

Тип «StringOperations» в Mobile SMARTS

Последние изменения: 2024-03-26

StringOperations – тип, содержащий методы, позволяющие работать со строками (обрезать, разбивать, форматировать и т.д.).

Доступные методы:

Имя метода
Описание
FillToWidth(s1, s2, totalWidth, c)
Объединяет две строки с разделителем между ними, дополняя итоговую строку до нужной длины
FromBase64(valueBase64)
Конвертирует строку из Base64 в обычную форму
Join(source, c)
Объединяет коллекцию строк, добавляя разделитель между ними
PadCenter(s, totalWidth, c)
Дополняет строку с обеих сторон до нужной длины, если она короче
PadLeft(s, totalWidth, c)
Дополняет строку слева до нужной длины, если она короче
PadRight(s, totalWidth, c)
Дополняет строку справа до нужной длины, если она короче
ParseByTemplate(value, template)
Позволяет распарсить строку по шаблону
Split(source, oneSymbolSplitter)
Разбивает строку на коллекцию подстрок согласно односимвольному разделителю. Если передан более длинный разделитель, то используется только первый символ
ToBase64(str)
Конвертирует строку в Base64 строку
WordWrap(text, width)
Форматирует строку переносами по словам под заданную длину строки

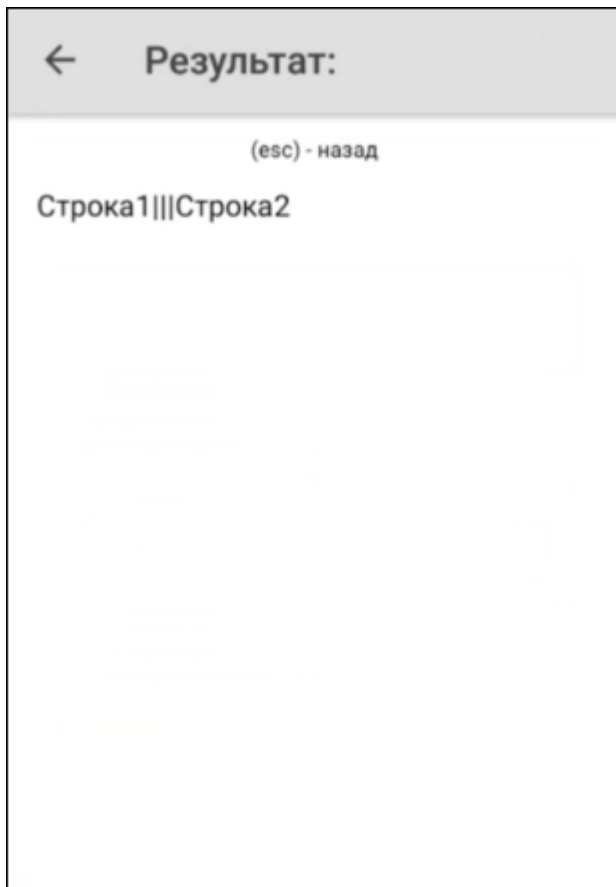
Примеры использования

FillToWidth

Код:

```
Строка = StringOperations.FillToWidth("Строка1","Строка2",17,"|");
```

Результат:



FromBase64

Код:

```
Строка = StringOperations.FromBase64("V29yZA==");
```

Строка «V29yZA==» расшифровывается как «Word»

Результат:

←

Результат:

(esc) - назад

Изначальная строка:

V29yZA==

Конвертированная строка:

Word

PadCenter

Код:

```
Строка = StringOperations.PadCenter("Текст",12,"|");
```

Очередность добавления символа: сначала справа, потом слева.

Результат:

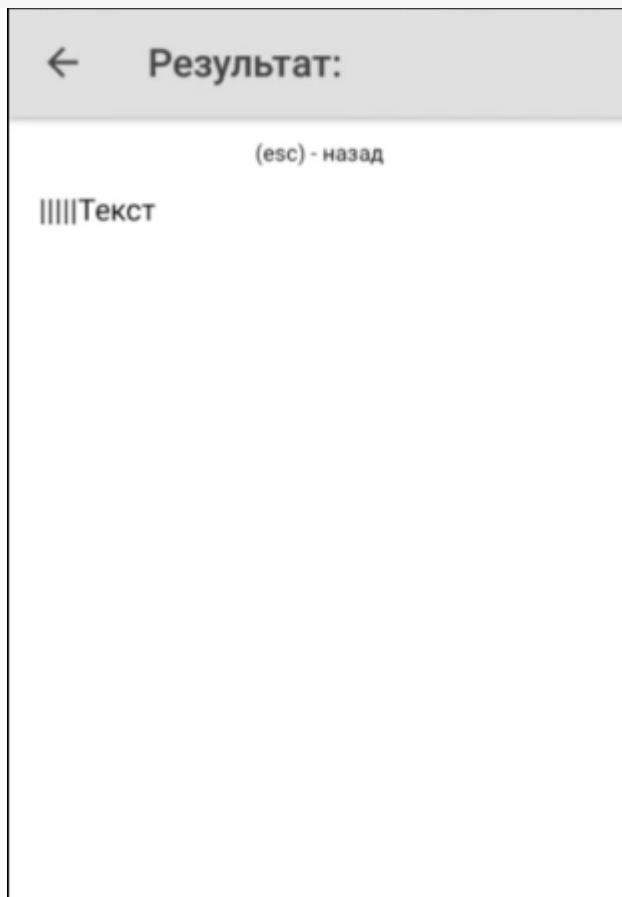


PadLeft

Код:

```
Строка = StringOperations.PadLeft("Текст",10,"|");
```

Результат:



PadRight

Код:

```
Строка = StringOperations.PadRight("Текст",10,"|");
```

Результат:

←

Результат:

(esc) - назад

Текст|||||

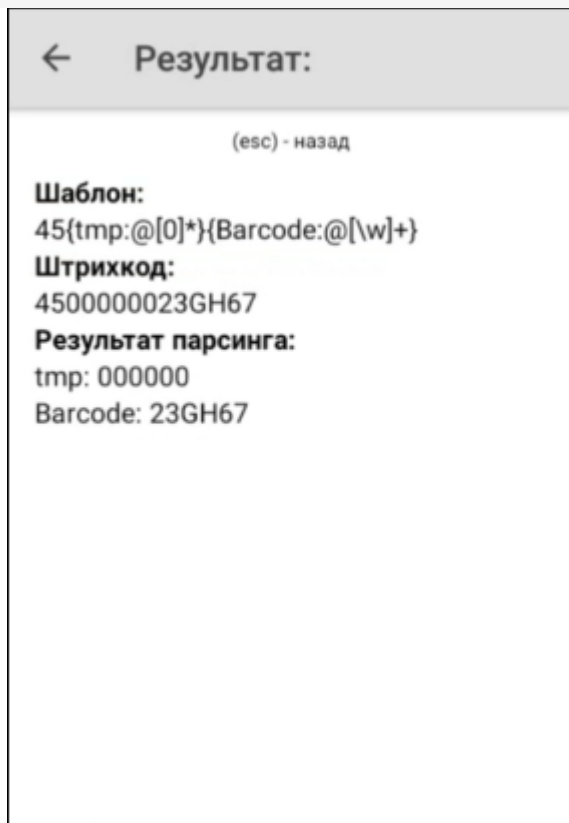
ParseByTemplate

Метод поддерживает любые шаблоны в конфигурации (regex, ячеек, номенклатуры и т.д.). Более подробно про регулярные выражения можно прочитать [тут](#).

Код для регулярных выражений:

```
Строка = StringOperations.ParseByTemplate("4500000023GH67","45{tmp:@{0}*}{Barcode:@{\\w}+}")
```

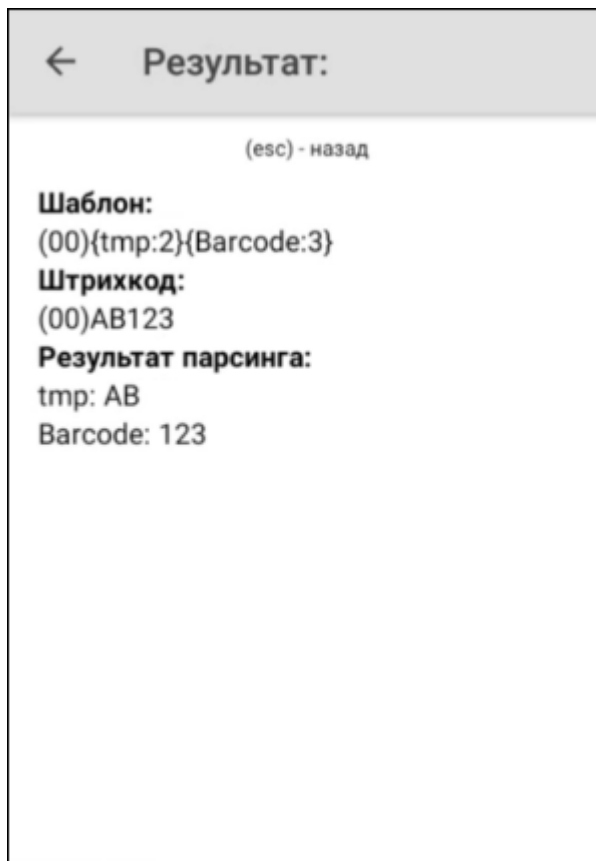
Результат:



Код для штрихкода:

```
Строка = StringOperations.ParseByTemplate("(00)AB123","(00){tmp:2}{Barcode:3}");
```

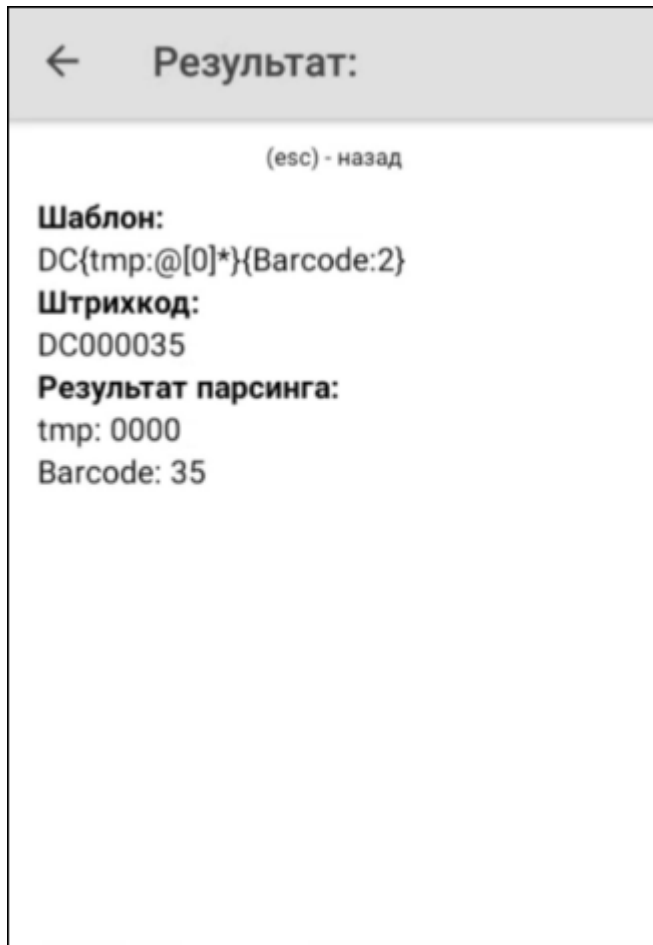
Результат:



Код для совмещённого случая:

```
Строка = StringOperations.ParseByTemplate("DC000035","DC{tmp:@[0]*}{Barcode:2}");
```

Результат:



Split

Код:

```
Строка = StringOperations.Split("Текст|для|разбиения","|");
```

Результат:

← **Результат:**

(esc) - назад

Текст; для; разбиения

ToBase64

Код:

```
Строка = StringOperations.ToBase64("Привет");
```

Результат:

← Результат:

(esc) - назад

Изначальная строка:

Привет

Конвертированная строка:

0J/RgNC40LLQtdGC

WordWrap

Код:

```
Строка = StringOperations.WordWrap("Длинный текст для переносов",3);
```

Результат:

← Результат:

(esc) - назад

Дли
нны
й
тек
ст
для
пер
ено
сов

Не нашли что искали?



Задать вопрос в техническую поддержку