Fields

- `const string BankCode = "BNK001";`
- A constant value representing the bank code (never changes).
- `readonly DateTime CreatedDate;`

A read-only field set only once in the constructor, stores account creation date.

- `private int _accountNumber;`

Stores the account number (manually set for now).

- Other private fields:
  - `_fullName`
  - `_nationalID`
  - `_phoneNumber`
  - `_address`
  - `_balance`

Properties (with Validation)

For each private field, create a public property with appropriate validation:

- `FullName`

Must not be null or empty.

- `NationalID`

Must be exactly **14 digits**.

- `PhoneNumber`

Must start with **"01"** and be **11 digits long**.

- `Balance`

Must be **greater than or equal to 0**.

- `Address`

Optional (no specific validation required).

Constructors

Implement **3 types of constructors**:

1. **Default constructor**

Assigns default values.

2. **Parameterized constructor**

Accepts full name, national ID, phone number, address, and balance.

3. **Overloaded constructor**

Accepts all values **except balance** (set it to `0` by default).

Methods

Create the following methods inside the class:

- `ShowAccountDetails()`

Prints account info (name, phone, balance, etc.) to the console.

- `IsValidNationalID()`

Returns `true` if the national ID is exactly 14 digits.

- `IsValidPhoneNumber()`

Returns `true` if the phone starts with "01" and is 11 digits.

Main Method Task

In your `Main()` method:

- Create **2 BankAccount objects** using different constructors.
- Call `ShowAccountDetails()` for each to display their info.