

Práctica: Creación de objetos

Objetivos

En esta práctica modificará la clase **BankAccount** creada en prácticas anteriores de forma que utilice constructores. Creará también una nueva clase, **BankTransaction**, y la utilizará para almacenar información sobre las transacciones (depósitos y retiradas de dinero) que se realicen en una cuenta.

Al final de esta práctica, usted será capaz de:

- Sustituir el constructor por defecto.
- Crear constructores sobrecargados.
- Inicializar datos **readonly**.

Requisitos previos

Antes de realizar la práctica debe estar familiarizado con los siguientes temas:

- Creación de clases y de instancias de objetos.
- Definición y llamadas a métodos.

Es necesario también haber realizado la práctica 8.1. Si no la terminó, puede usar el código de la solución.

Ejercicio 1

Uso de constructores

En este ejercicio modificará la clase **BankAccount** creada en prácticas anteriores. Eliminará los métodos que asignan valores a las variables de instancia para el número y el tipo de cuenta, y los sustituirá por una serie de constructores que se puedan usar al crear una instancia de **BankAccount**.

Tendrá que sustituir el constructor por defecto para generar un número de cuenta (usando la técnica empleada anteriormente), fijar el tipo de cuenta en **Checking** y poner el saldo a cero.

También creará otros tres constructores que recibirán distintas combinaciones de parámetros:

- El primero recibirá un **AccountType**. El constructor generará un número de cuenta, pondrá el saldo a cero y asignará al tipo de cuenta el valor recibido.
- El segundo recibirá un **decimal**. El constructor generará un número de cuenta, fijará el tipo de cuenta en **Checking** y asignará al saldo el valor recibido.
- El tercero recibirá un **AccountType** y un **decimal**. El constructor generará un número de cuenta y asignará al tipo de cuenta el valor del parámetro **AccountType** y al saldo el valor del parámetro **decimal**.

Cómo crear el constructor por defecto

1. Abra el proyecto Constructors.sln en la carpeta *Lab Files*(archivos de Starter\Constructors dentro del fichero lab09.zip.
2. Borre el método **Populate** en la clase **BankAccount**.
3. Cree un constructor por defecto de la siguiente manera:
 - a. El nombre debe ser **BankAccount**.
 - b. Debe ser público.
 - c. No debe recibir ningún parámetro.
 - d. No debe tener tipo de retorno.
 - e. El cuerpo del constructor debe generar un número de cuenta usando el método **NextNumber**, fijar el tipo de cuenta en **AccountType.Checking** e inicializar a cero el saldo de la cuenta.

El constructor finalizado debería ser como éste:

```
public BankAccount( )
{
    accNo = NextNumber( );
    accType = AccountType.Checking;
    accBal = 0;
}
```

Cómo crear los demás constructores

1. Añada otro constructor que reciba un solo parámetro **AccountType** llamado **aType**. Este constructor tendrá que:
 - a. Generar un número de cuenta, como antes.
 - b. Fijar *accType* en **aType**.
 - c. Poner *accBal* a cero.
2. Defina otro constructor que reciba un solo parámetro **decimal** llamado **aBal**. Este constructor tendrá que:
 - a. Generar un número de cuenta.
 - b. Fijar *accType* en **AccountType.Checking**.
 - c. Poner *accBal* a **aBal**.
3. Defina un último constructor que reciba dos parámetros: un **AccountType** llamado **aType** y un **decimal** llamado **aBal**. Este constructor tendrá que:
 - a. Generar un número de cuenta.
 - b. Fijar *accType* en **aType**.
 - c. Poner *accBal* a **aBal**.

El código finalizado para los tres constructores es el siguiente:

```
public BankAccount(AccountType aType)
{
    accNo = NextNumber( );
    accType = aType;
    accBal = 0;
}

public BankAccount(decimal aBal)
{
    accNo = NextNumber( );
    accType = AccountType.Checking;
    accBal = aBal;
}

public BankAccount(AccountType aType, decimal aBal)
{
    accNo = NextNumber( );
    accType = aType;
    accBal = aBal;
}
```

Cómo probar los constructores

1. En el método **Main** de la clase **CreateAccount**, defina cuatro variables **BankAccount** llamadas *acc1*, *acc2*, *acc3*, y *acc4*.
2. Cree una instancia de *acc1* usando el constructor por defecto.
3. Cree una instancia de *acc2* usando el constructor que recibe sólo un **AccountType**. Fije el tipo de *acc2* en **AccountType.Deposit**.
4. Cree una instancia de *acc3* usando el constructor que recibe sólo un saldo **decimal**. Ponga el saldo de *acc3* en 100.
5. Cree una instancia de *acc4* usando el constructor que recibe un **AccountType** y un saldo **decimal**. Fije el tipo de *acc4* en **AccountType.Deposit** y ponga el saldo en 500.
6. Use el método **Write** (incluido en la clase **CreateAccount**) para mostrar los contenidos de cada una de las cuentas. El código finalizado es el siguiente:

```
static void Main( )
{
    BankAccount acc1, acc2, acc3, acc4;

    acc1 = new BankAccount( );
    acc2 = new BankAccount(AccountType.Deposit);
    acc3 = new BankAccount(100);
    acc4 = new BankAccount(AccountType.Deposit, 500);

    Write(acc1);
    Write(acc2);
    Write(acc3);
    Write(acc4);
}
```

7. Compile el proyecto y corrija los posibles errores. Ejecútelo y compruebe que produce los resultados esperados.