Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра Информатики

Дисциплина «Программирование»

**ОТЧЕТ**

к лабораторной работе № 8

на тему:

**«Полиморфизм»**

БГУИР 6-05-0612-02 124

|  |
| --- |
| Выполнил студент группы 453504  ЯРЦЕВ Александр Александрович |
|  |
| (дата, подпись студента) |
| Проверил ассистент каф. Информатики  РОМАНЮК Максим Валерьевич |
|  |
| (дата, подпись преподавателя) |

Минск 2025

**1 ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ**

**Задание 1. Вариант 5.**

– Составить диаграмму классов проектируемой системы.

– Запрограммировать классы в соответствии с новой диаграммой.

– Проиллюстрировать использование интерфейсов.

– Показать вызов метода интерфейса через интерфейсную ссылку.

– Применить в программе шаблон проектирования Strategy.

Предметная область: Банк. Система хранит информацию о вкладчиках и сделанных ими вкладах. Класс “вкладчик” содержит имя вкладчика и величину вклада. Некоторым вкладчикам при создании вклада на счет может дополнительно перечисляться фиксированная сумма. В классе банк реализовать методы добавления нового вкладчика и метод вычисления общей суммы вкладов.

# 2 Выполнение работы

Для выполнения задания были созданы классы и интерфейсы, организованные в проекте (см. рисунок 1).

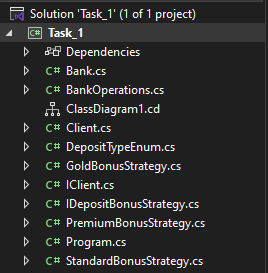


Рисунок 1 – Обозреватель решений

Для реализации предметной области был создан класс Client.

Класс Client содержит поля Name, DepositType, DepositAmount и реализует интерфейс IClient. Конструктор принимает стратегию начислению бонуса (IDepositBonusStrategy) и добавляет её результат к начальному вкладу.

public class Client : IClient

{

public string Name { get; }

public DepositTypeEnum DepositType { get; }

public decimal DepositAmount { get; private set; }

public Client(string name, DepositTypeEnum depositType, decimal initialDeposit, IDepositBonusStrategy bonusStrategy)

{

Name = name;

DepositType = depositType;

DepositAmount = initialDeposit + bonusStrategy.CalculateBonus(initialDeposit);

}

public void AddToDeposit(decimal amount)

{

if (amount > 0) DepositAmount += amount;

}

}

Теперь рассмотрим реализацию шаблона проектирование Strategy. Он реализован через интерфейс IDepositBonusStrategy и классы StandardBonusStrategy, PremiumBonusStrategy, GoldBonusStrategy. Интерфейсы определяют метод CalculateBonus для расчёта бонуса при создании вклада.

public interface IDepositBonusStrategy

{

decimal CalculateBonus(decimal initialDeposit);

}

public class PremiumBonusStrategy : IDepositBonusStrategy

{

public decimal CalculateBonus(decimal initialDeposit) => 500m;

}

Рассмотрим класс Bank. Этот класс управляет клиентами, храня их в коллекции List<IClient>. Основные методы: AddClient – добавляет нового клиента, TryAddToClientDeposit – пополняет вклад через интерфейсную ссылку IClient, CalculateTotalDeposits – вычисляет общую сумму вкладов.

public class Bank

{

private readonly List<IClient> clients = new List<IClient>();

public bool AddClient(string name, DepositTypeEnum depositType, decimal initialDeposit)

{

if (ClientExists(name) || initialDeposit <= 0) return false;

clients.Add(new Client(name, depositType, initialDeposit, bonusStrategies[depositType]));

return true;

}

public bool TryAddToClientDeposit(string name, decimal amount, out string message)

{

IClient client = clients.FirstOrDefault(c => c.Name.Equals(name, StringComparison.OrdinalIgnoreCase));

if (client == null)

{

message = $"Клиент '{name}' не найден.";

return false;

}

if (amount <= 0)

{

message = "Сумма пополнения должна быть положительной.";

return false;

}

client.AddToDeposit(amount); // Вызов через интерфейс

message = null;

return true;

}

public decimal CalculateTotalDeposits()

{

return clients.Sum(client => client.DepositAmount);

}

}

В структуре программы был создан класс BankOperations. Этот класс реализует логику пользовательского интерфейса, вызывая методы Bank для добавления клиентов и пополнения вкладов.

При выполнении работы была создана диаграмма классов для лучшего понимания взаимодействия между классами (см. рисунок 2).

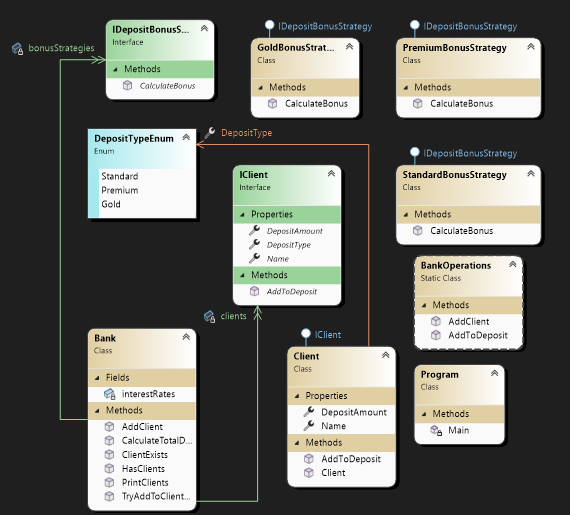


Рисунок 2 – Диаграмма классов

# Вывод

В ходе лабораторной работы были изучены принципы использования интерфейсов, вызов методов через интерфейсные ссылки и применение шаблона проектирования Strategy. Освоена разработка классов в соответствии с диаграммой классов и реализована банковская система согласно требованиям задания.