

Расчетные функции для работы с товарами

- `inventQuantInCheckByOption` – Количество товаров в чеке с указанной опцией
- `inventSumInCheckByOption` – Сумма товаров в чеке с указанной опцией
- `inventQuantInCheckWithAllOptions` – Количество товаров в чеке со всеми указанными опциями
- `inventSumInCheckWithAllOptions` – Сумма товаров в чеке со всеми указанными опциями
- `inventSumInCheckByCode` – Сумма товаров в чеке с указанным кодом
- `inventSumInCheckByBcode` – Сумма товаров в чеке с указанным штрих-кодом
- `inventQuantInCheckByCode` – Количество товаров в чеке с указанным кодом
- `inventQuantInCheckByBcode` – Количество товаров в чеке с указанным штрих-кодом
- `inventQuantInCheckByCatalog` – Количество товаров в чеке с указанным каталогом
- `inventSumInCheckByCatalog` – Сумма товаров в чеке с указанным каталогом
- `inventQuantInCheckWithAllCatalogs` – Количество товаров в чеке со всеми указанными каталогами
- `inventSumInCheckWithAllCatalogs` – Сумма товаров в чеке со всеми указанными каталогами
- `inventQuantInCheck` – Количество товаров в чеке
- `inventQuantInCheckByGroup` – Количество товаров с указанными группами
- `inventSumInCheckByGroup` – Сумма товаров с указанными группами
- `getValueByOptionPrefix` – Значение из заданного словаря
- `everyNextNumber` – На каждый N-й товар в чеке
- `cheapGoodsWithOptions` – На N товаров из M с признаком
- `cheapGoods` – На N товаров из M в чеке
- `sumCheapest` – На N самых дешевых товаров с признаком
- `sumCheapestEx` – На N самых дешевых товаров из M товаров с признаком
- `expensiveGoods` – Скидка на самый дорогой товар дешевле N руб
- `expensiveGoodsFromCatalogs` – Скидка на самый дорогой товар из каталога(ов)
- `everyNextNumberOnGoods` – Скидка на каждый N-й товар не более M раз
- `limitCount` – Скидка на ограниченное количество товара в чеке
- `sumCheapestFromCatalog` – На каждый N-й товар из каталога
- `sumCheapestFromCatalogEx` – На N самых дешевых товаров из M товаров в каталоге
- `cheapGoodsFromCatalog` – На N товаров из M в каталоге
- `catalogGoods` – Назначить на товар цену из каталога
- `progressiveDiscount` – Прогрессирующая скидка
- `personalDiscount` – Персональная цена
- `countersQuantByAccumulation` – Расчет индекса скидки по счетчику

Возможности кассового ПО Artix позволяют организовывать маркетинговые акции, используя расчетные функции для работы с товарами. Товарами считаются товары в чеке, имеющие одинаковые коды и штрих-коды.

`inventQuantInCheckByOption` – Количество товаров в чеке с указанной опцией

`inventQuantInCheckByOption(options)` – функция возвращает количество товаров в чеке с указанными опциями.

Параметры функции:

- `options` – опции товара.

Пример

```
rf.inventQuantInCheckByOption([u"num"])
Расчетная ставка будет равна количеству товаров в чеке с опцией "num".
```

`inventSumInCheckByOption` – Сумма товаров в чеке с указанной опцией

`inventSumInCheckByOption(options,withPartOfKit)` – функция возвращает сумму товаров в чеке с указанной опцией.

Параметры функции:

- `options` – опции товара;
- `withPartOfKit` – параметр для учета товаров, являющихся частью комплекта.

Пример

```
rf.inventSumInCheckByOption([u"num"],False)
```

Расчетная ставка будет равна сумме цен на все товары в чеке с опцией "num". При этом не учитываются товары, которые входят в наборы.

inventQuantInCheckWithAllOptions – Количество товаров в чеке со всеми указанными опциями

`inventQuantInCheckWithAllOptions(options,withPartOfKit)` – функция возвращает количество товаров с указанными опциями.

Параметры функции:

- `options` – опции товара;
- `withPartOfKit` – параметр для учета товаров, являющихся частью комплекта. При значении параметра `withPartOfKit = True` функция при подсчете количества позиций товара с опцией учитывает позиции, которые включены в набор.

Пример

```
cf.inventQuantInCheckWithAllOptions([u"уценка",u"распродажа"],True) <= 4
```

Скидка с таким условием сработает, если количество товаров в чеке с опциями "уценка" и "распродажа" меньше либо равно 4. При подсчете количества товаров учитываются товары, входящие в комплект.

inventSumInCheckWithAllOptions – Сумма товаров в чеке со всеми указанными опциями

`inventSumInCheckWithAllOptions(options,withPartOfKit)` – функция возвращает сумму товаров в чеке со всеми указанными опциями.

Параметры функции:

- `options` – опции товара;
- `withPartOfKit` – параметр для учета товаров, являющихся частью комплекта. При значении параметра `withPartOfKit = True` функция при подсчете количества позиций товара с опцией учитывает позиции, которые включены в набор.

Пример

```
rf.inventSumInCheckWithAllOptions([u"num"],True)
```

Расчетная ставка будет равна сумме товаров в чеке с опцией "num", при этом учитываются товары, входящие в комплект.

inventSumInCheckByCode – Сумма товаров в чеке с указанным кодом

`inventSumInCheckByCode(code)` – функция возвращает сумму товаров в чеке с указанным кодом.

Параметры функции:

- `code` – код товара.

Пример

```
rf.inventSumInCheckByCode([u"102"])
```

Расчетная ставка будет равна сумме цен всех товаров в чеке с кодом 102.

inventSumInCheckByBcode – Сумма товаров в чеке с указанным штрих-кодом

`inventSumInCheckByBcode(bcode)` – функция возвращает сумму товаров в чеке с указанным штрих-кодом.

Параметры функции:

- bcode – штрих-код товара.

Пример

```
rf.inventSumInCheckByBCode([u"4607032142362"])
```

Расчетная ставка будет равна сумме цен всех товаров в чеке с штрих-кодом 4607032142362.

inventQuantInCheckByCode – Количество товаров в чеке с указанным кодом

inventQuantInCheckByCode(code) – функция возвращает количество товаров в чеке с указанным кодом равно.

Параметры функции:

- code – код товара.

Пример

```
rf.inventQuantInCheckByCode([u"102"])
```

Расчетная ставка будет равна количеству всех товаров в чеке с кодом 102.

inventQuantInCheckByBcode – Количество товаров в чеке с указанным штрих-кодом

inventQuantInCheckByBcode(bcode) – функция возвращает количество товаров в чеке с указанным штрих-кодом.

Пример функции:

- bcode – штрих-код товара.

Пример

```
rf.inventQuantInCheckByBcode([u"4607032142362"])
```

Расчетная ставка будет равна количеству всех товаров в чеке со штрих-кодом 4607032142362.

inventQuantInCheckByCatalog – Количество товаров в чеке с указанным каталогом

inventQuantInCheckByCatalog(catalog, withPartOfKit) – функция возвращает количество товаров в чеке с указанным каталогом.

Параметры функции:

- catalog – код каталога;
- withPartOfKit – параметр для учета товаров, являющихся частью комплекта (набора). При значении параметра withPartOfKit = True функция при подсчете количества позиций товара с опцией учитывает позиции, которые включены в набор.

Пример

```
rf.inventQuantInCheckByCatalog([u"num"], True)
```

Расчетная ставка будет равна количеству товаров в чеке с каталогом "num". При подсчете количества позиций учитываются позиции с товарами, входящими в набор.

inventSumInCheckByCatalog – Сумма товаров в чеке с указанным каталогом

`inventSumInCheckByCatalog(catalog, withPartOfKit)` – функция возвращает сумму товаров в чеке с указанным каталогом.

Параметры функции:

- `catalog` – код каталога;
- `withPartOfKit` – параметр для учета товаров, являющихся частью комплекта (набора). При значении параметра `withPartOfKit = True` функция при подсчете количества позиций товара с опцией учитывает позиции, которые включены в набор.

Пример

```
rf.inventSumInCheckByCatalog(["num"], True)
```

Расчетная ставка будет равна сумме товаров в чеке с каталогом "num". При подсчете количества позиций учитываются позиции с товарами, входящими в набор.

inventQuantInCheckWithAllCatalogs – Количество товаров в чеке со всеми указанными каталогами

`inventQuantInCheckWithAllCatalogs(catalog, withPartOfKit)` – функция возвращает количество товаров в чеке со всеми указанными каталогами.

Параметры функции:

- `catalog` – код каталога;
- `withPartOfKit` – параметр для учета товаров, являющихся частью комплекта (набора). При значении параметра `withPartOfKit = True` функция при подсчете количества позиций товара с опцией учитывает позиции, которые включены в набор.

Пример

```
rf.inventQuantInCheckWithAllCatalogs(["num1", "num2"], True)
```

Расчетная ставка будет равна количеству товаров в чеке, входящих одновременно в каталоги "num1" и "num2". При подсчете количества позиций учитываются позиции с товарами, входящими в набор.

inventSumInCheckWithAllCatalogs – Сумма товаров в чеке со всеми указанными каталогами

`inventSumInCheckWithAllCatalogs(catalog, withPartOfKit)` – функция возвращает сумму товаров в чеке со всеми указанными каталогами.

Параметры функции:

- `catalog` – код каталога;
- `withPartOfKit` – параметр для учета товаров, являющихся частью комплекта (набора). При значении параметра `withPartOfKit = True` функция при подсчете количества позиций товара с опцией учитывает позиции, которые включены в набор.

Пример

```
rf.inventSumInCheckWithAllCatalogs(["11"], True)
```

Расчетная ставка будет равна сумме товаров в чеке с каталогом 11, при этом учитываются товары, являющиеся частью комплекта.

inventQuantInCheck – Количество товаров в чеке

`inventQuantInCheck(uniteFractional)` – функция возвращает количество товаров в чеке.

Параметры функции:

- `uniteFractional` – правила подсчета позиций весового товара.
 - При значении `uniteFractional = True` функция считает за единицу один весовой товар, даже если он был добавлен в чек несколькими позициями (объединяет при подсчете несколько позиций с одним весовым товаром в одну).
 - При значении `uniteFractional = False` функция считает за единицу каждый добавленный в чек весовой товар.

Пример

```
rf.inventQuantInCheck()
```

Расчетная ставка будет равна количеству товаров в чеке.

inventQuantInCheckByGroup – Количество товаров с указанными группами

`inventQuantInCheckByGroup(group, withPartOfKit)` – функция возвращает количество товаров с указанной группой в чеке.

Параметры функции:

- `group` – код группы;
- `withPartOfKit` – параметр для учета товаров, являющихся частью комплекта (набора).

Пример

```
rf.inventQuantInCheckByGroup([u"1"],False)
```

Расчетная ставка будет равна количеству товаров, относящихся к группе 1, при этом товары, являющиеся частью комплекта, не учитываются.

inventSumInCheckByGroup – Сумма товаров с указанными группами

`inventSumInCheckByGroup(groups, withPartOfKit)` – функция возвращает сумму товаров с указанной группой в чеке.

Параметры функции:

- `groups` – список кодов групп;
- `withPartOfKit` – параметр для учета товаров, являющихся частью комплекта (набора).

Пример

```
rf.inventSumInCheckByGroup([u"1",u"2"],True)
```

Расчетная ставка будет равна сумме товаров, входящих в группы "1" и "2", при этом учитываются товары, входящие в наборы.

getValueByOptionPrefix – Значение из заданного словаря

`getValueByOptionPrefix(prefix, values, defaultValue)` – функция для расчетной ставки чтобы можно было получить значение из заданного словаря (передается в параметре функции) по префиксу ключа.

Параметры функции:

- `prefix` – префикс опции товара;
- `values` – словарь пар: "<опция товара>" : <числовое значение>;
- `defaultValue` – значение по умолчанию, которое возвращается функцией при отсутствии элемента в списке значений.



Функция может быть использована для работы с акциями, объектами которых являются позиции.

Функция разбирает чек на позиции, выбирает товары с опцией и задает значение расчетной ставки, определенное заданным словарем пар. Если в товаре указано несколько опций, подходящих под условия расчетной ставки, то будет выбрано первое значение. В том случае, когда параметрами функции не определена пара опция-значение ключа, то размер скидки определяется значением по умолчанию.

Пример

Чтобы дать скидку на товары, имеющие префикс MC, и определить размер скидки через словарь пар, необходимо: сделать расчетную ставку `rf.getValueByOptionPrefix(u'MC', {u'MC01': 10, u'MC02': 20, u'MC03': 30}, 5)`

Укажем результатом расчетной ставки проценты.

Скидка будет предоставлена следующим образом:

- на товар с опцией MC01 в размере 10%,
- на товар с опцией MC02 в размере 20%,
- на товар с опцией MC03 в размере 30%,
- на остальные товары, имеющие префикс MC, в размере 5%.

everyNextNumber – На каждый N-й товар в чеке

`everyNextNumber(number, percent, desc)` – скидка сработает на каждый N-й товар в чеке.

Для функции используется только суммовая расчетная ставка.

Параметры функции:

- `number` – номер позиции;
- `desc` – сортировка товаров по цене.
 - При значении параметра `desc = True` скидка предоставляется на каждый N-й товар по убыванию цены.
 - При значении `desc = False` скидка предоставляется на каждый N-й товар по возрастанию цены.
- `percent` – процент скидки на позиции.

Возвращает словарь из номеров позиций и абсолютных сумм скидок соответственно.

Пример

Для того, чтобы дать скидку на каждую 2-ю позицию по убыванию цены на 5% необходимо в условия скидки добавить:

```
object["posnum"] in cf.everyNextNumber(2,True)
```

Скидку сделать расчетной по сумме и добавить функцию:

```
rf.everyNextNumber(2, 5, True)[object["posnum"]]
```

cheapGoodsWithOptions – На N товаров из M с признаком

`cheapGoodsWithOptions(count, extops, rate, scout, toCheapest)` – скидка предоставляется на N товаров из M с указанным признаком.

Для функции используется только суммовая расчетная ставка.

Параметры функции:

- `count` – M товаров с признаком;
- `extops` – признак, указывающий на то, что товар участвует в акции, задаваемой данной функцией. Является обязательным параметром;
- `rate` – размер скидки в процентах;
- `scount` – N количество товаров из M, к которым применяется скидка;
- `toCheapest` – позиции, к которым будет применена скидка.
 - При значении параметра `toCheapest = True` скидка применяется только к самым дешевым товарам.
 - При значении `toCheapest = False` скидка распределяется между всеми товарами с опциями.

Функция разворачивает чек на позиции, выбирает все позиции с опциями, сортирует эти позиции по цене, выбирает `scount` из `count` товаров и рассчитывает скидку на товары. Скидка применяется ко всем товарам, участвующим в акции, или только к самым дешевым в соответствии с параметрами функции.

Пример

Для того, чтобы дать скидку в размере 50% на 2 товара из 3-х с признаком "num" необходимо:

```
создать условие object["posnum"] in cf.cheapGoodsWithOptions(3, [u"num"], 50, 2, False)
```

```
сделать расчетную скидку по сумме rf.cheapGoodsWithOptions(3, [u"num"], 50, 2, False)[object["posnum"]]
```

cheapGoods – На N товаров из M в чеке

cheapGoods(count, rate, scout, toCheapest) – скидка предоставляется на N товаров из M в чеке.

Скидка предоставляется только на позицию. Для функции используется только суммовая расчетная ставка.

Параметры функции:

- count – M товаров;
- rate – величина скидки в процентах;
- scout – N количество товаров из M, к которым применяется скидка;
- toCheapest – позиции, к которым будет применена скидка.
 - При значении параметра toCheapest = True скидка применяется только к самым дешевым товарам.
 - При значении toCheapest = False скидка распределяется между всеми товарами в чеке.

Функция разворачивает чек на позиции, сортирует эти позиции по цене, выбирает scout из count товаров и рассчитывает скидку на товары. Скидка применяется ко всем товарам, участвующим в акции, или только к самым дешевым в соответствии с параметрами функции.

Пример

Для того чтобы дать скидку в размере 50% на 2 товара из 3-х в чеке, необходимо:
создать условие `object["posnum"] in cf.cheapGoods(3, 50, 2, False)`
сделать расчетную скидку по сумме `rf.cheapGoods(3, 50, 2, False)[object["posnum"]]`

Пример поведения функции

Чек состоит из позиций:

1. Картофель 7 кг*50 рублей = 350 рублей
2. Молоко 4 шт*25 рублей = **100** рублей
3. Гречка 10 кг*20 рублей = **200** рублей

Сумма чека = 650 рублей.

Скидка будет применена к следующим товарам:

1. Молоко
2. Гречка

Сумма скидки на чек **100***50% + **200***50% = 150 рублей.

Так как toCheapest = False, то скидка распределяется между всеми товарами в чеке.



При задании акции необходимо учитывать, что воздействие функции применяется к позициям с товарами, входящими в набор.

sumCheapest – На N самых дешевых товаров с признаком

sumCheapest(count, extops, rate) – функция для расчетной ставки, дающая скидку на N самых дешевых товаров с признаком.

Параметры функции:

- count – коэффициент для расчета количества товаров, к которым будет применяться скидка;
- extops – признак, указывающий на то, что товар участвует в акции, задаваемой данной функцией. Является обязательным параметром;
- rate – размер скидки в процентах.

Для корректного применения скидки результатом расчетной ставки должна быть скидка по сумме. Сумма скидки определяется на основании стоимости самых дешевых товаров в признаке. Количество самых дешевых товаров, на которые будет предоставлена скидка, определяется по правилу: Количество товаров в чеке с признаком, деленное без остатка, на значение, заданное в параметре count.

Пример расчета количества товаров для предоставления скидки

Пусть в чеке имеется 7 товаров с признаком. В параметре count задано значение 3. Количество товаров, на которые будет предоставлена скидка, $7/3 = 2,333$. Количество товаров со скидкой равно 2.

Пример применения функции

Зададим скидку на чек с расчетной ставкой `rf.sumCheapest(2,[u"num"],100)`

Пример чека:

товар 1 (с признаком "num") цена 100

товар 2 (с признаком "num") цена 90

товар 3 (без признака) цена 80

товар 4 (с признаком "num") цена 70

товар 5 (с признаком "num") цена 60

В чеке имеется 4 товара с признаком "num". Количество самых дешевых товаров, к которым будет применена скидка 100%, равна 2. Скидка 100% сработает на товар 4 и товар 5. Сумма скидки будет распределена между всеми позициями чека.



Если в значении `extops` используются символы кириллицы, то при задании функции перед текстом необходимо ставить символ `'u'`.

Например, `rf.sumCheapest(2, [u'Акция "Подарок к празднику"]',50)`.

sumCheapestEx – На N самых дешевых товаров из M товаров с признаком

`sumCheapestEx(count, extops, rate, scout)` – функция для расчетной суммовой ставки, дающая скидку на N самых дешевых товаров из M товаров с признаком.

Параметры функции:

- `count` – коэффициент для расчета количества товаров, к которым будет применяться скидка;
- `extops` – признак, указывающий на то, что товар участвует в акции, задаваемой данной функцией. Является обязательным параметром;
- `rate` – размер скидки в процентах;
- `scount` – количество применений скидки.

Для корректного применения скидки результатом расчетной ставки должна быть скидка по сумме. Сумма скидки определяется на основании стоимости самых дешевых товаров в признаке. Количество самых дешевых товаров, на которые будет предоставлена скидка, определяется по правилу: Количество товаров в чеке с признаком, деленное без остатка, на значение, заданное в параметре `count`, и умноженное на количество применений `scount`.

Пример

Зададим скидку на чек с расчетной ставкой `rf.sumCheapestEx(5,[u"num"],50,3)`

Пример чека:

товар 1 (с признаком "num") цена 40

товар 2 (с признаком "num") цена 30

товар 3 (с признаком "num") цена 20

товар 4 (с признаком "num") цена 40

товар 5 (с признаком "num") цена 30

В чеке имеется 5 товаров с опцией "num", скидка 50% будет применена к 3 товарам из 5 самых дешевых. Количество товаров, к которым применяется скидка определяется как 5 (количество товаров в чеке), деленное на 5 (коэффициент для расчета) и умноженное на 5 (количество применений скидки). Скидка 50% сработает на товары 2,3 и 5, как на самые дешевые. Сумма скидки будет распределена между всеми позициями чека.



Если в значении `extops` используются символы кириллицы, то при задании функции перед текстом необходимо ставить символ `'u'`.

Например, `rf.sumCheapestEx(2, [u'Акция "Подарок к празднику"]',50,2)`.

expensiveGoods – Скидка на самый дорогой товар дешевле N руб

`expensiveGoods(countGoods, conditionSum, resultSum, ignorePosWithPriceEqualMin)` – функция для расчетной суммовой ставки, которая устанавливает цену со скидкой на N товаров дешевле или равной пороговой суммы.

Параметры функции:

- `countGoods` – количество товаров в чеке, на которые дается скидка;
- `conditionSum` – пороговая цена для товара. Цена товара за 1шт/кг должна быть меньше пороговой или равна ей, тогда к товару будет применена скидка;
- `resultSum` – цена товара, которую будет иметь товар после применения скидки;
- `ignorePosWithPriceEqualMin` – игнорировать позицию с ценой равной минимальной цене.

- При значении параметра `ignorePosWithPriceEqualMin = True` функция игнорирует товары, для которых цена равна минимальной цене (`true` значение по умолчанию).
- При значении `ignorePosWithPriceEqualMin = False` функция будет учитывать товары, для которых цена равна минимальной цене. Чтобы на такой товар была дана скидка, необходимо в утилите администрирования Yuki включить опцию "[Не учитывать минимальную цену](#)".

Функция разбирает чек по товарам и сортирует их по цене. Выбирается `countGoods` товаров из самых дорогих товаров, имеющих цену ниже или равную `conditionSum`. Для выбранных товаров задается цена `resultSum`.

Пример

Чтобы дать скидку на 2 самых дорогих товара в чеке дешевле 35 рублей и установить цену после скидки в 1 рубль необходимо: сделать расчетную ставку по сумме `rf.expensiveGoods(2, 35, 1, True)[object["posnum"]]`



Расчетная функция применяется совместно с функцией в условиях скидок. О поведении функции "Скидка на самый дорогой товар дешевле N руб" можно подробнее прочитать в разделе "[Функции для работы с товарами](#)".

expensiveGoodsFromCatalogs – Скидка на самый дорогой товар из каталога(ов)

`expensiveGoodsFromCatalogs(countGoods, conditionSum, resultSum, ignorePosWithPriceEqualMin)` – функция для расчетной суммовой ставки, которая устанавливает цену со скидкой на N товаров из каталога(ов).

Параметры функции:

- `countGoods` – количество товаров в чеке, на которые дается скидка;
- `conditionCatalogs` – список каталогов для которых действует скидка. (Пример: `[u"2"]` или `[u"1",u"2",u"4"]`);
- `resultSum` – цена товара, которую будет иметь товар после применения скидки;
- `ignorePosWithPriceEqualMin` – игнорировать позицию с ценой равной минимальной цене.
 - При значении параметра `ignorePosWithPriceEqualMin = True` функция игнорирует товары, для которых цена равна минимальной цене (`true` значение по умолчанию).
 - При значении `ignorePosWithPriceEqualMin = False` функция будет учитывать товары, для которых цена равна минимальной цене. Чтобы на такой товар была дана скидка, необходимо в утилите администрирования Yuki включить опцию "[Не учитывать минимальную цену](#)".

Функция разбирает чек по товарам и сортирует их по цене. Выбирается `countGoods` товаров из самых дорогих товаров, входящих в каталог(и) `conditionCatalogs`. Для выбранных товаров задается цена `resultSum`.

Пример

Чтобы дать скидку на 2 самых дорогих товара из каталога 2 и установить цену после скидки в 1 рубль необходимо: сделать расчетную ставку по сумме `rf.expensiveGoodsFromCatalogs(2, [u"2"], 1, True)[object["posnum"]]`



Расчетная функция применяется совместно с функцией в условиях скидок. О поведении функции "Скидка на самый дорогой товар из каталога(ов)" можно подробнее прочитать в разделе "[Функции для работы с товарами](#)".

everyNextNumberOnGoods – Скидка на каждый N-й товар не более M раз

`everyNextNumberOnGoods(number, countLimit, percent, extops)` – функция для расчетной суммовой ставки, которая дает скидку на каждый N-й товар, имеющий дополнительные опции, не более M раз.

Параметры расчета скидки задаются в параметрах функции и могут быть дополнительно указаны в свойствах товара.

Параметры функции:

- `number` – кратность товара, указывает на какой по счету товар будет применена скидка;
- `countLimit` – количество вхождений, показывающее на сколько кратных повторений товара будет предоставлена скидка;
- `percent` – размер скидки, выраженный в процентах;
- `extops` – список дополнительных опций товаров, в параметрах функции `everyNextNumberOnGoods` указывается только значение `deepdiscount`.

Дополнительные опции товара:

- `deepdiscount` – признак, указывающий на то, что товар участвует в акции, задаваемой данной функцией, является обязательным параметром.



Название признака может быть любым, функция подбирает товар с таким признаком, который задан в функции. Например, для предоставления скидки на товар можно указать признак `deepdiscount`, а для начисления бонусов признак `deepbonus`.

- `ddmultiplicity=ЗНАЧЕНИЕ` – кратность товара, аналогична параметру `number`
- `ddmaxcount=ЗНАЧЕНИЕ` – количество вхождений, аналогично параметру `countLimit`
- `ddpercent=ЗНАЧЕНИЕ` – размер скидки, аналогичен параметру `percent`

Параметры, указанные в товаре, имеют приоритет перед параметрами функции. Если в свойствах товара указан только признак товара, участвующего в акции, а другие опции отсутствуют, то при расчете скидки будут применены параметры функции. Если в свойствах товара задана только часть опций для расчета скидки, то недостающие значения будут получены из параметров функции.

Функция разбирает чек на позиции, выбирает товары с признаком и группирует их по штрих-коду. Скидка в X процентов дается на каждый N-й товар M раз. В чеке позиция со скидкой поднимается вверх списка идентичных товаров.

Пример

Чтобы дать скидку 50% на каждый 2-й товар с признаком `deepdiscount`, но ограничить предоставление скидки 4 вхождениями, необходимо:

создать условие `object["posnum"] in cf.everyNextNumberOnGoods(2, 4, 50, ["deepdiscount"])`

сделать расчетную ставку по сумме `rf.everyNextNumberOnGoods(2, 4, 50, ["deepdiscount"])(object["posnum"])`

Скидка в этом случае сработает не более 4 раз (`countLimit <= 4`), т.е. будет применена к каждому 2, 4, 6 и 8 товару с признаком `deepdiscount`, ко всем последующим скидка не применяется. Следует учесть, что описанным образом скидка будет применена для товаров с одинаковым штрих-кодом. Если в чеке будет присутствовать 3 товара с признаком `deepdiscount`, но разными штрих-кодами, то скидка не будет применена ни разу.

Если в свойствах товара помимо признака `deepdiscount` будут заданы дополнительные опции `ddmultiplicity=3, ddmaxcount=2, ddpercent=30`, то в этом случае скидка в размере 30% (`ddpercent=30`) будет применена к каждому 3 товару (`ddmultiplicity=3`) не более 2 раз (`ddmaxcount=2`). Параметры функции будут игнорироваться при расчете.



Расчетная функция применяется совместно с функцией в условиях скидок. О поведении функции "Скидка на каждый N-й товар не более M раз" можно подробнее прочитать в разделе "[Функции для работы с товарами](#)".

limitCount – Скидка на ограниченное количество товара в чеке

`limitCount(maxCount, index)` – функция предоставит скидку равную разнице основной и индексной цены на ограниченное количество товара в чеке.

Для функции используется только суммовая расчетная ставка.

Параметры функции:

- `maxCount` – количество товара в чеке, на которое будет предоставлена скидка. Может принимать целые и дробные значения, применяемые к штучному и весовому товару. При дробном значении параметра применяется к тому количеству товара, которое указано в параметре;
- `index` – индекс дополнительной цены.

Функция разворачивает чек на позиции, имеющие дополнительную цену с заданным кодом, выбирает указанное количество товара и присваивает им скидку равную разнице основной и индексной цены.



Без задания дополнительных ограничивающих условий скидка будет предоставлена на все товары в чеке, имеющие дополнительную цену с заданным индексом.

Функция может быть использована только для условий в скидках на позицию.

Пример

Чтобы предоставить N количества товара по скидке равной разнице основной и индексной цены, необходимо:

создать условие `object['posnum'] in cf.limitCount(4,2)`

создать расчетную ставку по сумме `rf.limitCount(4,2)(object['posnum'])`

Скидка в этом случае сработает на все товары, количеством до 4 единиц (шт или кг), и присвоит им скидку равную разнице основной цены и цены с индексом 2.

Функция не применяется для товаров, входящих в набор, и товаров, не имеющих дополнительной цены. Для весовых товаров скидка применяется на все товары до заданного количества.



Расчетная функция применяется совместно с функцией в условиях скидок. О поведении функции "Скидка на ограниченное количество товара в чеке" можно подробнее прочитать в разделе ["Функции для работы с товарами"](#).

sumCheapestFromCatalog – На каждый N-й товар из каталога

sumCheapestFromCatalog(count, extops, rate) – функция для расчетной ставки, дающая скидку на каждый N-й товар из каталога.

Параметры функции:

- count – коэффициент для расчета количества товаров, к которым будет применяться скидка;
- extops – идентификаторы каталогов, товары из которых участвуют в акции;
- rate – размер скидки в процентах.

Для корректного применения скидки результатом расчетной ставки должна быть скидка по сумме. Сумма скидки определяется на основании стоимости самых дешевых товаров из каталога. Количество самых дешевых товаров из каталога, на которые будет предоставлена скидка, определяется по правилу: Количество товаров в чеке из каталога, деленное без остатка, на значение, заданное в параметре count.

Пример расчета количества товаров для предоставления скидки

Пусть в чеке имеется 7 товаров из каталога 1. В параметре count задано значение 3. Количество товаров, на которые будет предоставлена скидка, $7/3 = 2,333$. Количество товаров со скидкой равно 2.

Пример применения функции

Зададим скидку на чек с расчетной ставкой `rf.sumCheapestFromCatalog(2,[1,2],50)`

Пример чека:

товар 1 (из каталога 1) цена 100

товар 2 (из каталога 2) цена 90

товар 3 (без каталога) цена 80

товар 4 (из каталога 1) цена 70

товар 5 (из каталога 2) цена 60

В чеке имеется 4 товара из каталогов 1 и 2. Количество самых дешевых товаров, к которым будет применена скидка 50%, равна 2. Скидка 50% сработает на товар 4 и товар 5. Сумма скидки будет распределена между всеми позициями чека.



При предоставлении скидки на весовые товары функция определяет 1 кг весового товара, как 1 единицу штучного товара.

sumCheapestFromCatalogEx – На N самых дешевых товаров из M товаров в каталоге

sumCheapestEx(count, extops, rate, countDisc) – функция для суммовой расчетной ставки, дающая скидку на N самых дешевых товаров из M товаров в каталоге.

Параметры функции:

- count – коэффициент для расчета количества товаров, к которым будет применяться скидка;
- extops – идентификаторы каталогов, товары из которых участвуют в акции;
- rate – размер скидки в процентах;
- countDisc – количество применений скидки.

Для корректного применения скидки результатом расчетной ставки должна быть скидка по сумме. Сумма скидки определяется на основании стоимости самых дешевых товаров в каталоге. Количество самых дешевых товаров, на которые будет предоставлена скидка, определяется по правилу: Количество товаров в чеке из каталога, деленное без остатка, на значение, заданное в параметре count, и умноженное на количество применений countDisc.

Пример

Зададим скидку на чек с расчетной ставкой `rf.sumCheapestFromCatalogEx(5,[1,2],50,3)`

Пример чека:

товар 1 (из каталога 1) цена 40

товар 2 (из каталога 1) цена 30

товар 3 (из каталога 2) цена 20

товар 4 (из каталога 1) цена 40

товар 5 (из каталога 2) цена 30

В чеке имеется 5 товаров из каталогов 1 и 2, скидка 50% будет применена к 3 товарам из 5 самых дешевых. Количество товаров, к которым применяется скидка определяется как 5 (количество товаров в чеке), деленное на 5 (коэффициент для расчета) и умноженное на 3 (количество применений). Скидка 50% сработает на товары 2, 3 и 5, как на самые дешевые. Сумма скидки будет распределена между всеми позициями чека.



При предоставлении скидки на весовые товары функция определяет 1 кг весового товара, как 1 единицу штучного товара.

cheapGoodsFromCatalog – На N товаров из M в каталоге

`cheapGoodsFromCatalog(count, extops, rate, scout, toCheapest)` – скидка сработает на N товаров из M в каталоге.

Скидка предоставляется только на позицию. Для функции используется только суммовая расчетная ставка.

Параметры функции:

- `count` – M товаров из каталога;
- `extops` – идентификаторы каталогов, товары из которых участвуют в акции;
- `rate` – размер скидки в процентах;
- `scount` – N количество товаров из M, к которым применяется скидка;
- `toCheapest` – позиции, к которым будет применена скидка.
 - При значении параметра `toCheapest = True` скидка применяется только к самым дешевым товарам.
 - При значении `toCheapest = False` скидка распределяется между всеми товарами из каталога.

Функция разворачивает чек на позиции, выбирает все позиции из каталога, сортирует эти позиции по цене, выбирает `scount` из `count` товаров и рассчитывает скидку на товары. Скидка применяется ко всем товарам, участвующим в акции, или только к самым дешевым в соответствии с параметрами функции.

Пример

Для того, чтобы дать скидку в размере 50% на 2 товара из 3-х, которые содержатся хотя бы в одном из каталогов с кодом 1 или 3 необходимо:
создать условие `object["posnum"] in cf.cheapGoodsFromCatalog(3, [1,3], 50, 2, False)`
сделать расчетную скидку по сумме `rf.cheapGoodsFromCatalog(3, [1,3], 50, 2, False)(object["posnum"])`



При предоставлении скидки на весовые товары функция определяет 1 позицию весового товара, как 1 единицу штучного товара.



Расчетная функция применяется совместно с функцией в условиях скидок. О поведении функции "На N товаров из M в каталоге" можно подробнее прочитать в разделе "[Функции для работы с товарами](#)".

catalogGoods – Назначить на товар цену из каталога

Скидка предоставляется только на позицию. Для функции используется только суммовая расчетная ставка.

Параметры функции:

- `catalogGoods(catalogid)` – на товар назначается цена из каталога;
- `catalogid` – код (идентификатор) каталога.

Функция находит в чеке товары, входящие в каталог, и предоставляет для них скидку (надбавку), равную разнице между начальной ценой и ценой товара в каталоге. В результате применения функции цена товара становится равной цене в каталоге. Если цена товара в каталоге не задана, то функция не срабатывает.

Пример

Для того, чтобы на товар, входящий в каталог 1, назначить цену, указанную в этом каталоге, необходимо:
создать условие `object["posnum"] in cf.catalogGoods("1")`
сделать расчетную ставку по сумме `rf.catalogGoods("1")[object["posnum"]]`



Расчетная функция применяется совместно с функцией в условиях скидок. О поведении функции "Назначить на товар цену из каталога" можно подробнее прочитать в разделе ["Функции для работы с товарами"](#).

progressiveDiscount – Прогрессирующая скидка

`progressiveDiscount(minDiscount, maxDiscount, discountStep, catalogs)` – на товары из каталога будет предоставлена прогрессирующая скидка (начислены баллы).

Скидка предоставляется только на позицию. Для функции используется только суммовая расчетная ставка.

Параметры функции:

- `minDiscount` – минимальное значение скидки (в процентах);
- `maxDiscount` – максимальное значение скидки (в процентах);
- `discountStep` – шаг, с которым рассчитывается размер скидки на каждый следующий товар;
- `catalogs` – список каталогов, на товары из которых может быть предоставлена скидка.

Функция находит в чеке товары из каталогов и предоставляет на них скидку с заданным шагом. При этом максимальная скидка будет предоставлена на товар с наименьшей стоимостью. Количество товаров со скидкой зависит от разницы пороговых значений скидки и заданного шага.

Пример

Для того чтобы на 3 товара из каталога с кодом 6 была предоставлена скидка 10, 20 и 30% необходимо:
создать условие `object["posnum"] in cf.progressiveDiscount(10, 30, 10, [u"6"])`
сделать расчетную ставку по сумме `rf.progressiveDiscount(10, 30, 10, [u"6"])[object["posnum"]]`



Расчетная функция применяется совместно с функцией в условиях скидок. О поведении функции "Назначить на товар цену из каталога" можно подробнее прочитать в разделе ["Функции для работы с товарами"](#).

personalDiscount – Персональная цена

Параметры функции:

- `personalDiscount()` – на товары в чеке будет предоставлена персональная цена.

Скидка предоставляется только на позицию. Для функции используется только суммовая расчетная ставка.


Функция срабатывает при наличии в чеке карты покупателя при переходе в подытог или при добавлении карты в подытог. Взаимодействие с другими скидками срабатывает по заданному правилу взаимодействия.

Пример

Для того чтобы на товары в чеке была предоставлена персональная цена необходимо:
создать условие `object["posnum"] in cf.personalDiscount()`
сделать расчетную ставку по сумме `rf.personalDiscount()[object["posnum"]]`



Расчетная функция применяется совместно с функцией в условиях скидок. О поведении функции "Назначить на товар цену из каталога" можно подробнее прочитать в разделе ["Функции для работы с товарами"](#).

 Значения цен для каждого товара будут получены с сервиса персональных цен.

countersQuantByAccumulation – Расчет индекса скидки по счетчику

countersQuantByAccumulation(idCounter, ranges, defaultIndex) – функция проверяет значения счетчика и возвращает процент скидки в соответствии с вхождением счетчика в определенный диапазон значений.

Параметры функции:

- idCounter – идентификатор счетчика, указывается в одинарных кавычках, например '111'. Ведение счетчиков выполняется сервисом счетчиков, правила применения счетчика задаются в системе лояльности Artix Loyalty Management;
- ranges – список диапазонов значений счетчика. Для карты может быть указано несколько диапазонов значения счетчика, каждому из которых будет присвоен определенный процент скидки. Например, [(0,4),(5,None)], где None – неограниченный верхний интервал.
- defaultIndex – индекс скидки по умолчанию. Скидка с указанным индексом будет использована в том случае, когда отсутствует связь с сервером счетчиков:
 - 0 – будет использована скидка, указанная первой в расчетной ставке;
 - -1 – скидка не предоставляется при отсутствии связи;
 - номер индекса – соответствующий индекс в списке % скидок.

В том случае, если счетчик акции в чеке не найден, то значение счетчика принимается равным 0 и для него подбирается соответствующий диапазон.

Пример

Для того чтобы разграничить процент скидки, предоставляемой по дисконтной карте необходимо:
сделать условие `cf.countersQuantByAccumulation('111',[(0,4),(5,None)],0) >= 0`
сделать процентную расчетную ставку `[1,10,50][rf.countersQuantByAccumulation('111', [(0,4),(5,None)],0)]`,
где `[1,10,50]` - список % скидок.

При отсутствии связи с сервисом счетчиков:

Если индекс скидки указан 0, то будет использована скидка 1%.


Если индекс скидки указан 1, то будет использована скидка 10% и т.д.


При наличии связи с сервером процент скидки определяется по значению счетчика.

Если счетчик попадает в диапазон (0,4), то по карте будет предоставлена скидка 10%.

Если счетчик попадает в диапазон (5,None), то по карте будет предоставлена скидка 50%.

Если счетчик не обнаружен, то его значение устанавливается в 0 и применяется соответствующий диапазон.

 Расчетная функция применяется совместно с функцией в условиях скидок. О поведении функции "Назначить на товар цену из каталога" можно подробнее прочитать в разделе ["Функции для работы с товарами"](#).

 Функция работает только совместно с сервисом счетчиков.