



Microsoft Partner
Silver Learning

C# Essential

Колекції



ITVDN
IT VIDEO DEVELOPERS NETWORK

C# Essential

Після уроку обов'язково



Повторіть цей урок в відео форматі на [ITVDN.com](http://itvdn.com)



Перевірте як Ви засвоїли даний матеріал на [TestProvider.com](http://testprovider.com)

C# Essential

Тема

Колекції

Колекція

Collection

Колекція – це клас, призначений для угруповання пов'язаних об'єктів, управління ними та обробки їх в циклах.



Колекції є важливим інструментом програміста, але рішення про їх застосуванні не завжди виявляється очевидним.

Застосування колекцій

Use of Collections

Колекції варто застосовувати, якщо:

- Окремі елементи використовуються для однакових цілей і однаково важливі.
- На момент компіляції число елементів не відомо чи не зафіксовано.
- Необхідна підтримка операції перебору всіх елементів.
- Необхідна підтримка упорядкування елементів.
- Необхідно використовувати елементи з бібліотеки, від якої споживач очікує наявності типу колекції.

Інтерфейс

IEnumerable

Методи інтерфейсу **IEnumerable**:

IEnumerator GetEnumerator() – повертає нумератор, який можна використовувати для навігації по колекції.

Інтерфейс

IEnumerator

Властивості інтерфейсу **IEnumerator**:

object Current { **get**; } – повертає поточний елемент колекції.

Методи інтерфейсу **IEnumerator**:

bool MoveNext() – переміщує нумератор на наступний елемент колекції.

void Reset() – переміщує нумератор на початок колекції.

Ключове слово

yield

- Блок, в якому міститься ключове слово `yield`, розцінюється компілятором, як блок ітератора.
- Ключове слово `return` використовується для надання значення об'єкту перечіслювача.
- Ключове слово `break` використовується для позначення кінця ітерації.

```
public static IEnumerable Power()  
{  
    yield return "Hello world!";  
}
```

```
public static IEnumerable Power()  
{  
    yield break;  
}
```


Циклічна конструкція

foreach

Циклічна конструкція **foreach** дозволяє виконувати навігацію по колекції, використовуючи реалізації інтерфейсів **IEnumerable** та **IEnumerator**.

```
foreach (var element in myCollection)
{
}
```



var — локальна змінна з неявним типом має строгу типізацію, так як би тип був заданий явно, тільки тип визначає компілятор.

Інтерфейс

IEnumerable<T>

IEnumerable<T> – успадкований від **IEnumerable**

Методи інтерфейсу **IEnumerable<T>**:

IEnumerator<T> GetEnumerator() – повертає узагальнений нумератор, який можна використовувати для навігації по колекції.

C# Essential

Q&A

Інформаційний відеосервіс для розробників програмного забезпечення

