**Додаток А. Інспекція програмного коду**

**ТзОВ "НАШ СОФТ"**



**ЗВІТ ПРО РЕЗУЛЬТАТИ**

**Інспекції програмного коду**

для проекту:

**«Інформаційна система коміркового складу»**

Виконавець: *ТзОВ «Наш Софт»*

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2014 р.

*Гринчук Т.А.*

Зміст

**1. Питання для інспектування………………….3**

**2. Програмний код………………………….………5**

**3. Висновки інспекції ………………………...….12**

# Питання для інспектування

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| **Питання** | **Відповідь** | **Примітки** |
| **Проект "ІС коміркового складу"** |  |  |
| ***1. Простота розуміння і відповідність стандартам кодування*** |  |  |
| 1.1. Чи зрозумілі специфікації елементу? | так |  |
| 1.2. Чи зрозумілий код елементу? Чи є проблеми в розумінні? | так | Змінні названі відповідно до їх призначення |
| 1.3. Чи достатньо код супроводжується коментарями? Чи допомагають коментарі зрозуміти процедури і функції елементу? | ні | Коментарів дуже мало |
| 1.4. Чи можуть бути труднощі при модифікації елементу? | так | Бізнес-логіка та GUI не розділені |
| 1.5. Чи встановлені в проекті стандарти кодування? | так | Використовуються загальприйняті стандарти |
| 1.6. Чи відповідає код встановленим стандартам кодування? | так |  |
| 1.7. Чи розроблені тести для перевірки дотримання стандартів? | ні |  |
| ***2. Коректність звертання до даних*** |  |  |
| 2.1. Чи використовуються непроініціалізовані змінні? | ні |  |
| 2.2. Чи не виходить значення кожного з індексів за межі, визначені для відповідного виміру при всіх звертаннях до масиву? | ні |  |
| 2.3. Чи приймає кожен індекс цілі значення при всіх звертаннях до масиву? Не цілі індекси не обов'язково є помилкою, але представляють практичну небезпеку. | так |  |
| 2.4. Чи для всіх звертань за допомогою показників або змінних-посилань пам'ять, до якої проводиться звертання, розподілена (чи є "підвішені" звартання)? | ні |  |
| 2.5. Чи коректні атрибути при всіх псевдонімах? Якщо одна і та ж область пам'яті має декілька псевдонімів (імен) з різними атрибутами, то чи мають значення даних в цій області коректні атрибути при звертанні по одному з цих псев¬донімів? | так |  |
| 2.6. Чи відповідають атрибути запису і структури? | Так |  |
| 2.7. Чи відрізняються типи або атрибути змінних величин від тих, які передбачались компілятором? Це може відбутися у тому випадку, коли програма зчитує запис із пам'яті і звертається до них як до структур, але фізичне представ¬лення записів відмінне від опису структури. | ні |  |
| 2.8. Чи обчислювані адреси бітових рядків? Чи передаються бітові рядки як аргументи? | --- |  |
| 2.9. Якщо до структури даних звертаються з декількох процедур або підпрограм, то чи визначена ця структура однаково в кожній процедурі? |  |  |
| 2.10. Чи не перевищені межі рядка при індексації в ньому? | ні |  |
| 2.11. Чи існують які-небудь інші помилки в операціях з індексацією або при звертанні до масиву по індексу? | ні |  |
| ***3. Коректність опису даних*** |  |  |
| 3.1. Чи всі змінні описані явно? | так |  |
| 3.2. Якщо початкові значення присвоюються змінним в операторах опису, то чи правильно ініціалізувалися ці значення? | так |  |
| 3.3. Чи правильно для кожної змінної визначені довжина, тип і клас пам'яті? | так |  |
| 3.4. Чи узгоджується ініціалізація змінної з її типом пам'яті? | так |  |
| 3.5. Чи є змінні зі схожими іменами? | так | Але вони містять тимчасові дані для обчислення |
| ***4. Коректність обчислень*** |  |  |
| 4.1. Чи є обчислення, що використовують змінні неприпустимих типів? | ні |  |
| 4.2. Чи є обчислення, що використовують дані різного типу (змішані обчислення)? Чи можливе відкидання дробової частини? | ні |  |
| 4.3. Чи існують обчислення, що використовують змінні одного типу, але різної  довжини? | ні |  |
| 4.4. Чи не менша довжина результату, ніж довжина обчислюваного значення? | ні |  |
| 4.5. Чи можливе переповнення або втрата проміжного результату під час обчислення? | ні |  |
| 4.6. Чи можливо, щоб дільник в операторі дорівнював нулю? | ні |  |
| 4.7. Чи можливий вихід значення змінної за межі її діапазону ? | ні |  |
| 4.8. Чи правильно використовується цілочисельна арифметика, особливо ділення. | так |  |
| ***5. Коректність порівняння даних*** |  |  |
| 5.1. Чи порівнюються величини несумісних типів (наприклад, рядок символів з адресою)? | ні |  |
| 5.2. Чи порівнюються величини різних типів? | ні |  |
| 5.3. Чи коректні відношення порівняння? | ні |  |
| 5.4. Чи коректні булеві вирази? | так |  |
| 5.5. Чи об'єднуються порівняння і булеві вирази? | ні |  |
| 5.6. Чи порівнюються дробові величини, представлені в двійковій формі? | ні |  |
| 5.7. Чи зрозумілий порядок проходження операторів? | так |  |
| ***6. Коректність передачі управління*** |  |  |
| 6.1. Чи може значення індексу в операторі вибору перевищити кількість переходів? | ні |  |
| 6.2. Чи буде завершений кожен цикл? | так |  |
| 6.3. Чи буде завершена програма? | так |  |
| 6.4. Чи можливо, що через вхідні умови цикл ніколи не зможе виконуватися? | так |  |
| 6.5. Чи коректні можливі занулення в циклі? | так |  |
| 6.6. Чи є помилки відхилення кількості ітерацій від норми? | ні |  |
| 6.7. Чи відповідають один одному оператори початку і кінця блоку ( { і } чи begin і end )? | так |  |
| ***7. Коректність інтерфейсу між модулями*** |  |  |
| 7.1. Чи є невідповідність сигнатури параметрів у викликаючому модулі(підпрограмі) і тому, що викликається ? | ні |  |
| 7.2. Якщо модуль має декілька точок входу, чи передається параметр завжди не залежно від точки входу? | --- |  |
| 7.3. Чи не змінює підпрограма параметр, який повинен використовуватися тільки як вхідна величина? | ні |  |
| 7.4. Чи всі глобальні змінні використовуються правильно? | так |  |
| 7.5. Чи передаються константи в якості аргументів? | ні |  |
| ***8. Коректність вводу-виводу*** |  |  |
| 8.1. Чи немає помилок в описах атрибутів файлів і операторах вводу-виводу? | ні |  |
| 8.2. Чи відповідає розмір буфера розміру запису? | так |  |
| 8.3. Чи відкриті файли перед їх використанням? | так |  |
| 8.4. Чи виявляються ознаки кінця файлу? | так |  |
| 8.5. Чи виявляються помилки вводу-виводу? | так |  |
| 8.6. Чи існують текстові помилки у вихідній інформації? | ні |  |

# Програмний код

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 WMS

{

public partial class Form1 : Form

{

private WMSEntities context;

public Form1()

{

InitializeComponent();

}

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

{

context = new WMSEntities();

var query = context.Clients.OrderBy(p => p.ID);

clientsBindingSource.DataSource = query.ToList();

dataGridViewTextBoxColumn1.ReadOnly = true;

}

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

{

context.SaveChanges();

}

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

{

UsersForm form = new UsersForm();

form.Show();

}

private void menuStrip1\_ItemClicked(object sender, ToolStripItemClickedEventArgs e)

{

}

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

{

WarehousesForm form = new WarehousesForm();

form.Show();

}

//збереження змін при виході з комірки

private void clientsDataGridView\_CellEndEdit(object sender, DataGridViewCellEventArgs e)

{

context.SaveChanges();

}

//отримання останнього ID таблмці

private int LastID()

{

int res = 0;

using (WMSEntities cont = new WMSEntities())

{

var query = cont.Clients.OrderBy(p => p.ID);

if (query.Count() > 0) res = query.ToList().Last().ID;

}

return res;

}

//додавання нового рядка

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

{

using (WMSEntities cont = new WMSEntities())

{

Clients t = new Clients() { ID = 1 + LastID(), Name = textBox2.Text, Code = textBox1.Text};

cont.Clients.Add(t);

cont.SaveChanges();

}

Form1\_Load(null, null);

}

//видалення рядка

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

{

HashSet<int> rows = new HashSet<int>();

foreach (var row in clientsDataGridView.SelectedCells)

{

int RowIndex = ((System.Windows.Forms.DataGridViewTextBoxCell)row).RowIndex;

rows.Add((int)clientsDataGridView[0, RowIndex].Value);

}

foreach (int row in rows)

{

using (WMSEntities cont = new WMSEntities())

{

var query = from t in cont.Clients

where t.ID == row

select t;

if (query.Count() > 0)

{

var t = query.ToList().First();

cont.Clients.Remove(t);

cont.SaveChanges();

}

}

}

Form1\_Load(null, null);

}

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

{

RegionsForm form = new RegionsForm();

form.Show();

}

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

{

ProductsForm form = new ProductsForm();

form.Show();

}

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

{

CellTypesForm form = new CellTypesForm();

form.Show();

}

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

{

CellForms form = new CellForms();

form.Show();

}

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

{

}

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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 WMS

{

public partial class CellForms : Form

{

private WMSEntities context;

public CellForms()

{

InitializeComponent();

}

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

{

context = new WMSEntities();

context.Configuration.LazyLoadingEnabled = false;

//var query = context.Clients.OrderBy(p => p.ID);

var query =

from cell in context.Cells

join types in context.CellTypes on cell.TypeID equals types.ID

join region in context.Regions on cell.RegionID equals region.ID

join ware in context.Warehouses on cell.WarehouseID equals ware.ID

select new

{

cell.ID,

cell.Name,

cell.Width,

cell.Height,

cell.Depth,

cell.SizeControl,

cell.Amount,

cell.AmountControl,

cell.Stillage,

cell.Tier,

cell.Position,

TypeID = types.ID,

TypeName = types.Name,

RegionID = region.ID,

RegionName = region.Name,

WarehouseID = ware.ID,

WarehouseName = ware.Name,

};

productsDataGridView.DataSource = query.ToList();

}

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

{

AddEditCell f = new AddEditCell();

var res = f.ShowDialog();

if (res == DialogResult.OK) CellForms\_Load(null, null);

}

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

{

if (productsDataGridView.SelectedCells.Count != 1)

{

MessageBox.Show("Choose the one row to edit !");

return;

}

var row = productsDataGridView.SelectedCells[0];

int RowIndex = ((System.Windows.Forms.DataGridViewTextBoxCell)row).RowIndex;

int sID = (int)productsDataGridView[0, RowIndex].Value;

AddEditCell f = new AddEditCell(sID.ToString());

var res = f.ShowDialog();

if (res == DialogResult.OK) CellForms\_Load(null, null);

}

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

{

HashSet<int> rows = new HashSet<int>();

foreach (var row in productsDataGridView.SelectedCells)

{

int RowIndex = ((System.Windows.Forms.DataGridViewTextBoxCell)row).RowIndex;

rows.Add((int)productsDataGridView[0, RowIndex].Value);

}

foreach (int row in rows)

{

using (WMSEntities cont = new WMSEntities())

{

var query = from t in cont.Cells

where t.ID == row

select t;

if (query.Count() > 0)

{

var t = query.ToList().First();

cont.Cells.Remove(t);

cont.SaveChanges();

}

}

}

CellForms\_Load(null, null);

}

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.Entity.Migrations;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WMS

{

public partial class AddEditCell : Form

{

private WMSEntities context;

private bool editMode;

private int LastID()

{

int res = 0;

using (WMSEntities cont = new WMSEntities())

{

var query = cont.Cells.OrderBy(p => p.ID);

if (query.Count() > 0) res = query.ToList().Last().ID;

}

return res;

}

public AddEditCell()

{

editMode = false;

InitializeComponent();

IDText.Text = (1 + LastID()).ToString();

}

public AddEditCell(String ID)

{

editMode = true;

InitializeComponent();

IDText.Text = ID;

}

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

{

context = new WMSEntities();

context.Configuration.LazyLoadingEnabled = false;

//var query = context.Clients.OrderBy(p => p.ID);

var query =

from types in context.CellTypes

select new

{

types.ID,

types.Name,

};

clientsDataGridView.DataSource = query.ToList();

query =

from region in context.Regions

select new

{

region.ID,

region.Name,

};

regionDataGridView.DataSource = query.ToList();

query =

from ware in context.Warehouses

select new

{

ware.ID,

ware.Name,

};

warehoesesGridView.DataSource = query.ToList();

if (editMode)

{

int nID = int.Parse(IDText.Text);

using (WMSEntities cont = new WMSEntities())

{

var q =

from cells in cont.Cells

where cells.ID == nID

select cells;

Cells first = q.ToList().First();

NameText.Text = first.Name;

AmountText.Text = first.Amount.ToString();

WidthText.Text = first.Width.ToString();

HeightText.Text = first.Height.ToString();

DepthText.Text = first.Depth.ToString();

StillageText.Text = first.Stillage.ToString();

TierText.Text = first.Tier.ToString();

PositionText.Text = first.Position.ToString();

SizeControlText.Text = first.SizeControl.ToString();

AmountControlText.Text = first.AmountControl.ToString();

for (int i = 0; i < clientsDataGridView.RowCount; i++)

{

int nClientID = (int)clientsDataGridView[0, i].Value;

if (nClientID == first.TypeID)

{

clientsDataGridView[0, i].Selected = true;

break;

}

}

for (int i = 0; i < regionDataGridView.RowCount; i++)

{

int nRegionID = (int)regionDataGridView[0, i].Value;

if (nRegionID == first.RegionID)

{

regionDataGridView[0, i].Selected = true;

break;

}

}

for (int i = 0; i < warehoesesGridView.RowCount; i++)

{

int nWareID = (int)warehoesesGridView[0, i].Value;

if (nWareID == first.WarehouseID)

{

warehoesesGridView[0, i].Selected = true;

break;

}

}

}

}

}

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

{

if (NameText.Text.Trim().Length == 0)

{

MessageBox.Show("Name is empty !");

return;

}

if (AmountText.Text.Trim().Length == 0)

{

MessageBox.Show("Amount is empty !");

return;

}

if (WidthText.Text.Trim().Length == 0)

{

MessageBox.Show("Width is empty !");

return;

}

if (HeightText.Text.Trim().Length == 0)

{

MessageBox.Show("Height is empty !");

return;

}

if (DepthText.Text.Trim().Length == 0)

{

MessageBox.Show("Depth is empty !");

return;

}

if (StillageText.Text.Trim().Length == 0)

{

MessageBox.Show("Stillage is empty !");

return;

}

if (TierText.Text.Trim().Length == 0)

{

MessageBox.Show("Tier is empty !");

return;

}

if (PositionText.Text.Trim().Length == 0)

{

MessageBox.Show("Position is empty !");

return;

}

if (SizeControlText.Text.Trim().Length == 0)

{

MessageBox.Show("Size control is empty !");

return;

}

if (AmountControlText.Text.Trim().Length == 0)

{

MessageBox.Show("Amount control is empty !");

return;

}

decimal nAmount;

if (!decimal.TryParse(AmountText.Text.Trim(), out nAmount))

{

MessageBox.Show("Amount: bad format of number !");

return;

}

decimal nWidth;

if (!decimal.TryParse(WidthText.Text.Trim(), out nWidth))

{

MessageBox.Show("Width: bad format of number !");

return;

}

decimal nHeight;

if (!decimal.TryParse(HeightText.Text.Trim(), out nHeight))

{

MessageBox.Show("Height: bad format of number !");

return;

}

decimal nDepth;

if (!decimal.TryParse(DepthText.Text.Trim(), out nDepth))

{

MessageBox.Show("Depth: bad format of number !");

return;

}

int nStillage;

if (!int.TryParse(StillageText.Text.Trim(), out nStillage))

{

MessageBox.Show("Stillage: bad format of number !");

return;

}

int nTier;

if (!int.TryParse(TierText.Text.Trim(), out nTier))

{

MessageBox.Show("Tier: bad format of number !");

return;

}

int nPosition;

if (!int.TryParse(PositionText.Text.Trim(), out nPosition))

{

MessageBox.Show("Position: bad format of number !");

return;

}

int nSizeControl;

if (!int.TryParse(SizeControlText.Text.Trim(), out nSizeControl))

{

MessageBox.Show("Size control: bad format of number !");

return;

}

int nAmountControl;

if (!int.TryParse(AmountControlText.Text.Trim(), out nAmountControl))

{

MessageBox.Show("Amount control: bad format of number !");

return;

}

if (clientsDataGridView.SelectedCells.Count == 0)

{

MessageBox.Show("Choose the Client !");

return;

}

if (regionDataGridView.SelectedCells.Count == 0)

{

MessageBox.Show("Choose the Region !");

return;

}

if (warehoesesGridView.SelectedCells.Count == 0)

{

MessageBox.Show("Choose the Warehouse !");

return;

}

var row = clientsDataGridView.SelectedCells[0];

int RowIndex = ((System.Windows.Forms.DataGridViewTextBoxCell)row).RowIndex;

int nTypeID = (int)clientsDataGridView[0, RowIndex].Value;

row = regionDataGridView.SelectedCells[0];

RowIndex = ((System.Windows.Forms.DataGridViewTextBoxCell)row).RowIndex;

int nRegionID = (int)regionDataGridView[0, RowIndex].Value;

row = warehoesesGridView.SelectedCells[0];

RowIndex = ((System.Windows.Forms.DataGridViewTextBoxCell)row).RowIndex;

int nWareID = (int)warehoesesGridView[0, RowIndex].Value;

using (WMSEntities cont = new WMSEntities())

{

Cells t = new Cells

{

ID = int.Parse(IDText.Text),

Name = NameText.Text.Trim(),

Amount = nAmount,

Width = nWidth,

Height = nHeight,

Depth = nDepth,

Stillage = nStillage,

Tier = nTier,

Position = nPosition,

SizeControl = nSizeControl,

AmountControl = nAmountControl,

TypeID = nTypeID,

RegionID = nRegionID,

WarehouseID = nWareID

};

cont.Cells.AddOrUpdate(t);

cont.SaveChanges();

}

DialogResult = DialogResult.OK;

Close();

}

}

}

# Висновки інспекції

Провівши інспекцію коду, виявив в інспектованому проекті наступні порушення загальновідомих норм кодування, які рекомендую виправити:

* Код містить не достатньо коментарів;
* Не реалізовані шаблони розділення бізнес-логіки від графічного інтерфейсу, такі як: MVC, MVP, може призвести до труднощів при подальшому масштабуванні проекту. Наприклад якщо логіку і GUI будуть реалізовувати різні спеціалісти їм буде важко розділити роботу між собою.