Начало

ce, n

i = 0, n - 1

rpt = -1

Создать пустой объект SP

j = 0, SP.n - 1

ce[i].Producer = SP[j].Producer

rpt = j

rpt = -1

Добавить новый элемент SP.Add(ce[i])

Расчет количества SP[rpt].Count = SP[rpt].Count + ce[i].Count

Вернуть результат SP

Конец

нет

да

1. Расчет количественной статистики по производителям

Начало

ce, n

i = 0, n - 1

rpt = -1

Создать пустой объект SP

j = 0, SP.n - 1

ce[i].Type = SP[j].Type

rpt = j

rpt = -1

Расчет суммы

ce[i].Price = ce[i].Summa()

SP[rpt].Price = SP[rpt].Price + ce[i].Summa()

Вернуть результат SP

Конец

нет

да

Добавить новый элемент SP.Add(ce[i])

1. Расчет количественной статистики по категориям товаров

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.IO;

using System.Windows.Forms;

namespace compteh

{

public class Hardware//класс, описывающий компьютерную технику

{

private int art;//Артикул

private string prod;//Производитель

private string descr;//Описание

private int count;//Наличие

private double price;//Цена

private string type;//Тип

public Hardware()

//конструктор по умолчанию

{

this.art = 0;

this.prod = "";

this.descr = "";

this.count = 0;

this.price = 0;

this.type = "";

}

public Hardware(int art, string prod, string descr, int count, double price, string type)

//конструктор с параметрами

//art - артикул

//prod - производитель

//descr - описание

//count - наличие

//price - цена

//type - тип

{

this.art = art;

this.prod = prod;

this.descr = descr;

this.count = count;

this.price = price;

this.type = type;

}

public Hardware(Hardware A)

//конструктор копирования

{

this.art = A.art;

this.prod = A.prod;

this.descr = A.descr;

this.count = A.count;

this.price = A.price;

this.type = A.type;

}

//свойства

public int Article

//вернуть или задать значение артикула

{

get {return art;}

set {art = value;}

}

public string Producer

//вернуть или задать значение производителя

{

get {return prod;}

set {prod = value;}

}

public string Description

//вернуть или задать значение описания товара

{

get {return descr;}

set {descr = value;}

}

public int Count

//вернуть или задать значение количества товара

{

get {return count;}

set {count = value;}

}

public double Price

//вернуть или задать значение цены

{

get { return price; }

set { price = value; }

}

public string Type

//вернуть или задать значение категории товара

{

get { return type; }

set { type = value; }

}

//методы

public double Summa()

//вычисляет суммарную стоимость товара

{

return count \* price;

}

public void ToDataGrid(DataGridView table, int row, bool sum)

//вывод данных в строку таблицы

//table - указатель на таблицу

//row - номер строки

//sum - флажок вывода суммарной стоимости (true - выводить)

{

table[0, row].Value = art;

table[1, row].Value = prod;

table[2, row].Value = descr;

table[3, row].Value = count;

table[4, row].Value = price;

table[5, row].Value = type;

if (sum)

table[6, row].Value = count \* price;

}

public Hardware FromDataGrid(DataGridView table, int row)

//извлечь данные из строки таблицы

//table - указатель на таблицу

//row - номер строки

{

Hardware t = new Hardware();

t.art = Convert.ToInt32(table[0, row].Value);

t.prod = table[1, row].Value.ToString();

t.descr = table[2, row].Value.ToString();

t.count = Convert.ToInt32(table[3, row].Value);

t.price = Convert.ToDouble(table[4, row].Value);

t.type = table[5, row].Value.ToString();

return t;

}

}

public class ComputerEquipment : Hardware//класс компьютерная техника (магазин)

{

private int n;//количество техники

private Hardware[] ce;//массив данных

public ComputerEquipment()

//конструктор по умолчанию

{

n = 0;

ce = new Hardware[500];

}

public ComputerEquipment(ComputerEquipment A)

//конструктор копирования

{

int i;

ce = new Hardware[500];

n = A.n;

for (i = 0; i < n; i++)

ce[i] = new Hardware(A[i]);

}

public void Add(int art, string prod, string descr, int count, double price, string type)

//добавить объект в массив

//art - артикул

//prod - производитель

//descr - описание

//count - наличие

//price - цена

//type - тип

{

ce[n]=new Hardware(art, prod, descr, count, price, type);

n++;

}

public void Add(Hardware A)

//добавить объект в массив

//А - данные объекта

{

ce[n] = new Hardware(A);

n++;

}

public bool RemoveAt(int index)

//удаление элмента с заданным индексом из массива

{

int i;

if (index < 0 || index >= n)

return false;

for (i = index; i < n - 1; i++)

ce[i] = ce[i + 1];

n--;

return true;

}

public int Size

//возвращает число элементов

{

get { return n; }

}

public Hardware this[int i]

//индексатор: получить или задать элемент массива

{

get//вернуть

{

if (i >= 0 && i < n)//если индекс допустимый

return ce[i];//возвращаем значение

else

return null;//иначе - пустое значение

}

set//задать

{

if (i >= 0 && i < n)//если индекс допустимый

this.ce[i] = value;//присвоили элементу массива значение

}

}

public void Load(string fname)

//загрузка данных из файла

{

int i;//счетчик

int art;//Артикул

string prod;//Производитель

string descr;//Описание

int count;//Наличие

double price;//Цена

string type;//Тип

int N;

if (!File.Exists(fname)) //файл не существует

return;

StreamReader f\_baza = new StreamReader(fname, Encoding.Default);//указатель на файл

N = Convert.ToInt32(f\_baza.ReadLine());//читаем

//MessageBox.Show(n.ToString());

for (i = 0; i < N; i++)//читаем данные из файла

{

art = Convert.ToInt32(f\_baza.ReadLine());

prod = f\_baza.ReadLine();

descr = f\_baza.ReadLine();

count = Convert.ToInt32(f\_baza.ReadLine());

price = Convert.ToDouble(f\_baza.ReadLine());

type = f\_baza.ReadLine();

Add(art, prod, descr, count, price, type);//добавляем в массив объектов

}

f\_baza.Close();//закрыли файл

}

public void Save(string fname)

//сохранение базы данных в файле

{

StreamWriter f\_baza = new StreamWriter(fname, false, Encoding.Default);//указатель на файл

int i;//параметр цикла

f\_baza.WriteLine(n);

for (i = 0; i < n; i++)//запись

{

f\_baza.WriteLine(ce[i].Article);

f\_baza.WriteLine(ce[i].Producer);

f\_baza.WriteLine(ce[i].Description);

f\_baza.WriteLine(ce[i].Count);

f\_baza.WriteLine(ce[i].Price);

f\_baza.WriteLine(ce[i].Type);

}

f\_baza.Close();//закрыли файл

}

public void Sort(int field)

//сортировка списка

{

if (n == 0) return;//список пустой

int i, j;//параметры циклов

Hardware temp;//переменная для перестановки

bool cr;//результат сравнения

//пузырьковая сортировка

for (i = 1; i < n; i++)//внешний цикл

{

for (j = 0; j < n - i; j++)//внутренний цикл

{

switch (field)

{

case 2://определяем критерий сравнения

cr = (ce[j].Producer.ToUpper().CompareTo(ce[j + 1].Producer.ToUpper()) > 0);

break;

case 3:

cr = (ce[j].Description.ToUpper().CompareTo(ce[j + 1].Description.ToUpper()) > 0);

break;

case 4:

cr = ce[j].Count > ce[j + 1].Count;

break;

case 5:

cr = ce[j].Price > ce[j + 1].Price;

break;

case 6:

cr = (ce[j].Type.ToUpper().CompareTo(ce[j + 1].Type.ToUpper()) > 0);

break;

default:

cr = ce[j].Article > ce[j + 1].Article;

break;

}

if (cr)//критерий выполнен

{

temp = ce[j];//перестановка элементов

ce[j] = ce[j + 1];

ce[j + 1] = temp;

}

}

}

}

public void ToDataGrid(DataGridView table, bool sum)

//вывод массива в таблицу

//table - указатель на таблицу

//sum - флажок вывода суммарной стоимости (true - выводить)

{

int i;

table.RowCount = n;//задаем число строк таблицы

for (i = 0; i < n; i++)//вывод данных

ce[i].ToDataGrid(table, i, sum);

}

public double Total()

//подсчет общей стоимости всех товаров

{

int i;

double sum = 0;//обнулили сумму

for (i = 0; i < n; i++)//суммируем

sum = sum + ce[i].Summa();

return sum;

}

public ComputerEquipment StatProducer()

//расчет количественной статистики по производителям:

//сколько единиц товаров имеется в магазине

{

ComputerEquipment SP = new ComputerEquipment();//результат обработки

int i, j;//параметры циклов

int rpt;//номер совпавшего элемента

if (n == 0)//список пуст, выход

return null;

for (i = 0; i < n; i++)//цикл по товарам

{

rpt = -1;//совпадений пока не было

for (j = 0; j < SP.n; j++)//просмотр массива результата

if (ce[i].Producer == SP[j].Producer)//названия производителей совпали

{

rpt = j;//запомнили номер

break;//выход из цикла

}

if (rpt == -1)//совпадений не было

SP.Add(ce[i]);//добавляем новый элемент

else//есть совпадение

SP[rpt].Count = SP[rpt].Count + ce[i].Count;//увеличили счетчик количества

}

return SP;//вернули результат

}

public ComputerEquipment StatType()

//расчет количественной статистики по категориям товаров:

//сумма по каждой категории

{

ComputerEquipment SP = new ComputerEquipment();//результат обработки

int i, j;//параметры циклов

int rpt;//номер совпавшего элемента

if (n == 0)//список пуст, выход

return null;

for (i = 0; i < n; i++)//цикл по товарам

{

rpt = -1;//совпадений пока не было

for (j = 0; j < SP.n; j++)//просмотр массива результата

if (ce[i].Type == SP[j].Type)//названия типов совпали

{

rpt = j;//запомнили номер

break;//выход из цикла

}

if (rpt == -1)//совпадений не было

{

ce[i].Price = ce[i].Summa();

SP.Add(ce[i]);//добавляем новый элемент

}

else//есть совпадение

SP[rpt].Price = SP[rpt].Price + ce[i].Summa();//увеличили сумму

}

return SP;//вернули результат

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.IO;

namespace compteh

{

public partial class Form1 : Form

{

ComputerEquipment CE;//объект товары

ComputerEquipment Order;//объект заказы

public Form1()

{

InitializeComponent();

CE = new ComputerEquipment();//создаем объекты

Order = new ComputerEquipment();

}

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

//выход из программы

{

Close();

}

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

//выбрать файл и загрузить данные в список

{

string fname;//имя файла

if (openFileDialog1.ShowDialog() != DialogResult.OK)//файл не выбран

return;

fname = openFileDialog1.FileName;//имя файла

if (File.Exists(fname))//файл существует

{

CE.Load(fname);//загрузка

CE.ToDataGrid(dataGrid, false);//вывод в таблицу

MessageBox.Show("Данные успешно загружены.", "Информация");

}

}

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

//сохранить данные в файле

{

string fname;//имя файла

if (openFileDialog1.FileName != "")//если файл загружался

saveFileDialog1.FileName = openFileDialog1.FileName;//задаем имя по умолчанию равное загруженному

if (saveFileDialog1.ShowDialog() != DialogResult.OK)//файл не выбран

return;

fname = saveFileDialog1.FileName;//имя файла

CE.Save(fname);//сохранение

MessageBox.Show("Данные успешно сохранены.", "Информация");

}

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

//перейти к первой записи таблицы

{

if (CE.Size == 0) return;//список пустой, выход

dataGrid.CurrentCell = dataGrid.Rows[0].Cells[0];//выделяем первую запись

}

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

//перейти к предыдущей записи таблицы

{

if (CE.Size == 0) return;//список пустой, выход

int i = dataGrid.CurrentRow.Index;//определяем номер выделенной записи

if (i != 0)//если это не первая запись

dataGrid.CurrentCell = dataGrid.Rows[i - 1].Cells[0];//смещаемся к предыдущей

}

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

//перейти к следующей записи таблицы

{

if (CE.Size == 0) return;//список пустой, выход

int i = dataGrid.CurrentRow.Index;//определяем номер выделенной записи

if (i != CE.Size - 1)//если это не последняя запись

dataGrid.CurrentCell = dataGrid.Rows[i + 1].Cells[0];//смещаемся к следующей

}

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

//перейти к последней записи таблицы

{

if (CE.Size == 0) return;//список пустой, выход

dataGrid.CurrentCell = dataGrid.Rows[CE.Size - 1].Cells[0];//выделяем последнюю запись

}

private void Form1\_Load(object sender, EventArgs e)

//инициализация формы

{

cbSortField.SelectedIndex = 0;//задаем индекс выбранного элемента списка полей сортировки

}

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

//сортировка

{

int field;//поле сортировки

field = cbSortField.SelectedIndex + 1;//определяем номер выбранного поля в списке

CE.Sort(field);//сортируем

CE.ToDataGrid(dataGrid, false);//вывод результата в таблицу

}

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

//вставка новой записи в конец

{

Form2 f2 = new Form2();//форма работы с записью

int art;//Артикул

string prod;//Производитель

string descr;//Описание

int count;//Наличие

double price;//Цена

string type;//Тип

if (f2.ShowDialog() != DialogResult.OK)//если не была нажата кнопка Применить на форме

return;//выход

art = Convert.ToInt32(f2.textBox1.Text);//читаем данные из полей формы

prod = f2.textBox2.Text;

descr = f2.textBox3.Text;

count = Convert.ToInt32(f2.textBox4.Text);

price = Convert.ToDouble(f2.textBox5.Text);

type = f2.textBox6.Text;

CE.Add(art, prod, descr, count, price, type);//добавляем запись

CE.ToDataGrid(dataGrid, false);//обновляем таблицу

dataGrid.CurrentCell = dataGrid.Rows[CE.Size - 1].Cells[0];//выделяем добавленную запись

MessageBox.Show("Запись добавлена.", "Информация");

}

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

//редактирование выбранной записи

{

int idx;//индекс выбранной строки

Form2 f2 = new Form2();//форма работы с записью

int art;//Артикул

string prod;//Производитель

string descr;//Описание

int count;//Наличие

double price;//Цена

string type;//Тип

if (dataGrid.CurrentRow != null)//проверка - есть ли выделеннная строка

idx = dataGrid.CurrentRow.Index;//текущая строка

else//нет выделенных строк

{

MessageBox.Show("Необходимо выделить строку!", "Информация");//вывод сообщения

return;

}

f2.textBox1.Text = CE[idx].Article.ToString();//помещаем в поля формы данные текущей записи

f2.textBox2.Text = CE[idx].Producer;

f2.textBox3.Text = CE[idx].Description.ToString();

f2.textBox4.Text = CE[idx].Count.ToString();

f2.textBox5.Text = CE[idx].Price.ToString();

f2.textBox6.Text = CE[idx].Type;

if (f2.ShowDialog() != DialogResult.OK)//если не была нажата кнопка Применить на форме

return;

art = Convert.ToInt32(f2.textBox1.Text);//читаем данные из полей формы

prod = f2.textBox2.Text;

descr = f2.textBox3.Text;

count = Convert.ToInt32(f2.textBox4.Text);

price = Convert.ToDouble(f2.textBox5.Text);

type = f2.textBox6.Text;

CE[idx] = new Hardware(art, prod, descr, count, price, type);//изменяем запись

CE[idx].ToDataGrid(dataGrid, idx, false);//обновляем таблицу

}

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

//удаление текущей записи

{

int idx;//индекс выбранной строки

if (CE.Size == 0) return;//записей нет, выход

if (dataGrid.CurrentRow != null)//проверка - есть ли выделеннная строка

idx = dataGrid.CurrentRow.Index;//текущая строка

else//нет выделенных строк

{

MessageBox.Show("Необходимо выделить строку!", "Информация");//вывод сообщения

return;

}

CE.RemoveAt(idx);//удаление

CE.ToDataGrid(dataGrid, false);//обновляем таблицу

dataGrid.CurrentCell = dataGrid.Rows[0].Cells[0];//выделяем первую запись

MessageBox.Show("Запись удалена.", "Информация");//вывод сообщения

}

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

//перенос выбранной записи в таблицу заказов

{

int idx;//индекс выбранной строки

int kol;//количество приобретаемого товара

if (CE.Size == 0) return;//записей нет, выход

if (dataGrid.CurrentRow != null)//проверка - есть ли выделеннная строка

idx = dataGrid.CurrentRow.Index;//текущая строка

else//нет выделенных строк

{

MessageBox.Show("Необходимо выделить строку!", "Информация");//вывод сообщения

return;

}

kol = Convert.ToInt32(toolStripTextBox1.Text);//читаем количество

if (kol > CE[idx].Count)//проверка, если заказанное количество превышает имеющееся

{

MessageBox.Show("Недостаточное количество товара!", "Информация");//вывод сообщения

return;//выход

}

if (kol == 0)//если задан 0

{

MessageBox.Show("Необходимо задать количество товара!", "Информация");//вывод сообщения

return;//выход

}

int kt = CE[idx].Count;//запомнили текущее количество товара

CE[idx].Count = kol;//изменили количество на заказанное

Order.Add(CE[idx]);//добавили товар

CE[idx].Count = kt - kol;//уменьшили исходное количество на приобретенное

CE[idx].ToDataGrid(dataGrid, idx, false);//обновили основную таблицу

Order.ToDataGrid(orderGrid, true);//обновили таблицу заказов

}

private void toolStripTextBox1\_TextChanged(object sender, EventArgs e)

//блокировка кнопки добавления к заказу, если поле количества пустое

{

toorderButton.Enabled = toolStripTextBox1.Text != "";

}

private void toolStripTextBox1\_KeyPress(object sender, KeyPressEventArgs e)

//проверка ввода - только целые и положительные

{

if (!((e.KeyChar >= '0') && (e.KeyChar <= '9') || //если это не цифра и не клавиша BackSpace

e.KeyChar == (int)8))

{ //игнорируем нажатую клавишу

e.Handled = true;

}

}

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

//создать новый журнал

{

CE = new ComputerEquipment();//создаем пустые объекты классов

Order = new ComputerEquipment();

dataGrid.RowCount = 0;//очищаем таблицы

orderGrid.RowCount = 0;

toolStripLabel3.Text = "Сумма заказа: 0 р.";//очищаем метку вывода суммы

}

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

//очистка заказа

{

Order = new ComputerEquipment();//создаем пустой объект класса

orderGrid.RowCount = 0;//очищаем таблицу

toolStripLabel3.Text = "Сумма заказа: 0 р.";//очищаем метку вывода суммы

}

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

//расчет суммы заказа

{

if (CE.Size == 0) return;//записей нет, выход

toolStripLabel3.Text = "Сумма заказа: " + Order.Total().ToString() + " р.";//вывод суммы заказа

}

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

//вывод гистограммы статистики по производителям техники

{

Form3 f3 = new Form3();//форма вывод гистограммы

int i;//параметр цикла

ComputerEquipment SP = new ComputerEquipment();//создаем объект класса

if (CE.Size == 0) return;//записей нет, выход

SP = CE.StatProducer();//формируем статистику

f3.chart1.Series[0].Points.Clear();//очистка графика

for (i = 0; i < SP.Size; i++)//вывод на график данных

f3.chart1.Series[0].Points.AddXY(SP[i].Producer, SP[i].Count);

f3.ShowDialog();//вывод формы

}

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

//вывод диаграммы по суммам для каждой категории товара

{

Form4 f4 = new Form4();//форма вывод диаграммы

int i;//параметр цикла

ComputerEquipment SP = new ComputerEquipment();//создаем объект класса

if (CE.Size == 0) return;//записей нет, выход

SP = CE.StatType();//формируем статистику

f4.chart1.Series[0].Points.Clear();//очистка графика

for (i = 0; i < SP.Size; i++)//вывод на график данных

f4.chart1.Series[0].Points.AddXY(SP[i].Type, SP[i].Price);

f4.ShowDialog();//вывод формы

}

}

}