

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.05- 05.02/2/125.00.1.Б/ОК14- 2024
	Екземпляр № 1	Арк __ / 56

Лабораторна робота №10. Абстрактні класи та інтерфейси

Мета роботи: Освоїти використання абстрактних класів для визначення спільних властивостей і методів, а також інтерфейсів для додавання специфічної поведінки до об'єктів.

Теоретичні відомості

Абстрактні класи у C# дозволяють задавати базову структуру для інших класів. Вони можуть містити як реалізовані методи, так і абстрактні методи без реалізації, які мають бути перевизначені в похідних класах. Такі класи оголошуються за допомогою ключового слова `abstract` і не можуть бути інстанційовані напряму, тобто неможливо створити об'єкт абстрактного класу. Абстрактні класи можуть також включати властивості, поля, конструктори й події, що забезпечує їхню універсальність. Вони використовуються в тих випадках, коли необхідно забезпечити спільну поведінку для кількох похідних класів, залишаючи можливість реалізувати деталі в кожному з них окремо.

```
public abstract class ElectronicDevice
{
    public string Brand { get; set; }
    public string Model { get; set; }

    public abstract void TurnOn();
    public abstract void TurnOff();
}

// Використання
public class SmartDevice : ElectronicDevice
{
    public bool IsConnectedToWiFi { get; set; }

    public override void TurnOn()
    {
        Console.WriteLine($"{Brand} {Model} увімкнено.");
    }

    public override void TurnOff()
    {
        Console.WriteLine($"{Brand} {Model} вимкнено.");
    }
}
```

Інтерфейси в C# є контрактами, що визначають набір методів, властивостей або подій, які обов'язково мають бути реалізовані класом, що цей інтерфейс імплементує. Інтерфейси не можуть містити реалізації методів (у версіях до C# 8.0) і не дозволяють оголошувати поля. Вони служать для створення єдиного набору вимог, які можуть реалізовуватись різними класами, забезпечуючи поліморфізм. Клас або структура можуть реалізовувати кілька інтерфейсів одночасно, що робить їх зручними для задач, де потрібно забезпечити багатофункціональність. Це також дозволяє об'єктам з різною ієрархією мати однаковий інтерфейс взаємодії.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.05- 05.02/2/125.00.1.Б/ОК14- 2024
	Екземпляр № 1	Арк __/57

```
public interface IChargeable
{
    int BatteryLevel { get; set; } // Властивість, яка дозволяє читати/змінювати значення.
    void Charge(int percentage); // Метод для реалізації логіки зарядки.
}

// Використання
public class SmartPhone : IChargeable
{
    public int BatteryLevel { get; set; }

    public void Charge(int percentage)
    {
        BatteryLevel += percentage;
        if (BatteryLevel > 100)
        {
            BatteryLevel = 100;
        }
        Console.WriteLine($"Смартфон заряджено до {BatteryLevel}%.");
    }
}
```

Різниця між абстрактними класами та інтерфейсами

Характеристика	Абстрактний клас	Інтерфейс
Оголошення	abstract class	interface
Наявність реалізації	Може містити як реалізовані, так і абстрактні методи.	Не містить реалізації методів (до C# 8.0).
Поля	Може містити поля.	Не може містити полів (до C# 8.0).
Успадкування	Дозволяє успадковувати лише один абстрактний клас.	Можна реалізовувати кілька інтерфейсів.

Зміст роботи

Згідно з обраним у лабораторній роботі 5 варіантом теми, необхідно виконати наступну дії. Всі дії відбуваються в репозиторії OOPWPFProject.

	Абстрактний клас	Інтерфейс	Методи/властивості інтерфейсу
1	AbstractPerson	IValidatable	Властивості: ValidationStatus Методи: bool IsValid(), string GetValidationMessage()
2	AbstractItem	IBorrowable	Властивості: IsBorrowed, Borrower Методи: void Borrow(string borrower), void Return()
3	AbstractTask	ICompletable	Властивості: IsCompleted Методи: void Complete(), void ResetTask(), string GetCompletionStatus()
4	AbstractActivity	ITrackable	Властивості: StartTime, EndTime Методи: void Start(), void Stop(), TimeSpan GetDuration()
5	AbstractOrder	IOrderTrackable	Властивості: OrderStatus, AssignedEmployee Методи: void UpdateStatus(string newStatus), bool IsDelivered()
6	AbstractEvent	IRemindable	Властивості: IsReminderSet, ReminderMessage Методи: void SetReminder(DateTime

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.05- 05.02/2/125.00.1.Б/ОК14- 2024
	Екземпляр № 1	Арк __ / 58

			time), void CancelReminder()
7	AbstractReservation	ICancelable	Властивості: IsCanceled Методи: void Cancel(), string GetReservationDetails()
8	AbstractContact	ITaggable	Властивості: Tags, Category Методи: void AddTag(string tag), void RemoveTag(string tag), bool HasTag(string tag)
9	AbstractEvent	ICancelable	Властивості: IsCanceled Методи: void Cancel(), void Reschedule(DateTime newDate), string GetEventSummary()
10	AbstractVehicle	IMaintainable	Властивості: LastMaintenanceDate, MaintenanceCost Методи: void ScheduleMaintenance(), bool NeedsService()
11	AbstractPlace	IReviewable	Властивості: Reviews, Rating Методи: void AddReview(string review), void RemoveReview(string review), double GetAverageRating()
12	AbstractInventoryItem	IMovable	Властивості: CurrentLocation, IsFragile Методи: void MoveTo(string location), string GetLocationHistory()
13	AbstractTransaction	ITrackable	Властивості: Balance, TransactionType Методи: void Track(), string GetTransactionDetails()
14	AbstractOrder	IOrderTrackable	Властивості: OrderState, AssignedCourier Методи: void UpdateStatus(), bool IsReadyForPickup()
15	AbstractMedicalRecord	IPrintable	Властивості: Report, PrintDate Методи: string GenerateReport(), void PrintReport()
16	AbstractUser	ILoginTrackable	Властивості: LastLoginDate, FailedAttempts Методи: void RecordLogin(), void RecordFailure()
17	AbstractShipment	ITrackable	Властивості: CurrentLocation, Status Методи: void UpdateLocation(string location), string GetTrackingInfo()
18	AbstractVisit	IPriceCalculable	Властивості: BasePrice, Discount Методи: double CalculateFinalPrice(), string GetPriceBreakdown()
19	AbstractRental	IReturnable	Властивості: IsReturned, ReturnDate Методи: void MarkReturned(), bool IsLateReturn()
20	AbstractMaterial	IAccessible	Властивості: AccessLevel, IsRestricted Методи: bool HasAccess(string userRole), void GrantAccess(string userRole)
21	AbstractMember	IActivityLoggable	Властивості: LastActivityDate, ActivityCount Методи: void LogActivity(string description), string GetActivitySummary()
22	AbstractCredit	IPayable	Властивості: PaymentDueDate, IsOverdue Методи: bool MakePayment(double amount), string GetPaymentStatus()
23	AbstractWine	IRateable	Властивості: Rating, Reviews Методи: void AddReview(string comment), double GetAverageRating()

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідас ДСТУ ISO 9001:2015	Ф-22.05- 05.02/2/125.00.1.Б/ОК14- 2024
	Екземпляр № 1	Арк __ / 59

24	AbstractMeasurement	ISyncable	Властивості: LastSynced, SyncStatus Методи: void Sync(), bool IsSynced()
25	AbstractTournament	IRegistrable	Властивості: RegisteredTeams, MaxSlots Методи: bool Register(string teamName), bool IsFull()
26	AbstractOrder	IPayable	Властивості: IsPaid, PaymentMethod Методи: void Pay(string method), string GetPaymentInfo()
27	AbstractUsageRecord	IBillable	Властивості: Cost, BillingDate Методи: double CalculateBill(), string GenerateInvoice()
28	AbstractWeatherRecord	IExportable	Властивості: ExportFormat, ExportDate Методи: string Export(), void ScheduleExport(DateTime date)
29	AbstractCampaign	IEvaluable	Властивості: PerformanceScore, Feedback Методи: double Evaluate(), string GetReport()
30	AbstractDevice	IDiagnosable	Властивості: IsFaulty, LastCheckDate Методи: void RunDiagnostics(), string GetStatusReport()

Завдання 1: Злити гілку feature/add- inheritance в основну гілку (master або main). Створити нові гілку

```
git checkout -b feature/add-abstraction
```

Завдання 2: На основі обраної теми, створити абстрактний клас, що міститиме властивості базового класу та абстрактний метод GetDetails.

Завдання 3: Модифікувати базовий клас, успадкувавши його від абстрактного класу. Реалізувати абстрактний метод в базовому класі.

Завдання 4: На основі обраної теми, створити інтерфейс з описаними властивостями та методами.

Завдання 5: Реалізувати інтерфейс у похідних класах. Потрібно успадкувати інтерфейс похідним класом, додати властивості та реалізувати методи.

Завдання 6: В інтерфейсі програми додати можливість виклику методу інтерфейсу. Наприклад, можливість додавати теги, встановлювати нагадування, тощо. Оновити відображення даних, додати виведення нових властивостей.

Завдання 7: Створити коміт та завантажити зміни в віддалений репозиторій

Контрольні запитання

1. Що таке абстрактний клас у C#?
2. У чому різниця між абстрактним класом і звичайним класом?
3. Як оголосити абстрактний метод у класі?
4. Що таке інтерфейс у C#?
5. Як інтерфейси відрізняються від абстрактних класів?
6. Чи може абстрактний клас реалізовувати інтерфейс? Поясніть з прикладом.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.05- 05.02/2/125.00.1.Б/ОК14- 2024
	<i>Екземпляр № 1</i>	<i>Арк __ / 60</i>

7. Чи може клас реалізовувати кілька інтерфейсів одночасно? Якщо так, як це робиться?
8. Чи можна створити об'єкт абстрактного класу? Якщо ні, то як використовувати абстрактний клас у програмі?
9. Чим відрізняється використання ключового слова `abstract` від `virtual` у методах класу?
10. Що станеться, якщо клас, який реалізує інтерфейс, не надасть реалізацію для всіх методів інтерфейсу?