aardio 范例: 在 aardio 中自动支持 .NET 泛型 - ValueTuple 示例

```
//在 aardio 中自动支持 .NET 泛型 - ValueTuple 示例
import console;
import dotNet;

/*
ValueTuple 要 .NET 4.7 以上才能支持（ Windows 10 1703 开始自带.Net 4.7 ）。
也可以自行下载 System.ValueTuple.dll 并通过 dotNet.reference() 函数引入，可支持 .NET 4.4 以及更新版本。
*/
import System.ValueTuple;

/*
创建 ValueTuple 对象，最多支持 8 个参数。
注意这是一个典型的泛型函数，aardio 可以通过参数类型自动匹配泛型函数。
*/
var tuple = System.ValueTuple.Create(
    21,22,23,24,25,26,27,
    dotNet.double(28)
    /*
    可选用以下函数明确声明参数类型:
    dotNet.object,dotNet.byte,dotNet.ubyte,dotNet.word,dotNet.uword
    ,dotNet.int,dotNet.uint,dotNet.long,dotNet.float,dotNet.double
    参考「 编译 / 类型」范例。
    */
);

console.log("查看对象类型",tuple.GetType())

/*
ValueTuple 最多只能有7个元素，
可使用字段 Item1, Item2, Item3, Item4, Item5, Item6, Item7 访问。
*/
console.log(tuple) //console.log 会自动调用 tostring(tuple)
console.log(tuple.Item1) //返回第 1 个元素的值
console.log(tuple.Item2) //返回第 2 个元素的值

tuple.Item7 = 123; //可以修改值，并不是只读的
console.log(tuple.Item7) //返回第 7 个元素的值

//那如果有第 8 个元素怎么办呢？通过 Rest 字段放到下一个 ValueTuple 里。
console.log(tuple.Rest) //调用 tostring(tuple.Rest)
console.log(tuple.Rest.Item1) //返回第 8 个元素的值
/*
实际上你没有必要使用超过 8 个元素的元组。
这世上任何事都有上限，有边界，没有无所不能。
*/

//遍历 tuple
for i,v in System.ValueTuple.each(tuple){
    console.log(i,v);
}

console.pause();

/*
// ValueTuple C# 示例
namespace TestValueTuple
{
    public class Class1
    {
        public static (double, int, int, int, int, int, int, int, int, int, int) tupleValue = (12.3, 22, 22, 22, 22, 22, 22, 22, 22, 22, 22);
        public static (double, int) GetTupleValue()
        {
            return (12.3, 22);
        }
    }
}
*/
```

Markdown 格式