

1. Цель работы

Ознакомление с основами объектно-ориентированного программирования на С#. Разработать графический интерфейс для проекта из первой или второй работы.

* 1. Задание

Создать базовый класс Supermarket со следующими полями: название, адрес, перечень товаров, включающий наименования и количество, менеджеры и кассира. Кассир реализует операцию покупки конкретного товара посетителем, в результате чего должно измениться количество товара в перечне. Менеджер регулярно следит за перечнем товара и при необходимости дополняет количество до нормы. Все методы в базовом классе должны быть абстрактными. На основе этого класса создать 2 супермаркета с конкретными названиями, адресами, персоналом.

1. UML-диаграмма классов

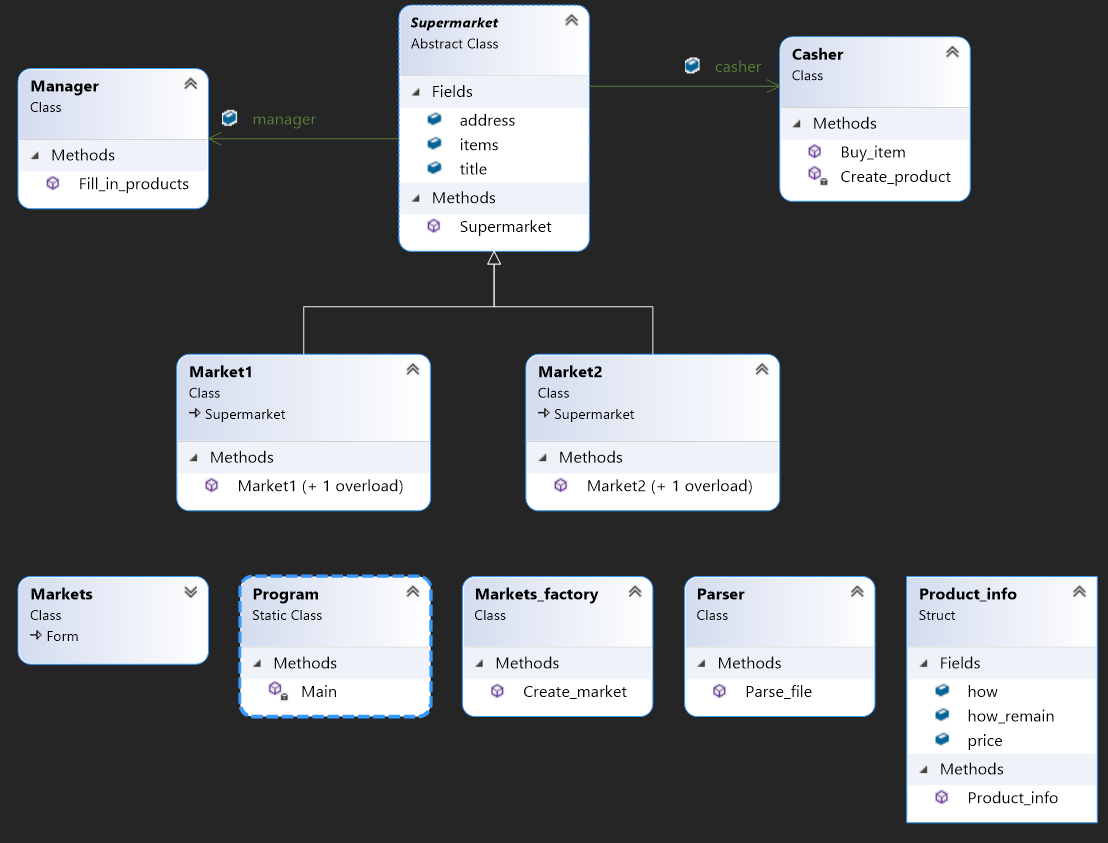


Рисунок 1 – Диаграмма классов

1. Графический интерфейс

На рисунках 1 и 2 изображено два интерактивных окна, созданных с помощью Windows Forms.

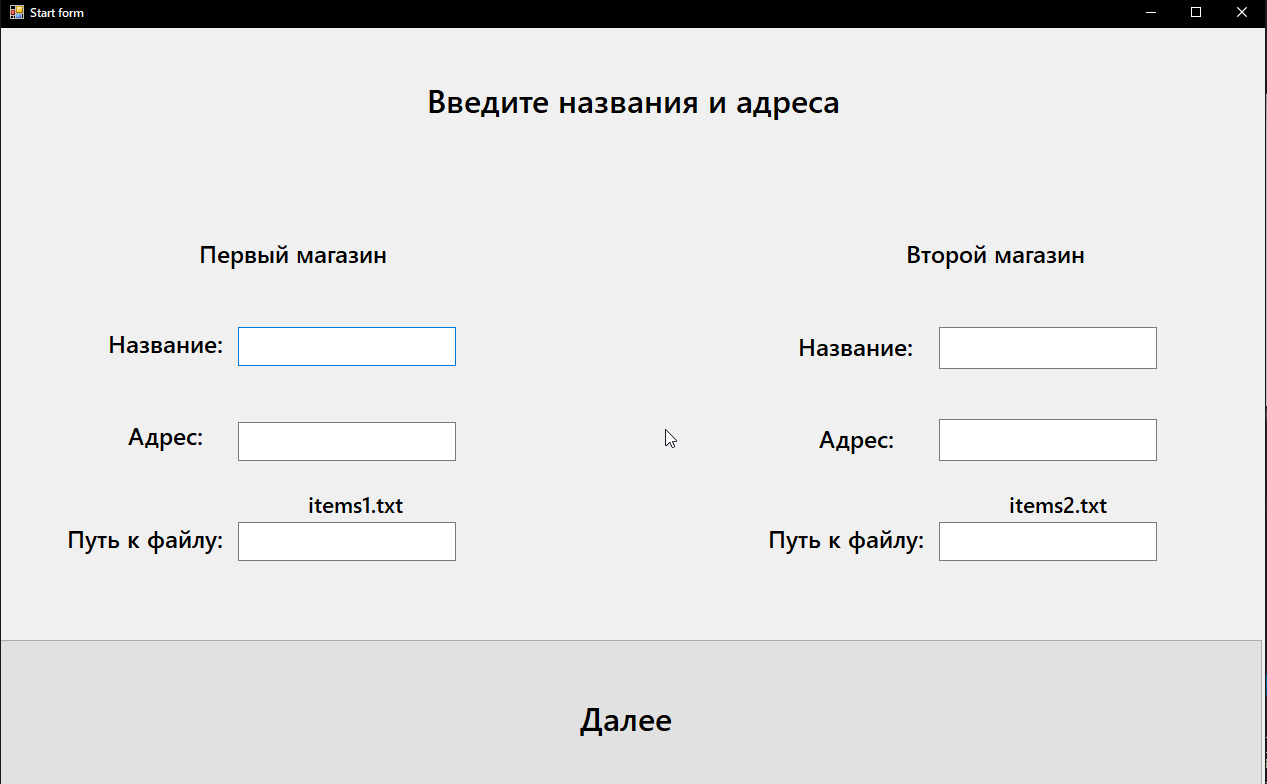


Рисунок 2 – Первое окно

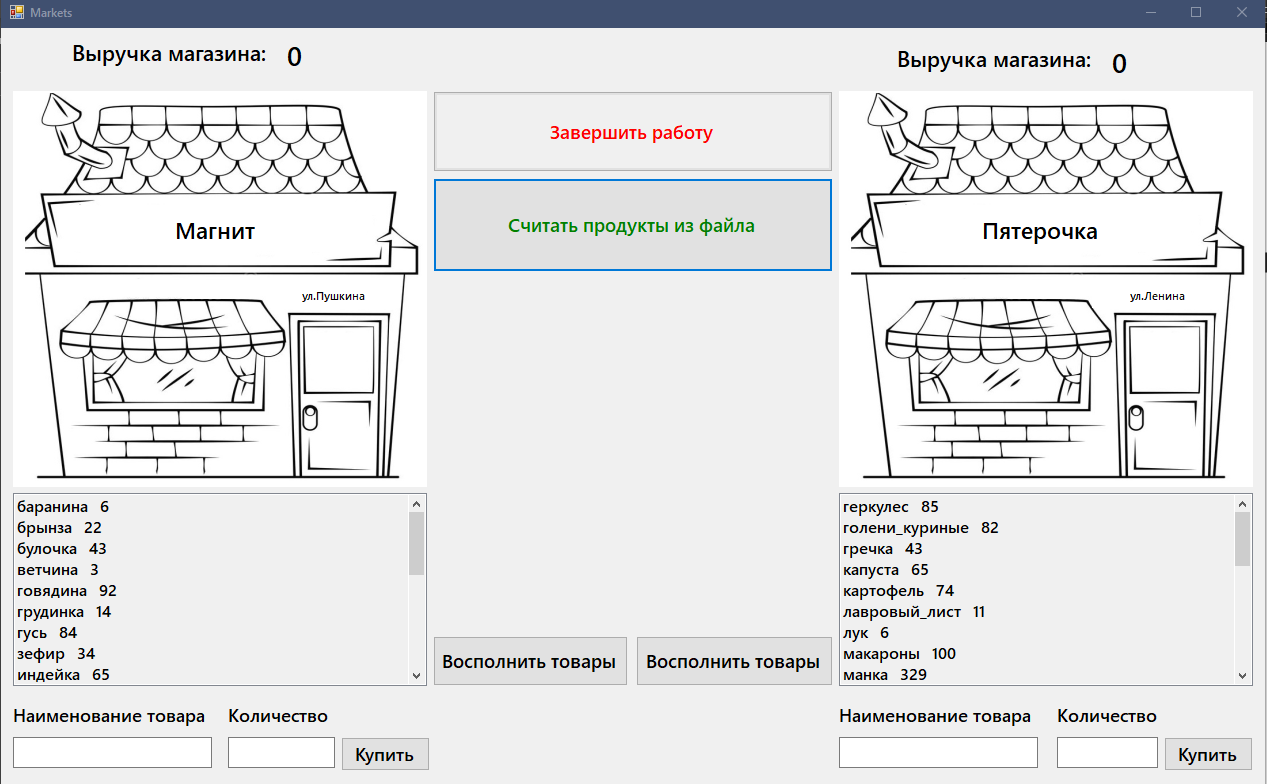


Рисунок 3 – Второе окно

1. Листинг программы
   1. Manager.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LAB\_6\_OOP

{

class Manager

{

public void Fill\_in\_products(Supermarket market)

{

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

foreach (string keyValue in market.items.Keys)

{

keys.Add(keyValue);

}

for(int i=0;i< market.items.Count();i++)

{

Product\_info temp = new Product\_info(market.items[keys[i]].price, market.items[keys[i]].how, market.items[keys[i]].how);

market.items[keys[i]] = temp;

}

}

}

}

* 1. Market1.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LAB\_6\_OcOP

{

class Market1 : Supermarket

{

public Market1():base("Магнит", "ул.Пушкина")

{}

public Market1(string title, string address) : base(title, address)

{ }

}

}

* 1. Market2.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LAB\_6\_OOP

{

class Market2 : Supermarket

{

public Market2() : base("Пятерочка", "ул.Ленина")

{ }

public Market2(string title, string address) : base(title, address)

{ }

}

}

* 1. Markets.cs

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Windows.Forms;

namespace LAB\_6\_OOP

{

public partial class Markets : Form

{

bool flag = false;

private Start\_form form1;

string path1;

string path2;

public Markets(string ft, string fa, string st, string sa, string path1, string path2, Start\_form form)

{

try

{

market1 = Markets\_factory.Create\_market(ft, fa, 1);

market2 = Markets\_factory.Create\_market(st, sa, 2);

form1 = form;

InitializeComponent();

first\_title.Text = market1.title;

first\_street\_label.Text = market1.address;

second\_title.Text = market2.title;

second\_street\_label.Text = market2.address;

this.path1 = path1;

this.path2 = path2;

}

catch (Exception ex)

{

Console.WriteLine("Exception: " + ex.Message);

}

}

private void Update\_list(ListBox list, Supermarket market)

{

list.Items.Clear();

foreach (KeyValuePair<string, Product\_info> keyValue in market.items)

{

list.Items.Add(keyValue.Key + " " + keyValue.Value.how\_remain);

}

}

private void end\_prog\_button\_MouseEnter(object sender, EventArgs e)

{

end\_prog\_button.BackColor = Color.Red;

end\_prog\_button.ForeColor = Color.Black;

}

private void end\_prog\_button\_MouseLeave(object sender, EventArgs e)

{

end\_prog\_button.BackColor = SystemColors.ButtonFace;

end\_prog\_button.ForeColor = Color.Red;

}

private void button3\_Click\_1(object sender, EventArgs e)

{

if (flag)

{

if (int.Parse(shop\_revenue\_1.Text) >= int.Parse(shop\_revenue\_2.Text))

{

MessageBox.Show("На этот раз первый магазин заработал больше!");

}

else

{

MessageBox.Show("На этот раз второй магазин заработал больше!");

}

}

form1.Close();

this.Close();

}

private void Replace\_by\_index(string index, int how\_remain, ListBox product\_list, string product\_title)

{

int i = product\_list.FindString(index);

product\_list.Items.RemoveAt(i);

product\_list.Items.Insert(i, product\_title + " " + how\_remain);

}

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

{

if (flag)

{

product\_title\_1.Text = product\_title\_1.Text.ToLower();

if (market1.items.ContainsKey(product\_title\_1.Text))

{

int count = market1.items[product\_title\_1.Text].how\_remain;

if (count < int.Parse(buy\_count\_1.Text))

{

MessageBox.Show("В магазине недостаточно товара, попробуйте ещё раз");

}

else

{

market1.casher.Buy\_item(market1, product\_title\_1.Text, buy\_count\_1.Text);

Replace\_by\_index(product\_title\_1.Text, market1.items[product\_title\_1.Text].how\_remain, product\_list\_1, product\_title\_1.Text);

Update\_revenue(shop\_revenue\_1, int.Parse(buy\_count\_1.Text), market1.items[product\_title\_1.Text].price);

Update\_list(product\_list\_1, market1);

}

}

else

{

MessageBox.Show("Данный магазин не имеет такого товара, попробуйте заново");

}

}

else

{

MessageBox.Show("Вы не считали товары из файла, нажмите на кнопку");

}

}

private void buy\_button2\_Click(object sender, EventArgs e)

{

if (flag)

{

product\_title\_2.Text = product\_title\_2.Text.ToLower();

if (market2.items.ContainsKey(product\_title\_2.Text))

{

int count = market2.items[product\_title\_2.Text].how\_remain;

if (count < int.Parse(buy\_count\_2.Text))

{

MessageBox.Show("В магазине недостаточно товара, попробуйте ещё раз");

}

else

{

market2.casher.Buy\_item(market2, product\_title\_2.Text, buy\_count\_2.Text);

Replace\_by\_index(product\_title\_2.Text, market2.items[product\_title\_2.Text].how\_remain, product\_list\_2, product\_title\_2.Text);

Update\_revenue(shop\_revenue\_2, int.Parse(buy\_count\_2.Text), market2.items[product\_title\_2.Text].price);

Update\_list(product\_list\_2, market2);

}

}

else

{

MessageBox.Show("Данный магазин не имеет такого товара, попробуйте заново");

}

}

else

{

MessageBox.Show("Вы не считали товары из файла, нажмите на кнопку");

}

}

private void Update\_revenue(Label label, int buy\_count, int price)

{

int temp = int.Parse(label.Text);

temp += buy\_count \* price;

label.Text = temp.ToString();

}

private void Read\_file\_Click(object sender, EventArgs e)

{

if (!flag)

{

try

{

Parser parser = new Parser();

parser.Parse\_file(market1, path1);

parser.Parse\_file(market2, path2);

flag = true;

Update\_list(product\_list\_1, market1);

Update\_list(product\_list\_2, market2);

}

catch (Exception ex)

{

Console.WriteLine("Exception: " + ex.Message);

MessageBox.Show("В дириктории отсутсвует файл");

Application.Exit();

}

}

}

private void call\_manager1\_Click(object sender, EventArgs e)

{

if (flag)

{

market1.manager.Fill\_in\_products(market1);

Update\_list(product\_list\_1, market1);

}

else

{

MessageBox.Show("Вы не считали товары из файла, нажмите на кнопку");

}

}

private void call\_manager2\_buton\_Click(object sender, EventArgs e)

{

if (flag)

{

market2.manager.Fill\_in\_products(market2);

Update\_list(product\_list\_2, market2);

}

else

{

MessageBox.Show("Вы не считали товары из файла, нажмите на кнопку");

}

}

private void Markets\_FormClosing(object sender, FormClosingEventArgs e)

{

}

private void Markets\_Load(object sender, EventArgs e)

{

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void textBox1\_TextChanged\_1(object sender, EventArgs e)

{

}

private void label1\_Click\_1(object sender, EventArgs e)

{

}

private void listBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void pictureBox3\_Click(object sender, EventArgs e)

{

}

private void textBox1\_TextChanged\_2(object sender, EventArgs e)

{

}

private void label1\_Click\_2(object sender, EventArgs e)

{

}

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

{

}

private void label3\_Click(object sender, EventArgs e)

{

}

private void pictureBox2\_Click(object sender, EventArgs e)

{

}

private void first\_title\_Click(object sender, EventArgs e)

{

}

private void second\_title\_Click(object sender, EventArgs e)

{

}

}

}

* 1. Markets\_factory.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LAB\_6\_OOP

{

class Markets\_factory

{

public static Supermarket Create\_market(string title, string address, int key)

{

if (title == "" && address == "")

{

switch (key)

{

case 1:

return new Market1();

case 2:

return new Market2();

default:

throw new Exception("Не удалось создать объект");

}

}

else

{

switch (key)

{

case 1:

return new Market1(title, address);

case 2:

return new Market2(title, address);

default:

throw new Exception("Не удалось создать объект");

}

}

}

}

}

* 1. Parser.cs

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LAB\_6\_OOP

{

class Parser

{

public void Parse\_file(Supermarket market,string path)

{

string[] items;

string[] split;

if (File.Exists(path))

{

items = File.ReadAllLines(path);

Product\_info info;

foreach (string line in items)

{

split = line.Split(' ');

info = new Product\_info(int.Parse(split[1]), int.Parse(split[2]), int.Parse(split[2]));

market.items.Add(split[0].ToLower(), info);

}

}

else

{

throw new Exception("Incorrect path");

}

}

}

}

* 1. Product.cs

susing System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LAB\_6\_OOP

{

struct Product\_info

{

public int price;

public int how;

public int how\_remain;

public Product\_info(int price, int how,int how\_remain)

{

this.price = price;

this.how = how;

this.how\_remain = how\_remain;

}

}

}

* 1. Programm.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace LAB\_6\_OOP

{

static class Program

{

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Start\_form());

}

}

}

* 1. Start\_form.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 LAB\_6\_OOP

{

public partial class Start\_form : Form

{

public Start\_form()

{

InitializeComponent();

}

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

{

if (!(path1.Text == "" || path2.Text == ""))

{

Markets newForm = new Markets(first\_title.Text, first\_address.Text,

second\_title.Text, second\_address.Text,

path1.Text,

path2.Text,

this);

newForm.Show();

this.Hide();

}

else

{

MessageBox.Show("Попробуйте ввести название файла");

}

}

private void Form2\_Load(object sender, EventArgs e)

{

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void label1\_Click\_1(object sender, EventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void textBox5\_TextChanged(object sender, EventArgs e)

{

}

private void second\_title\_TextChanged(object sender, EventArgs e)

{

}

private void textBox1\_TextChanged\_1(object sender, EventArgs e)

{

}

}

}

* 1. Supermarket.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LAB\_6\_OOP

{

abstract class Supermarket

{

public string title;

public string address;

public Manager manager = new Manager();

public Сasher casher = new Сasher();

public SortedDictionary<string, Product\_info> items = new SortedDictionary<string, Product\_info>();

public Supermarket(string title, string address)

{

this.title = title;

this.address = address;

}

}

}

* 1. Casher.cs

using System.Collections.Generic;

using System.Windows.Forms;

namespace LAB\_6\_OOP

{

partial class Сasher

{

public void Buy\_item(Supermarket market, string product\_title, string buy\_count)

{

int count = market.items[product\_title].how\_remain;

if (count < int.Parse(buy\_count))

{

MessageBox.Show("В магазине недостаточно товара, попробуйте ещё раз");

}

else

{

var prod = Create\_product(market.items, product\_title, buy\_count, count - int.Parse(buy\_count));

market.items[key: product\_title] = prod;

}

}

private Product\_info Create\_product(SortedDictionary<string, Product\_info> items, string product\_title, string buy\_count, int how\_remain)

{

var product = new Product\_info(

items[product\_title].price,

items[product\_title].how,

how\_remain

);

return product;

}

}

}