

Івано-Франківський національний технічний
університет нафти і газу

Кафедра
інженерії програмного забезпечення

Лабораторна робота №4

Структурні шаблони

Виконав
Ст. гр. ІП-22-1
Хімії Денис
Перевірила
Піх М.М.

Івано-Франківськ
2024

Мета: продемонструвати реалізацію структурних шаблонів проєктування в коді проєкту

Хід виконання роботи

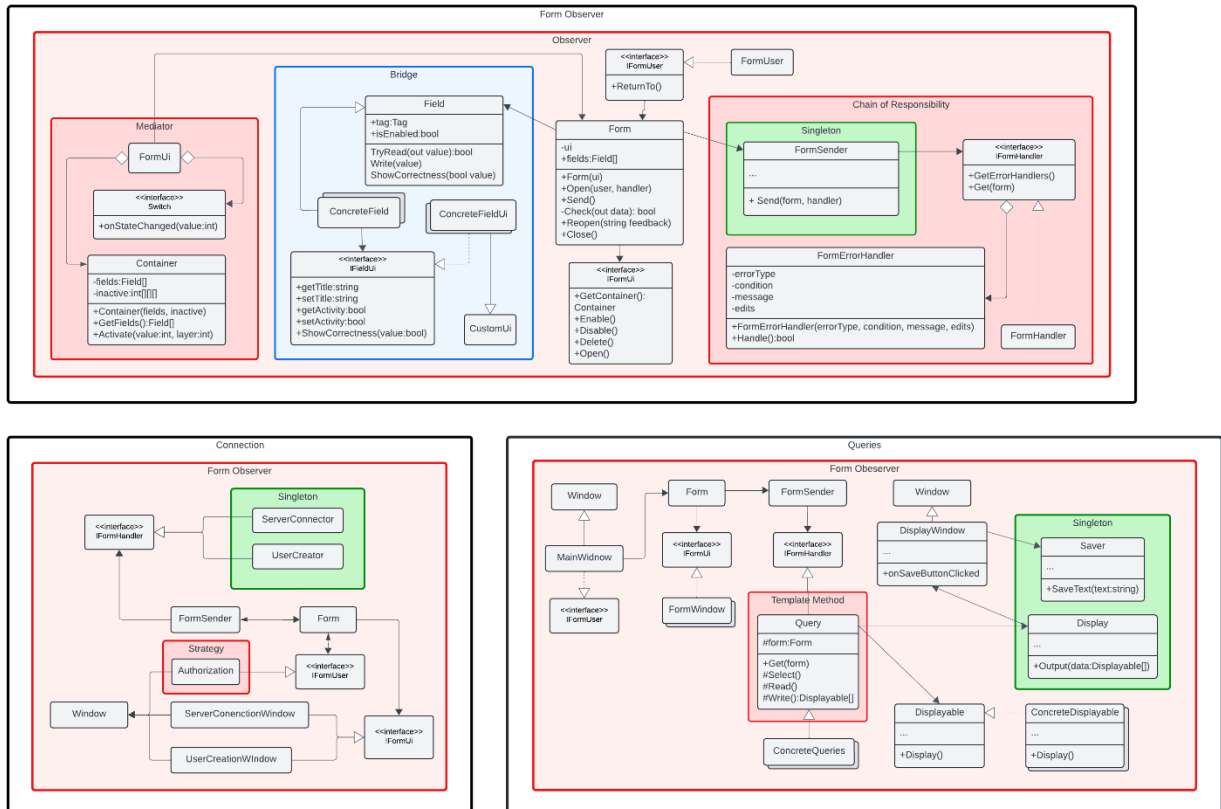


Рис. 1 – Загальна діаграма класів проєкту

Як випливає з рисунку 1 в роботі використано лише один структурний шаблон проєктування (позначено синім) – Міст. Нижче наведено всі класи, які його реалізують (всі вони стосуються розділення інтерфейсу та бізнес-логіки полів для вводу даних).

IFieldUi

```
namespace CS.General.Form.Field.UI
{
    public interface IFieldUi
    {
        public string Title { get; set; }
        public bool IsEnabled { get; set; }

        public void ShowCorrectness(bool correct);
    }
}
```

ITextFieldUi

```
namespace CS.General.Form.Field.UI
{
    public interface ITextFieldUi : IFieldUi
    {
        public string Text { get; set; }
    }
}
```

DataField

```
using CS.General.Form.Field.UI;
using System.Windows.Controls;
using System.Windows.Controls.Primitives;
using System.Windows.Media;

namespace CS.General.Form.Field
{
    public partial class DateField : UserControl, ITextFieldUi
    {
        public string Title
        {
            get => Header.Text;
            set => Header.Text = value;
        }

        public string Text
        {
            get => Field.SelectedDate.HasValue ?
Field.SelectedDate.Value.ToString("yyyy-MM-dd") : string.Empty;
            set => Field.SelectedDate = DateTime.ParseExact(value, "yyyy-MM-dd",
null);
        }

        public DateField()
        {
            InitializeComponent();

            Field.Loaded += (s, e) =>
            {
                if (Field.Template.FindName("PART_TextBox", Field) is
DatePickerTextBox datePickerTextBox)
                {
                    var watermarkProperty =
typeof(DatePickerTextBox).GetProperty("Watermark", System.Reflection.BindingFlags.NonPublic
| System.Reflection.BindingFlags.Instance);
                    watermarkProperty?.SetValue(datePickerTextBox, "Оберіть
дату");
                }
            };
        }

        public void ShowCorrectness(bool correct)
        {
            Header.Foreground = correct ? Brushes.Black : Brushes.Red;
        }
    }
}
```

ListField

```
using CS.General.Form.Field.UI;
using System.Windows.Controls;
using System.Windows.Media;

namespace CS.General.Form.Field
{
    public partial class ListField : UserControl, IFieldUi
    {
        public int Selected
        {
            get => Combo.SelectedIndex;
            set => Combo.SelectedIndex = value;
        }

        public string Title
        {
            get => Header.Text;
            set => Header.Text = value;
        }

        private int _previousSelected = -1;

        public ComboBox List { set => Combo.ItemsSource = Database.ComboLists[value]; }

        public void SetEvent(Action<int> action)
        {
            Combo.SelectionChanged += (s, e) => action(Selected);
        }

        public ListField()
        {
            InitializeComponent();
        }

        public void ShowCorrectness(bool correct)
        {
            Header.Foreground = correct ? Brushes.Black : Brushes.Red;
        }

        private void Combo_DropDownClosed(object sender, EventArgs e)
        {
            if (_previousSelected == Selected)
            {
                Selected = -1;
                _previousSelected = -1;
            }
            else _previousSelected = Selected;
        }
    }
}
```

TextField

```
using CS.General.Form.Field.UI;
using System.Windows.Controls;
using System.Windows.Media;

namespace CS.General.Form.Field
{
    public partial class TextField : UserControl, ITextFieldUi
    {
        public string Title
        {
            get => Header.Text;
            set => Header.Text = value;
        }

        public string Text
        {
            get => Field.Text;
            set => Field.Text = value;
        }

        public void ShowCorrectness(bool correct)
        {
            Header.Foreground = correct ? Brushes.Black : Brushes.Red;
        }

        public TextField()
        {
            InitializeComponent();
        }
    }
}
```

EnumField

```
using CS.General.Form.Field.UI;

namespace CS.General.Form.Field.Logic
{
    internal class EnumField : Field
    {
        public override bool IsEnabled
        {
            get => Ui.IsEnabled;
            set
            {
                if (!value) Ui.Selected = -1;
                Ui.IsEnabled = value;
            }
        }

        protected ListField Ui { get; private set; }

        public EnumField(ListField ui, ComboList type, Tag tag) : base(tag)
        {
            Ui = ui;
            Ui.List = type;
        }

        protected virtual bool Check() => Ui.Selected > -1;

        public override bool TryRead(out object value)
        {
            value = Ui.Selected;
            bool result = Check();
            ShowCorrectness(result);
            return result;
        }

        public override void Write(object value)
        {
            Ui.Selected = (int)value;
            ShowCorrectness(false);
        }

        protected override void ShowCorrectness(bool correct)
        {
            Ui.ShowCorrectness(correct);
        }
    }
}
```


Field

```
using CS.General.Form.Field.UI;

namespace CS.General.Form.Field.Logic
{
    public abstract class Field(Tag tag)
    {
        public Tag Tag => tag;

        public abstract bool IsEnabled { get; set; }

        public abstract bool TryRead(out object value);

        public abstract void Write(object value);

        protected abstract void ShowCorrectness(bool correct);
    }

    public enum Tag
    {
        Default, Host, Port, Database, Name, Password, Type,
        Unit, ProductCategory, Product, Workshop, Site,
        FirstDate, LastDate,
        EmployeeCategory, Engineer, Laborer,
        Laboratory, ProductsPart
    }
}
```

IntegerField

```
namespace CS.General.Form.Field.Logic
{
    internal class IntegerField(TextField ui, Tag tag) : StringField(ui, tag)
    {
        public override bool TryRead(out object value)
        {
            bool result = base.TryRead(out value) & int.TryParse(value.ToString(), out int
number);
            value = number;
            ShowCorrectness(result);
            return result;
        }
    }
}
```

OptionalEnumField

```
namespace CS.General.Form.Field.Logic
{
    internal class OptionalEnumField : EnumField
    {
        public OptionalEnumField(ListField ui, ComboList type, Tag tag) : base(ui, type,
tag)
        {
            ui.Title += " *";
        }

        protected override bool Check() => true;
    }
}
```

StringField

```
using CS.General.Form.Field.UI;

namespace CS.General.Form.Field.Logic
{
    public class StringField(ITextFieldUi ui, Tag tag) : Field(tag)
    {
        public override bool IsEnabled
        {
            get => Ui.IsEnabled;
            set
            {
                if (!value) Ui.Text = string.Empty;
                Ui.IsEnabled = value;
            }
        }

        protected ITTextFieldUi Ui { get; private set; } = ui;

        public override bool TryRead(out object value)
        {
            value = Ui.Text;
            bool result = Ui.Text.Length > 0;
            ShowCorrectness(result);
            return result;
        }

        public override void Write(object value)
        {
            Ui.Text = value.ToString();
            ShowCorrectness(false);
        }

        protected override void ShowCorrectness(bool correct)
        {
            Ui.ShowCorrectness(correct);
        }
    }
}
```

Висновок

На цій лабораторній роботі продемонструвати реалізацію структурного шаблону проєктування Міст в коді проєкту.