# **Indexers**

Індексатори дозволяють індексувати екземпляри класів або структур так само, як масиви. Індексоване значення можна встановити або отримати без явного зазначення типу або члена екземпляра. Індексатори схожі на властивості, за винятком того, що їхні аксесори приймають параметри.

У наступному прикладі визначено узагальнений клас з простими методами доступу get та set для присвоєння та отримання значень. Клас Program створює екземпляр цього класу для зберігання рядків.

using System;

class SampleCollection<T>

{

// Declare an array to store the data elements.

private T[] arr = new T[100];

// Define the indexer to allow client code to use [] notation.

public T this[int i]

{

get { return arr[i]; }

set { arr[i] = value; }

}

}

class Program

{

static void Main()

{

var stringCollection = new SampleCollection<string>();

stringCollection[0] = "Hello, World";

Console.WriteLine(stringCollection[0]);

}

}

// The example displays the following output:

// Hello, World.

## **Expression Body Definitions**

## **(Визначення тіла виразу)**

Аксесор ***get*** або ***set*** індексатора зазвичай складається з одного оператора, який повертає або встановлює значення. Члени у вигляді виразів надають спрощений синтаксис для підтримки такого сценарію. Індексатор, доступний лише для читання, може бути реалізовано у вигляді члена з тілом виразу, як показано у наступному прикладі.

using System;

class SampleCollection<T>

{

// Declare an array to store the data elements.

private T[] arr = new T[100];

int nextIndex = 0;

// Define the indexer to allow client code to use [] notation.

public T this[int i] => arr[i];

public void Add(T value)

{

if (nextIndex >= arr.Length)

throw new IndexOutOfRangeException($"The collection can hold only {arr.Length} elements.");

arr[nextIndex++] = value;

}

}

class Program

{

static void Main()

{

var stringCollection = new SampleCollection<string>();

stringCollection.Add("Hello, World");

System.Console.WriteLine(stringCollection[0]);

}

}

// The example displays the following output:

// Hello, World.

Зверніть увагу, що ***=>*** вводить тіло виразу, а ключове слово ***get*** не використовується.

Аксесор ***get*** і ***set*** можуть бути реалізовані як члени з тілом виразу. У цьому випадку слід використовувати обидва ключові слова ***get*** і ***set***.

Наприклад:

using System;

class SampleCollection<T>

{

// Declare an array to store the data elements.

private T[] arr = new T[100];

// Define the indexer to allow client code to use [] notation.

public T this[int i]

{

get => arr[i];

set => arr[i] = value;

}

}

class Program

{

static void Main()

{

var stringCollection = new SampleCollection<string>();

stringCollection[0] = "Hello, World.";

Console.WriteLine(stringCollection[0]);

}

}

// The example displays the following output:

// Hello, World.

**Indexers Overview**

**(Огляд індексаторів)**

Індексатори дозволяють індексувати об'єкти подібно до масивів.

* Аксесор ***get*** повертає значення. Аксесор ***set*** присвоює значення.
* Ключове слово ***this*** використовується для визначення індексатора.
* Ключове слово ***value*** використовується для визначення значення, яке присвоюється аксесором ***set***.
* Індексатори не обов'язково повинні індексуватися цілочисельними значеннями; ви самі визначаєте конкретний механізм пошуку.
* Індексатори можуть бути перевантажені.
* Індексатори можуть мати більше одного формального параметра, наприклад, при доступі до двовимірного масиву.