

Working with TuplesExtensions

Short functions for tuples creation

There are several short functions for value tuples creation. Suppose you want to create the tuple which consists of two items. So you must write the following code:

```
{$reference NETSquirrel.dll}  
uses TuplesExtensions;  
  
begin  
    ValueTupleOf(1, 2).PrintLine();  
end.
```

The output is: (1, 2)

Joining tuples and singular values

You can use the overloaded + operator to join tuples and singular values such as shown below in code:

```
{$reference NETSquirrel.dll}  
uses TuplesExtensions;  
  
begin  
    (ValueTupleOf(1, 2) + 3).PrintLine();  
end.
```

The output is: (1, 2, 3)

Notice that you can't join the singular value and some tuple. The operator + arguments order is important.

Tuples step by step comparison

If you want to compare your tuples step by step you can accomplish as following code shows:

```
{$reference NETSquirrel.dll}  
uses TuplesExtensions;
```

```
begin
  (ValueTupleOf(1, 2) < ValueTupleOf(3, 4)).PrintLine();
end.
```

The output is: **true**

Pay attention at that comparison is proceeded over corresponding items.

Value tuples into reference tuples implicit conversions

Suppose you use twice tuples types: value and reference ones. You may want to convert reference tuple to value one or do the opposite task. So you can use the following decision:

```
{$reference NETSquirrel.dll}
uses TuplesExtensions;

begin
  var x: (integer, integer) := ValueTupleOf(1, 2);
end.
```

Note that if the first range side is more than the second one then the empty sequence is returned.

For proposals: MrFresnel@yandex.ru.