

Projection operators

Projection refers to the operation of transforming an object into a new form that often consists only of those properties subsequently used. By using projection, you can construct a new type that is built from each object. You can project a property and perform a mathematical function on it. You can also project the original object without changing it.

Methods:

1. Select
2. SelectMany
3. ZIP

1. Select :

Project values that are based on the transform function.

Example:

```
List<string> words = ["an", "apple", "a", "day"];
```

```
var query = from word in words
            select word.Substring(0, 1);
```

```
foreach (string s in query)
{
    Console.WriteLine(s);
}
```

/* This code produces the following output:

```
a
a
a
d
*/
```

The equivalent query using method syntax is shown in the following code:

```
List<string> words = ["an", "apple", "a", "day"];
```

```
var query = words.Select(word => word.Substring(0, 1));
```

```
foreach (string s in query)
{
```

```
    Console.WriteLine(s);  
}
```

/* This code produces the following output:

```
a  
a  
a  
d  
*/
```

2. SelectMany

Projects sequences of values that are based on a transform function and then flattens them into one sequence.

The following example uses multiple `from` clauses to project each word from each string in a list of strings.

```
List<string> phrases = ["an apple a day", "the quick brown fox"];
```

```
var query = from phrase in phrases  
            from word in phrase.Split(' ')  
            select word;
```

```
foreach (string s in query)  
{  
    Console.WriteLine(s);  
}
```

/* This code produces the following output:

```
an  
apple  
a  
day  
the  
quick  
brown  
fox  
*/
```

