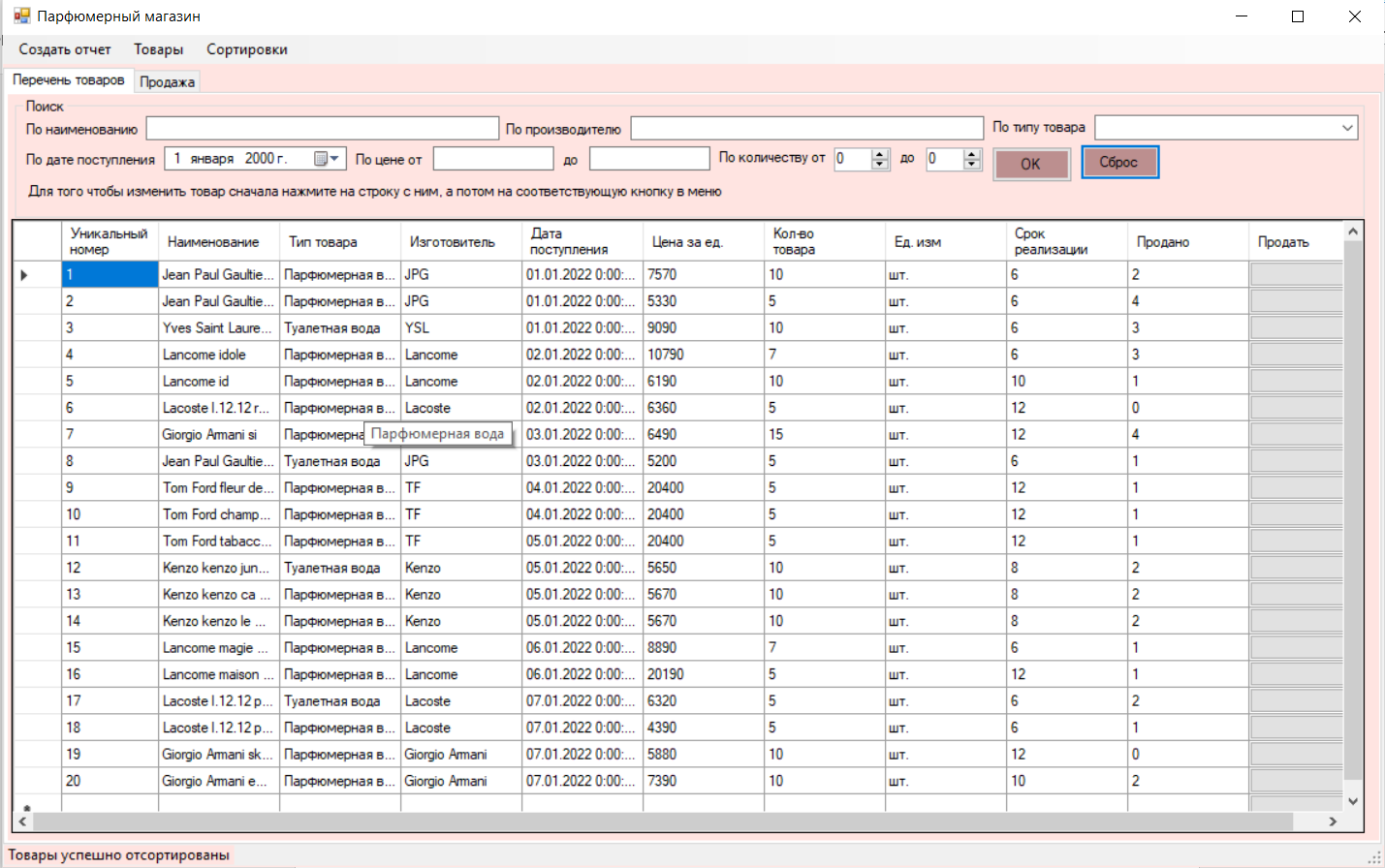
1. Программа должна проводить сортировку по виду товара и дате поступления в магазин



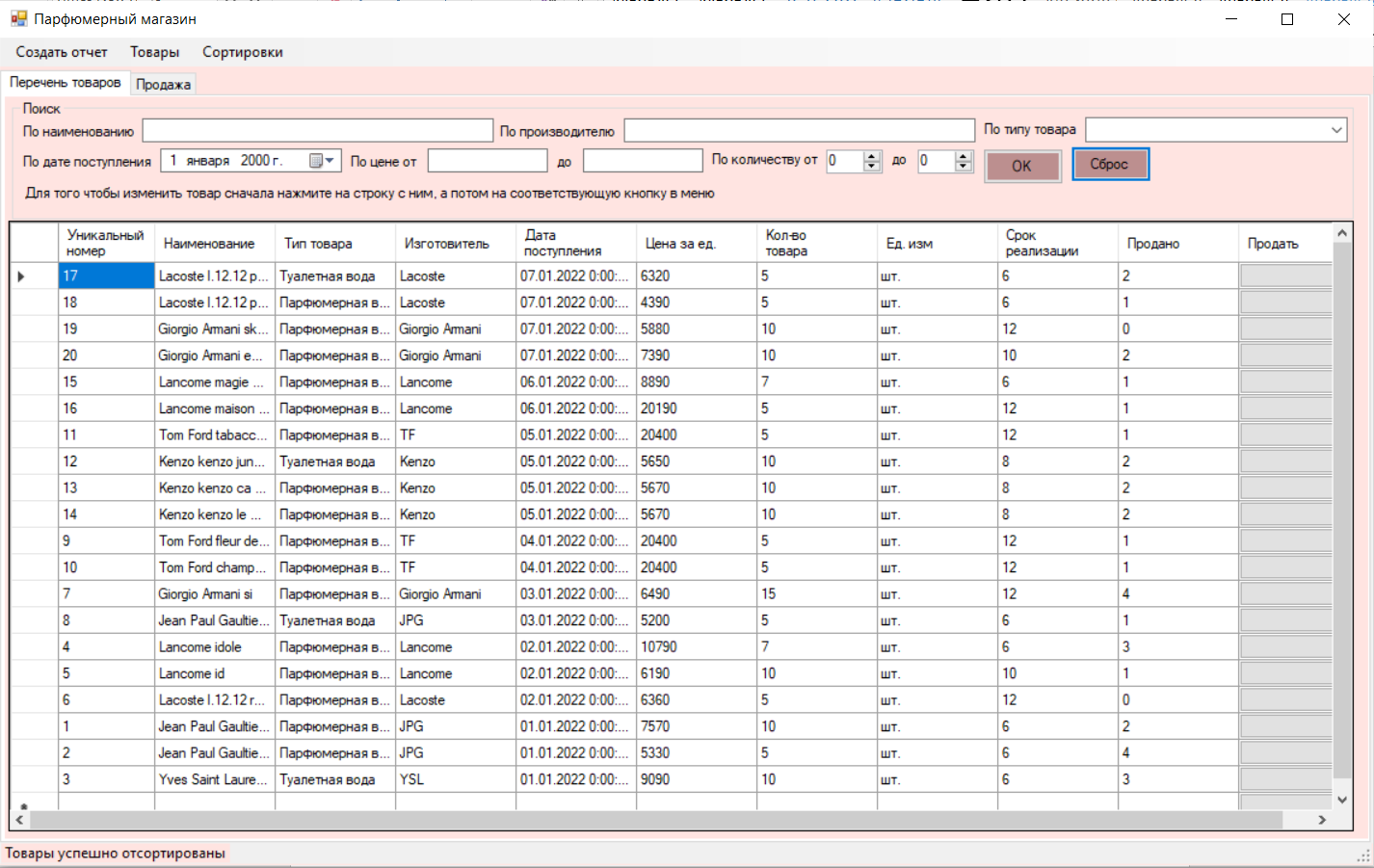
1. Сортировка по виду товара



. Сортировка по виду товара (2)

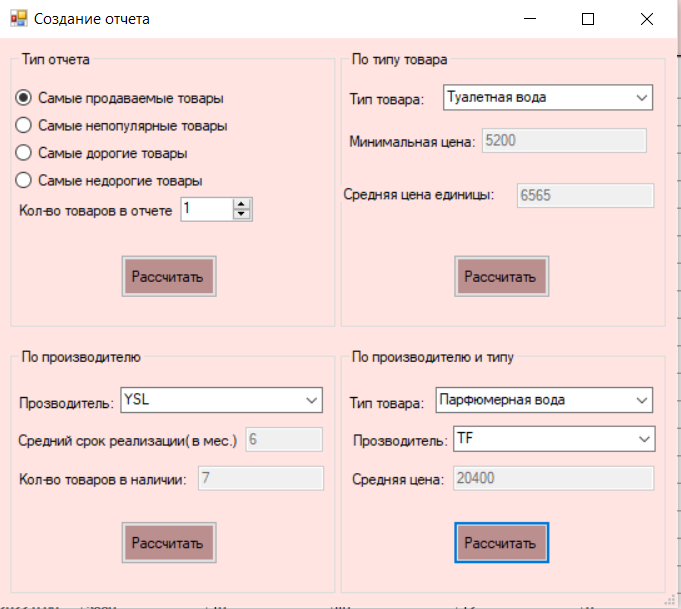


. Сортировка по дате с начала



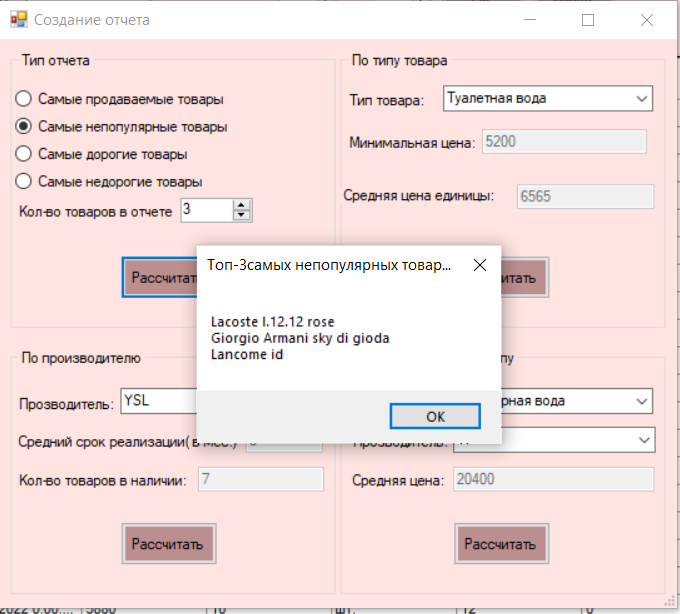
. Сортировка по дате с конца

1. Программа должна рассчитать: минимальную цену конкретного вида товара, средний срок реализации товара конкретной фирмы, количество наименований товара конкретной фирмы. Для каждого вида товара находить среднюю цену единицы и количество наименований, среднюю цену единицы товара в зависимости от его фирмы и вида.

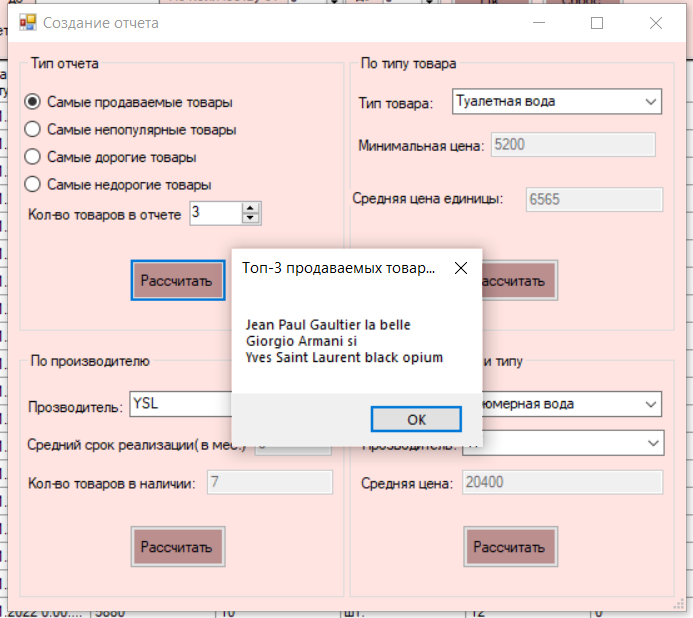


5. Результаты расчетов

1. Нахождение самых популярных/непопулярных товаров.



6. ТОП-3 непопулярных товара

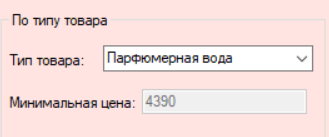


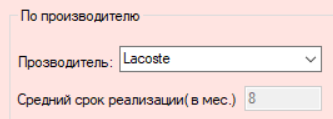
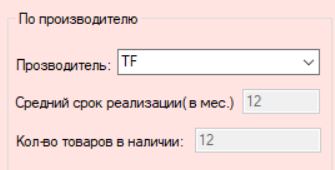
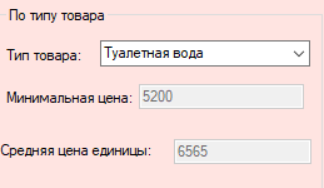
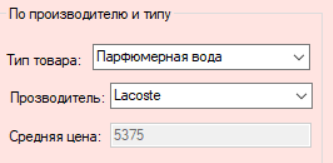
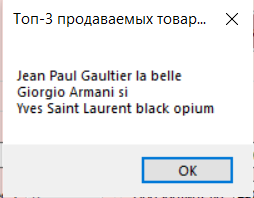
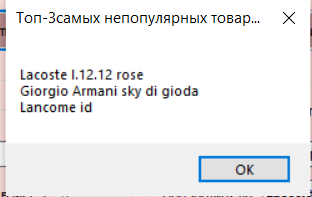
7 ТОП-3 популярных товара

# Описание процесса (методов и результатов) тестирования

Тестовый пример:

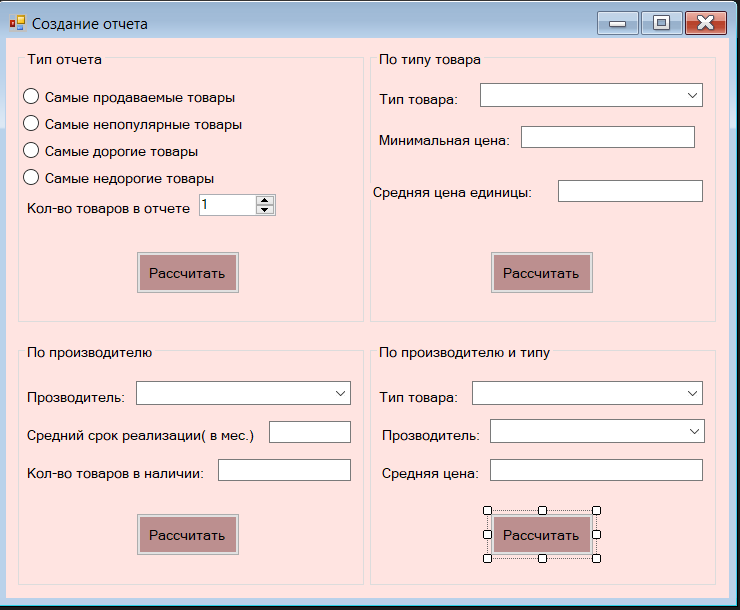
1. Вычисление минимальной цены конкретного типа товара.  
   Вход: список товаров, которые имеются в магазине (таблица1) и строка «Парфюмерная вода». Выход: 4390.00 рубля.

Результат работы программы:  


1. Вычисление среднего срока реализации товара конкретной фирмы-изготовителя. Вход: список товаров, которые имеются в магазине и строка «Lacoste». Выход: 8 месяцев.  
   Результат работы программы:  
   
2. Вычисление количества товаров конкретной фирмы изготовителя, которые сейчас имеются в наличии. Вход: список товаров, которые имеются в магазине (таблица 1) и строка «TF». Выход: 12 штук.   
   Результат работы программы:  
   
3. Вычисление средней цены конкретного вида товара. Вход: список товаров, которые имеются в магазине (таблица 1) и строка «Туалетная вода». Выход: 6565,00 рубля.   
   Результат работы программы:  
   
4. Вычисление средней цены товара в зависимости от его вида и фирмы-изготовителя. Вход: список товаров, которые имеются в магазине (таблица 1), строка «Парфюмерная вода» и строка «Lacoste». Выход: 5375,00 рубля.   
   Результат работы программы:  
   
5. Вычисление ТОП-n проданных товаров. Вход: список товаров, которые имеются в магазине (таблица 1) и число n=3. Выход: Giorgio Armani si, Jean Paul Gaultier la belle и Yves Saint Laurent black opium.   
   Результат работы программы:  
   
6. Вычисление ТОП-n непопулярных товаров. Вход: список товаров, которые имеются в магазине (таблица 1) и число n=3. Выход: Lacoste l.12.12 rose, Giorgio Armani sky di gioda и Lancome id.  
   Результат работы программы:  
   

# Обоснование и описание изменений проекта

1. Была изменена форма создания отчета. Были добавлены инструменты для расчета минимальной цены конкретного вида товара, среднего срока реализации товара конкретной фирмы, количества наименований товара конкретной фирмы. Так же были добавлены инструменты для нахождения средней цены единицы и количество наименований для каждого вида товара, средней цены единицы товара в зависимости от его фирмы и вида.



1. В форме создания/изменения товара была убрана возможность выбора единицы измерения товара, потому что штучный способ учета единиц продукции оказался наиболее удобным и практичным.

