МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики
Кафедра програмного забезпечення комп’ютерних систем

# **Лабораторна робота №** 3

з дисципліни “ООП”
тема “ C# .Net. Розширені можливості реалізації ООП у мові C#. Події”

Виконав
Студент 2 курсу
групи КП-03
Хоменко Максим Вячеславович
(*прізвище, ім’я, по батькові*)

Перевірив
“____” “____________” 20___ р.
викладач
Заболотня Тетяна Миколаївна
(*прізвище, ім’я, по батькові*)

варіант №20

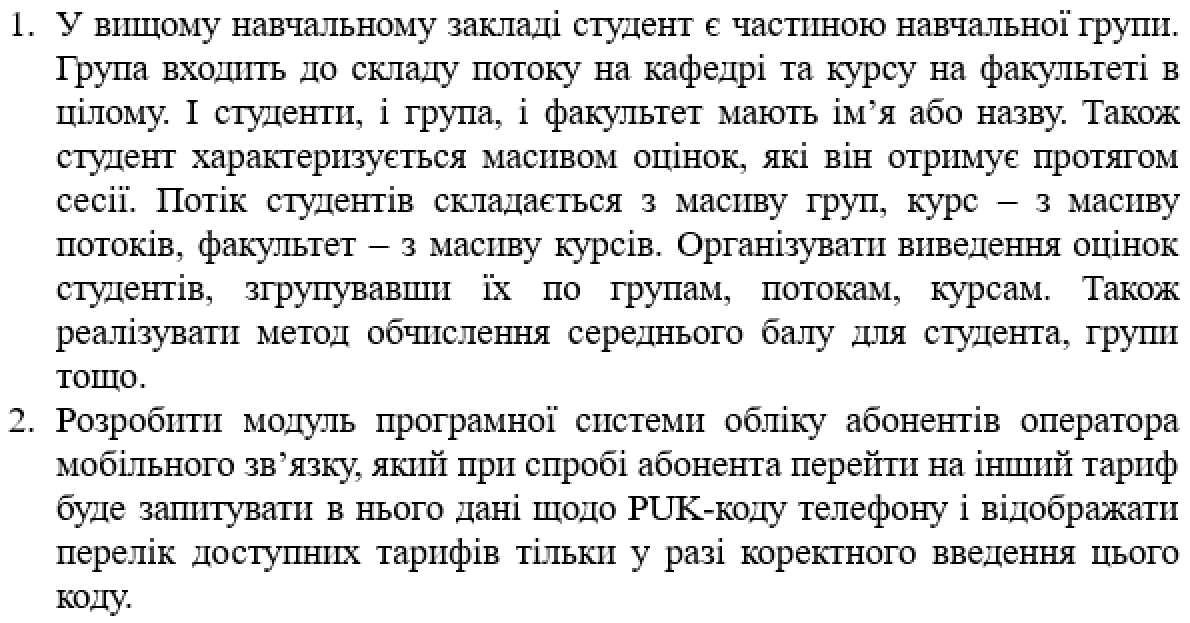
Київ 2021

## Мета роботи

Ознайомитися з такими наблонами проектування, як Composite та Proxy.

Розв’язати проблему, реалізувавши шалон певною мовою.

## Постановка завдання



1. У вищому навчальному закладі студент є частиною навчальної групи. Група входить до складу потоку на кафедрі та курсу на факультеті в цілому. І студенти, і група, і факультет мають ім’я або назву. Також студент характеризується масивом оцінок, які він отримує протягом сесії. Потік студентів складається з масиву груп, курс – з масиву потоків, факультет – з масиву курсів. Організувати виведення оцінок студентів, згрупувавши їх по групам, потокам, курсам. Також реалізувати метод обчислення середнього балу для студента, групи тощо.
2. Розробити модуль програмної системи обліку абонентів оператора мобільного зв’язку, який при спробі абонента перейти на інший тариф буде запитувати в нього дані щодо PUK-коду телефону і відображати перелік доступних тарифів тільки у разі коректного введення цього коду.

## Коротке обгрунтування вибору шаблонів

1. У системі, простежується ієрархія композицій. Ці відношення можна уявити у вигляді дерева. Ми наперед не знаємо скільки нащадків матиме той чи інший вузол, тому не можемо застосувати цикли для виконання певних операцій. Тому використовуємо шаблон Composite
2. У системі виникає потреба у тому, щоб обмежувати доступ до певного сервісу зміни тарифного плану. Це можливо, якщо скористатися захисним проксі(protecting proxy)

## Фрагменти коду

Task 1

### StudAnalyzeComponent.cs

```
using System;
using System.Collections.Generic;
using System.Linq;

namespace Part1
{
	public abstract class StudAnalyzeComponent // component defines general operations for simple and complex
objects
	{
	protected double ComponentsAverage = 0;

	protected IEnumerable<StudAnalyzeComponent> Components;

	public virtual string Naming => "undefined";

	public StudAnalyzeComponent(IEnumerable<StudAnalyzeComponent> components)
	{
		Components = components;
	}

	public virtual void Add(StudAnalyzeComponent c)
	{
		Components.ToList().Add(c);
	}

	public virtual void Remove(StudAnalyzeComponent c)
	{
		Components.ToList().Remove(c);
	}
```

```
        public virtual double Avg()
        {
                ComponentsAverage = 0;
                foreach (var item in Components)
                {
                ComponentsAverage += item.Avg();
                }
                return ComponentsAverage / Components.Count();
        }

        public virtual void Display(int ident)
        {
                ident += 3;
                foreach (var item in Components)
                {
                Console.WriteLine($"{new String(' ', ident)}{item.Naming} - {item.Avg():f2}");
                item.Display(ident);
                Console.WriteLine();
                }
        }
        }
}
```

## Course.cs

```
using System.Collections.Generic;

namespace Part1.Models
{
        public class Course : StudAnalyzeComponent // composite(Container, box)
        {
        public string CourseName { get; set; }
        public override string Naming => CourseName;

        public Course(IEnumerable<CourseStream> streams) : base(streams) { }
        }
}
```

## CourseStream.cs

```
using System.Collections.Generic;

namespace Part1.Models
{
        public class CourseStream : StudAnalyzeComponent // composite(Container, box)
        {
        public string StreamName { get; set; }
        public override string Naming => StreamName;
        public CourseStream(IEnumerable<Group> groups) : base(groups) { }

        }
}
```

## Faculty.cs

```
using System.Collections.Generic;

namespace Part1.Models
{
        public class Faculty : StudAnalyzeComponent // composite(Container, box)
        {
        public string FacultyName { get; set; }
        public override string Naming => FacultyName;

        public Faculty(IEnumerable<Course> courses) : base(courses)
        {
                this.Components = courses;
        }
        }
}
```

## Group.cs

```
using System.Collections.Generic;

namespace Part1.Models
{
        public class Group : StudAnalyzeComponent // composite(Container, box)
        {
        public string GroupName { get; set; }
        public override string Naming => GroupName;
        public Group(IEnumerable<StudAnalyzeComponent> students) : base(students) { }
        }
}
```

## Student.cs

```
using System;
using System.Collections.Generic;
using System.Linq;

namespace Part1.Models
{
        public class Student : StudAnalyzeComponent // Leaf
        {
        public string Firstname { get; set; }
        public string Lastname { get; set; }
        public string Middlename { get; set; }
        public override string Naming => $"{Firstname} {Middlename} {Lastname}";

        public IEnumerable<double> Grades { get; set; }

        public Student() : base(null) { }
        public override double Avg()
        {
                return Grades.Average(g => g);
        }

        public override void Display(int ident)
        {
                string str = "";
                foreach (var item in Grades)
                {
                str += $"{item} ";

                }
                Console.WriteLine($"{new String(' ', ident)}Grades: {str}");
        }

        public override void Add(StudAnalyzeComponent c)
        {
                throw new InvalidOperationException();
        }

        public override void Remove(StudAnalyzeComponent c)
        {
                throw new InvalidOperationException();
        }
        }
}
```

## Program.cs

```
using Part1.Models;
using System;
using System.Collections.Generic;

namespace Part1
{
        class Program
        {
        private static StudAnalyzeComponent InitFAM()
        {
                return new Faculty(new List<Course>
                {
                        new Course(new List<CourseStream>
                                {
                                new CourseStream(new List<Group>
                                        {
                                        new Group(new List<Student>
                                                {
                                                        new Student
                                                        {
                                                        Firstname = "Petro3",
                                                        Lastname = "Petrenko3",
                                                        Middlename = "Petrovych3",
                                                        Grades = new double[]{ 50, 100, 14, 78, 89, 56.5, 88.75 }
                                                        },
                                                        new Student
                                                        {
                                                        Firstname = "Petro4",
                            Lastname = "Petrenko4",
                            Middlename = "Petrovych4",
                            Grades = new double[]{ 50, 90, 14, 78, 89, 56.5, 88.75 }
                            },

                            new Student
                            {
                            Firstname = "Petro4",
                            Lastname = "Petrenko4",
                            Middlename = "Petrovych4",
                            Grades = new double[]{ 50, 90, 14, 78, 89, 56.5, 88.75 }
                            },
                        })
                    {
                        GroupName = "KP-11",
                    },

                    new Group(new List<Student>
                        {
                            new Student
                            {
                            Firstname = "Petro1",
                            Lastname = "Petrenko1",
                            Middlename = "Petrovych1",
                            Grades = new double[]{ 50, 100, 14, 78, 89, 56.5, 88.75 }
                            },
                            new Student
                            {
                            Firstname = "Petro2",
                            Lastname = "Petrenko2",
                            Middlename = "Petrovych2",
                            Grades = new double[]{ 50, 90, 14, 78, 89, 56.5, 88.75 }
                            },

                            new Student
                            {
                            Firstname = "Petro0",
                            Lastname = "Petrenko0",
                            Middlename = "Petrovych0",
                            Grades = new double[]{ 50, 90, 14, 78, 89, 56.5, 88.75 }
                            },
                        })
                    {
                        GroupName = "KP-12",
                    }
                    })
                {
                    StreamName = "KP-1x",
                },

                new CourseStream(new List<Group>
                    {
                    new Group(new List<Student>
                        {
                            new Student
                            {
                            Firstname = "Petro3",
                            Lastname = "Petrenko3",
                            Middlename = "Petrovych3",
                            Grades = new double[]{ 50, 100, 14, 78, 89, 56.5, 88.75 }
                            },
                            new Student
                            {
```

```
            Firstname = "Petro4",
            Lastname = "Petrenko4",
            Middlename = "Petrovych4",
            Grades = new double[]{ 50, 90, 14, 78, 89, 56.5, 88.75 }
            },

            new Student
            {
            Firstname = "Petro4",
            Lastname = "Petrenko4",
            Middlename = "Petrovych4",
            Grades = new double[]{ 50, 90, 14, 78, 89, 56.5, 88.75 }
            },
        })
    {
        GroupName = "KP-01",
    },

    new Group(new List<Student>
        {
            new Student
            {
            Firstname = "Petro1",
            Lastname = "Petrenko1",
            Middlename = "Petrovych1",
            Grades = new double[]{ 50, 100, 14, 78, 89, 56.5, 88.75 }
            },
            new Student
            {
            Firstname = "Petro2",
            Lastname = "Petrenko2",
            Middlename = "Petrovych2",
            Grades = new double[]{ 50, 90, 14, 78, 89, 56.5, 88.75 }
            },

            new Student
            {
            Firstname = "Petro0",
            Lastname = "Petrenko0",
            Middlename = "Petrovych0",
            Grades = new double[]{ 50, 90, 14, 78, 89, 56.5, 88.75 }
            },
        })
    {
        GroupName = "KP-02",
    }
    })
{
    StreamName = "KP-0x",
},

new CourseStream(new List<Group>
    {
    new Group(new List<Student>
        {
            new Student
            {
            Firstname = "Petro3",
            Lastname = "Petrenko3",
            Middlename = "Petrovych3",
            Grades = new double[]{ 50, 100, 14, 78, 89, 56.5, 88.75 }
            },
            new Student
```

```
                            {
                            Firstname = "Petro4",
                            Lastname = "Petrenko4",
                            Middlename = "Petrovych4",
                            Grades = new double[]{ 50, 90, 14, 78, 89, 56.5, 88.75 }
                            },

                            new Student
                            {
                            Firstname = "Petro4",
                            Lastname = "Petrenko4",
                            Middlename = "Petrovych4",
                            Grades = new double[]{ 50, 90, 14, 78, 89, 56.5, 88.75 }
                            },
                        })
                    {
                        GroupName = "KP-91",
                    },

                    new Group(new List<Student>
                        {
                            new Student
                            {
                            Firstname = "Petro1",
                            Lastname = "Petrenko1",
                            Middlename = "Petrovych1",
                            Grades = new double[]{ 50, 100, 14, 78, 89, 56.5, 88.75 }
                            },
                            new Student
                            {
                            Firstname = "Petro2",
                            Lastname = "Petrenko2",
                            Middlename = "Petrovych2",
                            Grades = new double[]{ 50, 90, 14, 78, 89, 56.5, 88.75 }
                            },
                            new Student
                            {
                            Firstname = "Petro0",
                            Lastname = "Petrenko0",
                            Middlename = "Petrovych0",
                            Grades = new double[]{ 50, 90, 14, 78, 89, 56.5, 88.75 }
                            },
                        })
                    {
                        GroupName = "KP-92",
                    }
                    })
                {
                    StreamName = "KP-9x",
                },

                })
            {
                CourseName = "Computer system engineering",
            },

            new Course(new List<CourseStream>
                {
                new CourseStream(new List<Group>
                    {
                    new Group(new List<Student>
                        {
                            new Student
            {
            Firstname = "Petro3",
            Lastname = "Petrenko3",
            Middlename = "Petrovych3",
            Grades = new double[]{ 50, 100, 14, 78, 89,  }
            },
            new Student
            {
            Firstname = "Petro4",
            Lastname = "Petrenko4",
            Middlename = "Petrovych4",
            Grades = new double[]{ 78, 89, 56.5, 88.75 }
            },

            new Student
            {
            Firstname = "Petro4",
            Lastname = "Petrenko4",
            Middlename = "Petrovych4",
            Grades = new double[]{ 50, 90, 14, 78, 89, 56.5, 88.75 }
            },
        })
    {
        GroupName = "KA-11",
    },

    new Group(new List<Student>
        {
            new Student
            {
            Firstname = "Petro1",
            Lastname = "Petrenko1",
            Middlename = "Petrovych1",
            Grades = new double[]{ 50, 100 }
            },
            new Student
            {
            Firstname = "Petro2",
            Lastname = "Petrenko2",
            Middlename = "Petrovych2",
            Grades = new double[]{ 50, 88.75 }
            },

            new Student
            {
            Firstname = "Petro0",
            Lastname = "Petrenko0",
            Middlename = "Petrovych0",
            Grades = new double[]{ 50, 90, 14, 78, 89, 56.5, 88.75 }
            },
        })
    {
        GroupName = "KA-12",
    }
    })
{
    StreamName = "KA-1x",
},

new CourseStream(new List<Group>
    {
    new Group(new List<Student>
        {
```

```
                        new Student
                        {
                        Firstname = "Petro3",
                        Lastname = "Petrenko3",
                        Middlename = "Petrovych3",
                        Grades = new double[]{ 50, 100, 56.5, 88.75 }
                        },
                        new Student
                        {
                        Firstname = "Petro4",
                        Lastname = "Petrenko4",
                        Middlename = "Petrovych4",
                        Grades = new double[]{ 50, 88.75 }
                        },

                        new Student
                        {
                        Firstname = "Petro4",
                        Lastname = "Petrenko4",
                        Middlename = "Petrovych4",
                        Grades = new double[]{ 88.75, 0, 15 }
                        },
                })
        {
                GroupName = "KA-01",
        },

        new Group(new List<Student>
                {
                        new Student
                        {
                        Firstname = "Petro1",
                        Lastname = "Petrenko1",
                        Middlename = "Petrovych1",
                        Grades = new double[]{ 50, 100, 14, 78, 89, 56.5, 88.75 }
                        },
                        new Student
                        {
                        Firstname = "Petro2",
                        Lastname = "Petrenko2",
                        Middlename = "Petrovych2",
                        Grades = new double[]{ 50, 90, 14, 78, 89, 56.5, 88.75 }
                        },

                        new Student
                        {
                        Firstname = "Petro0",
                        Lastname = "Petrenko0",
                        Middlename = "Petrovych0",
                        Grades = new double[]{ 50, 90, 14, 78, 89, 56.5, 88.75 }
                        },
                })
        {
                GroupName = "KA-02",
        }
        })
{
        StreamName = "KA-0x",
},

new CourseStream(new List<Group>
        {
        new Group(new List<Student>
                        {
                            new Student
                            {
                            Firstname = "Petro3",
                            Lastname = "Petrenko3",
                            Middlename = "Petrovych3",
                            Grades = new double[]{ 50, 100, 14, 78, 89, 56.5, 88.75 }
                            },
                            new Student
                            {
                            Firstname = "Petro4",
                            Lastname = "Petrenko4",
                            Middlename = "Petrovych4",
                            Grades = new double[]{ 50, 90, 14, 78, 89, 56.5, 88.75 }
                            },

                            new Student
                            {
                            Firstname = "Petro4",
                            Lastname = "Petrenko4",
                            Middlename = "Petrovych4",
                            Grades = new double[]{ 50, 90, 14, 78, 89, 56.5, 88.75 }
                            },
                        })
                    {
                        GroupName = "KA-91",
                    },

                    new Group(new List<Student>()
                        {
                            new Student
                            {
                            Firstname = "Petro1",
                            Lastname = "Petrenko1",
                            Middlename = "Petrovych1",
                            Grades = new double[]{ 50, 100, 14, 88.75 }
                            },
                            new Student
                            {
                            Firstname = "Petro2",
                            Lastname = "Petrenko2",
                            Middlename = "Petrovych2",
                            Grades = new double[]{ 89, 56.5, 88.75 }
                            },

                            new Student
                            {
                            Firstname = "Petro0",
                            Lastname = "Petrenko0",
                            Middlename = "Petrovych0",
                            Grades = new double[]{ 14, 78, 56.5, 88.75 }
                            },
                        })
                    {
                        GroupName = "KA-92",
                    }
                    })
                {
                    StreamName = "KA-9x",
                },

                })
            {
                CourseName = "Applied Math",
            },
        })
        {
        FacultyName = "FPM",
        };
    }

    static void Main(string[] args)
    {
        var FAM = InitFAM();
        FAM.Display(3);
    }
    }
}
```

## Console output

```
    Computer system engineering - 67.08
    KP-1x - 67.08
        KP-11 - 67.08
        Petro3 Petrovych3 Petrenko3 - 68.04
        Grades: 50 100 14 78 89 56.5 88.75

        Petro4 Petrovych4 Petrenko4 - 66.61
        Grades: 50 90 14 78 89 56.5 88.75

        Petro4 Petrovych4 Petrenko4 - 66.61
        Grades: 50 90 14 78 89 56.5 88.75


        KP-12 - 67.08
        Petro1 Petrovych1 Petrenko1 - 68.04
        Grades: 50 100 14 78 89 56.5 88.75

        Petro2 Petrovych2 Petrenko2 - 66.61
        Grades: 50 90 14 78 89 56.5 88.75

        Petro0 Petrovych0 Petrenko0 - 66.61
        Grades: 50 90 14 78 89 56.5 88.75



    KP-0x - 67.08
        KP-01 - 67.08
        Petro3 Petrovych3 Petrenko3 - 68.04
        Grades: 50 100 14 78 89 56.5 88.75

        Petro4 Petrovych4 Petrenko4 - 66.61
        Grades: 50 90 14 78 89 56.5 88.75
        Petro4 Petrovych4 Petrenko4 - 66.61
        Grades: 50 90 14 78 89 56.5 88.75


        KP-02 - 67.08
        Petro1 Petrovych1 Petrenko1 - 68.04
        Grades: 50 100 14 78 89 56.5 88.75

        Petro2 Petrovych2 Petrenko2 - 66.61
        Grades: 50 90 14 78 89 56.5 88.75

        Petro0 Petrovych0 Petrenko0 - 66.61
        Grades: 50 90 14 78 89 56.5 88.75



KP-9x - 67.08
        KP-91 - 67.08
        Petro3 Petrovych3 Petrenko3 - 68.04
        Grades: 50 100 14 78 89 56.5 88.75

        Petro4 Petrovych4 Petrenko4 - 66.61
        Grades: 50 90 14 78 89 56.5 88.75

        Petro4 Petrovych4 Petrenko4 - 66.61
        Grades: 50 90 14 78 89 56.5 88.75


        KP-92 - 67.08
        Petro1 Petrovych1 Petrenko1 - 68.04
        Grades: 50 100 14 78 89 56.5 88.75

        Petro2 Petrovych2 Petrenko2 - 66.61
        Grades: 50 90 14 78 89 56.5 88.75

        Petro0 Petrovych0 Petrenko0 - 66.61
        Grades: 50 90 14 78 89 56.5 88.75




Applied Math - 66.82
KA-1x - 70.31
        KA-11 - 70.29
        Petro3 Petrovych3 Petrenko3 - 66.20
        Grades: 50 100 14 78 89

        Petro4 Petrovych4 Petrenko4 - 78.06
        Grades: 78 89 56.5 88.75

        Petro4 Petrovych4 Petrenko4 - 66.61
        Grades: 50 90 14 78 89 56.5 88.75


        KA-12 - 70.33
        Petro1 Petrovych1 Petrenko1 - 75.00
        Grades: 50 100

        Petro2 Petrovych2 Petrenko2 - 69.38
        Grades: 50 88.75

        Petro0 Petrovych0 Petrenko0 - 66.61
```

```
                Grades: 50 90 14 78 89 56.5 88.75



        KA-0x - 63.17
                KA-01 - 59.26
                Petro3 Petrovych3 Petrenko3 - 73.81
                Grades: 50 100 56.5 88.75

                Petro4 Petrovych4 Petrenko4 - 69.38
                Grades: 50 88.75

                Petro4 Petrovych4 Petrenko4 - 34.58
                Grades: 88.75 0 15


                KA-02 - 67.08
                Petro1 Petrovych1 Petrenko1 - 68.04
                Grades: 50 100 14 78 89 56.5 88.75

                Petro2 Petrovych2 Petrenko2 - 66.61
                Grades: 50 90 14 78 89 56.5 88.75

                Petro0 Petrovych0 Petrenko0 - 66.61
                Grades: 50 90 14 78 89 56.5 88.75



        KA-9x - 66.97
                KA-91 - 67.08
                Petro3 Petrovych3 Petrenko3 - 68.04
                Grades: 50 100 14 78 89 56.5 88.75

                Petro4 Petrovych4 Petrenko4 - 66.61
                Grades: 50 90 14 78 89 56.5 88.75

                Petro4 Petrovych4 Petrenko4 - 66.61
                Grades: 50 90 14 78 89 56.5 88.75


                KA-92 - 66.86
                Petro1 Petrovych1 Petrenko1 - 63.19
                Grades: 50 100 14 88.75

                Petro2 Petrovych2 Petrenko2 - 78.08
                Grades: 89 56.5 88.75

                Petro0 Petrovych0 Petrenko0 - 59.31
                Grades: 14 78 56.5 88.75





D:\MyFiles\Prog\OOP_CSharp_KPI_Labs\Lab3\Part1\bin\Debug\net5.0\Part1.exe (процесс 12172) завершил работу
с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно…
```

## Task 2

### AccessRquiredOperatorProxy.cs

```
using System;

namespace Part2_Proxy.Models
{
        public class AccessRquiredOperatorProxy : Operator
        {
        private readonly FullAccessOperator _operator;

        public AccessRquiredOperatorProxy(FullAccessOperator oper)
        {
                _operator = oper;
        }

        public override bool ChangeClientsTariff(Client client)
        {
                if (GetPUK() == client.PUK_code)
                {
                return _operator.ChangeClientsTariff(client);
                }
                else
                {
                Console.WriteLine("Invalid PUK!");
                }

                return false;
        }

        protected override void DisplayTariffs()
        {
                throw new InvalidOperationException();
        }

        private string GetPUK()
        {
                Console.Write($"Enter PUK:\n>");
                return Console.ReadLine();
        }
        }
}
```

### Client.cs

```
using System;

namespace Part2_Proxy.Models
{
        public class Client
        {
        public Guid Id { get; } = Guid.NewGuid();
        public TariffPlan CurrentTariffPlan { get; set; }
        public decimal Balance { get; set; }
        public string PhoneNumber { get; set; }
        public string PUK_code { get; set; }

        public override string ToString()
        {
                return $"{nameof(Id)}: {Id}; {nameof(Balance)}: {Balance};\n{nameof(CurrentTariffPlan)}:
{CurrentTariffPlan};";
        }
        }
}
```

### FullAccessOperator.cs

```
using System;
using System.Collections.Generic;

namespace Part2_Proxy.Models
{
        public class FullAccessOperator: Operator
        {
        private IDictionary<int, TariffPlan> _operatorTariffPlans => new Dictionary<int, TariffPlan>
        {
                { 1, new TariffPlan(){ Name = "Super NET Start", Price = 200 } },
                { 2, new TariffPlan(){ Name = "Super NET Advanced", Price = 300 } },
                { 3, new TariffPlan(){ Name = "MLS ss 123", Price = 50 } },
                { 4, new TariffPlan(){ Name = "5G Net Super", Price = 280 } }
        };

        protected override void DisplayTariffs() // чи краще зробити метод з нелогічним передаванням
        {
                Console.WriteLine("Avaliable tariffs");
                foreach (var item in _operatorTariffPlans)
                {
                Console.WriteLine($"{item.Key} - {item.Value}");
                }
                Console.WriteLine();
        }

        public override bool ChangeClientsTariff(Client client)
        {
                DisplayTariffs();

                var tariffToChangeOn = _operatorTariffPlans[GetTariffId()];

                decimal balanceAfterPurchase = client.Balance - tariffToChangeOn.Price;
                bool isEnoughMoney = balanceAfterPurchase >= 0;

                if (isEnoughMoney)
                {
                client.CurrentTariffPlan = tariffToChangeOn;
                client.Balance = balanceAfterPurchase;
                Console.WriteLine($"Succeed! Current tariff is {client.CurrentTariffPlan}");
                }
                else
                {
                Console.WriteLine("There is not enough money for this tariff!");
                }
```

```
		return isEnoughMoney;
	}
	}
}
```

## Operator.cs

```
using System;

namespace Part2_Proxy.Models
{
	public abstract class Operator
	{
	protected virtual void DisplayTariffs()
	{
		throw new InvalidOperationException();
	}
	protected virtual int GetTariffId()
	{
		Console.Write("Enter tariff id> ");
		return int.Parse(Console.ReadLine());
	}
	public abstract bool ChangeClientsTariff(Client client);
	}
}
```

## TariffPlan.cs

```
namespace Part2_Proxy.Models
{
	public class TariffPlan
	{
	public string Name { get; set; }
	public decimal Price { get; set; }
	public override string ToString()
	{
		return $"Name: {Name}; Price: {Price}";
	}
	}
}
```

## Program.cs

```
using Part2_Proxy.Models;
using System;

namespace Part2_Proxy
{
       class Program
       {
       private static Client InitCurrentClient()
       {
              return new Client()
              {
              PhoneNumber = "000-111-88-99",
              Balance = 100,
              CurrentTariffPlan = null,
              PUK_code = "123"
              };
       }
       static void Main(string[] args)
       {
              var currentClient = InitCurrentClient();

              Console.WriteLine(currentClient);

              Operator oper = new AccessRquiredOperatorProxy(new FullAccessOperator());

              oper.ChangeClientsTariff(currentClient);
              Console.WriteLine(currentClient);
       }
       }
}
```

## Console output - Operator oper = new AccessRquiredOperatorProxy(new FullAccessOperator());

```
Id: faee5073-f8f7-466b-9243-ada960e746c0; Balance: 100;
CurrentTariffPlan: ;
Enter PUK:
>123
Avaliable tariffs
1 - Name: Super NET Start; Price: 200
2 - Name: Super NET Advanced; Price: 300
3 - Name: MLS ss 123; Price: 50
4 - Name: 5G Net Super; Price: 280

Enter tariff id> 3
Succeed! Current tariff is Name: MLS ss 123; Price: 50
Id: faee5073-f8f7-466b-9243-ada960e746c0; Balance: 50;
CurrentTariffPlan: Name: MLS ss 123; Price: 50;

D:\MyFiles\Prog\OOP_CSharp_KPI_Labs\Lab3\Part2_Proxy\bin\Debug\net5.0\Part2_Proxy.exe (процесс 6076)
завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно…
```

## Console output - Operator oper = new FullAccessOperator();

```
Id: a0d76cd8-342a-4eae-a96d-d9e5e736c811; Balance: 100;
CurrentTariffPlan: ;
Avaliable tariffs
```

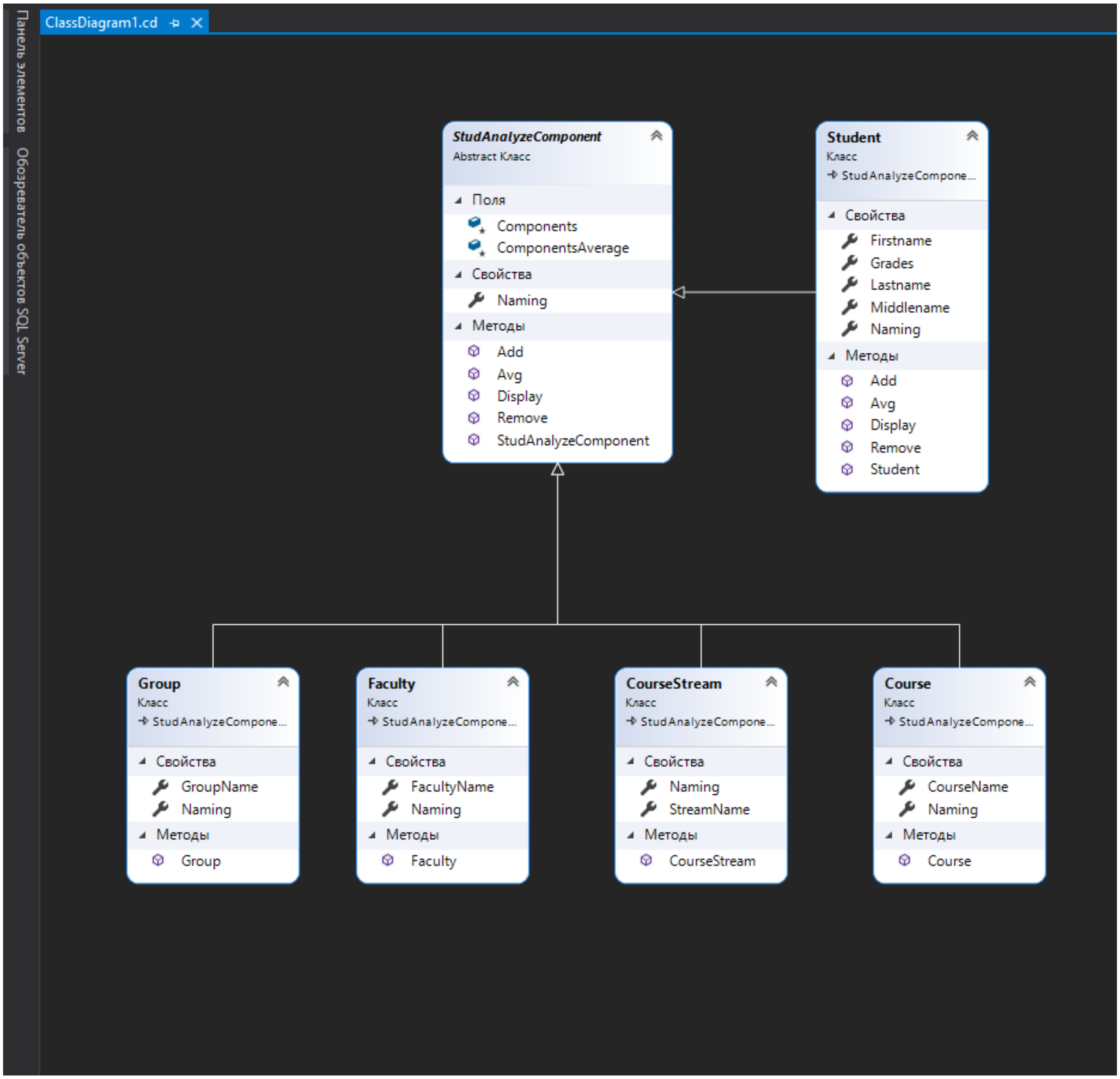
```
1 - Name: Super NET Start; Price: 200
2 - Name: Super NET Advanced; Price: 300
3 - Name: MLS ss 123; Price: 50
4 - Name: 5G Net Super; Price: 280

Enter tariff id> 3
Succeed! Current tariff is Name: MLS ss 123; Price: 50
Id: a0d76cd8-342a-4eae-a96d-d9e5e736c811; Balance: 50;
CurrentTariffPlan: Name: MLS ss 123; Price: 50;

D:\MyFiles\Prog\OOP_CSharp_KPI_Labs\Lab3\Part2_Proxy\bin\Debug\net5.0\Part2_Proxy.exe (процесс 9504)
завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно…
```

**Console output**

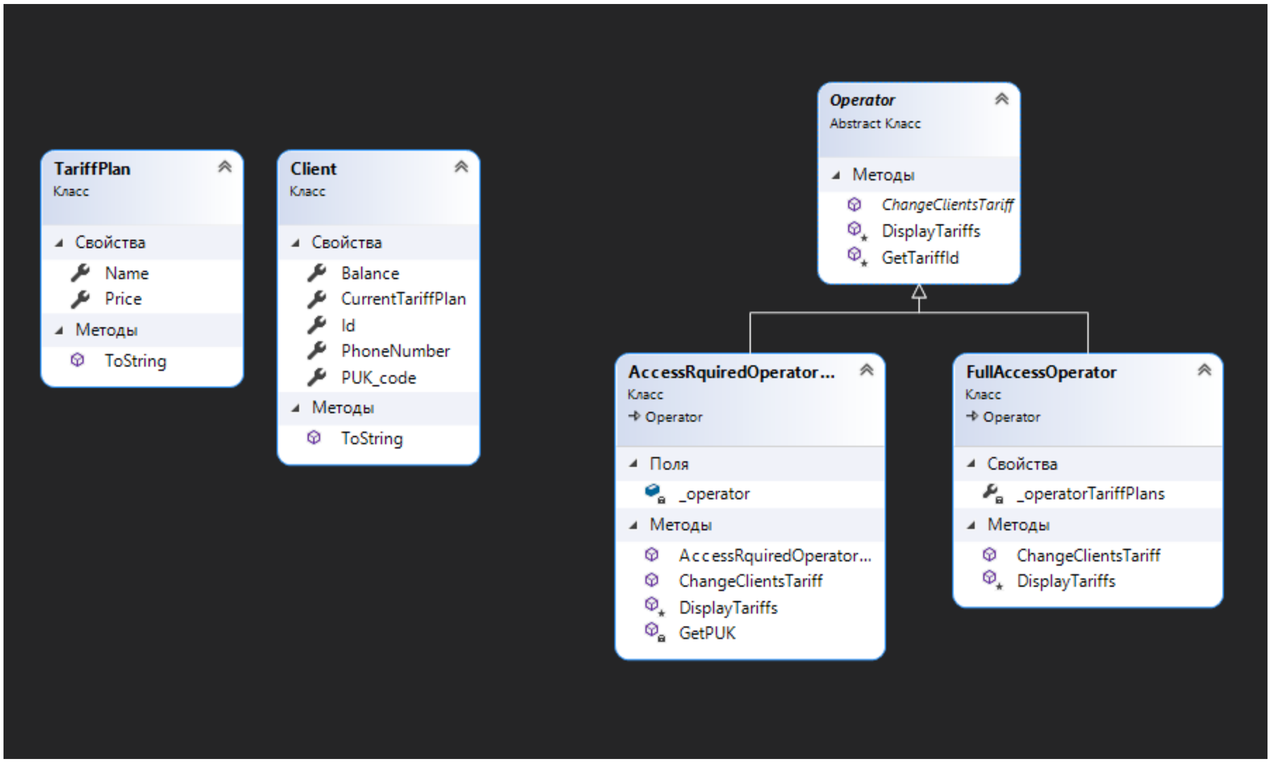
```
Id: 1282aba8-f516-436c-9608-8acd7be292e9; Balance: 100;
CurrentTariffPlan: ;
Enter PUK:
>123
Avaliable tariffs
1 - Name: Super NET Start; Price: 200
2 - Name: Super NET Advanced; Price: 300
3 - Name: MLS ss 123; Price: 50
4 - Name: 5G Net Super; Price: 280

Enter tariff id> 4
There is not enough money for this tariff!
Id: 1282aba8-f516-436c-9608-8acd7be292e9; Balance: 100;
CurrentTariffPlan: ;

D:\MyFiles\Prog\OOP_CSharp_KPI_Labs\Lab3\Part2_Proxy\bin\Debug\net5.0\Part2_Proxy.exe (процесс 8964)
завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно…
```

## UML-діаграми

1 Задача


Панель элементов
Обозреватель объектов SQL Server
ClassDiagram1.cd
StudAnalyzeComponent
Abstract Класс
Поля
Components
ComponentsAverage
Свойства
Naming
Методы
Add
Avg
Display
Remove
StudAnalyzeComponent
Student
Класс
StudAnalyzeCompone...
Свойства
Firstname
Grades
Lastname
Middlename
Naming
Методы
Add
Avg
Display
Remove
Student
Group
Класс
StudAnalyzeCompone...
Свойства
GroupName
Naming
Методы
Group
Faculty
Класс
StudAnalyzeCompone...
Свойства
FacultyName
Naming
Методы
Faculty
CourseStream
Класс
StudAnalyzeCompone...
Свойства
Naming
StreamName
Методы
CourseStream
Course
Класс
StudAnalyzeCompone...
Свойства
CourseName
Naming
Методы
Course

2 Задача



## Висновок

Виконавши дану роботу, ми познайомилися з такими структурними шаблонами, як Proxy та Composite. Проаналізували пробеми, що були надані задачами й реалізували шаблони на практиці.