В модуль формы документа ЗаказКлиента в процедуру при СозданииНаСервере

// {Экзамен

экзаменНастроитьФорму();

Если ЗначениеЗаполнено(Объект.Ссылка) Тогда

экзаменЗаполнитьМинимальныеЦены();

КонецЕсли;

// }Экзамен

В процедуру ПослеЗаписиНаСервере, ДатаПриИзмененииЗавершение, ПриИзмененииСоглашенияСервер, ОбработкаВыбораПодборНаСервере добавляем вызов

// {Экзамен

экзаменЗаполнитьМинимальныеЦены();

// }Экзамен

ТоварыНоменклатураПриИзменении, ТоварыХарактеристикаПриИзменении, ТоварыУпаковкаПриИзменении

// {Экзамен

экзаменЗаполнитьМинимальныеЦены(ТекущаяСтрока.НомерСтроки);

// }Экзамен

В конец модуля формы добавляем

#Область Экзамен

&НаСервере

Процедура экзаменНастроитьФорму()

Массив = Новый Массив;

НовыйРеквизит = Новый РеквизитФормы("экзаменПорогРентабельности", ОбщегоНазначения.ОписаниеТипаЧисло(15,2),"Объект.Товары","Цена (min)");

Массив.Добавить(НовыйРеквизит);

ИзменитьРеквизиты(Массив);

НовыйЭлемент = Элементы.Вставить("экзаменПорогРентабельности", Тип("ПолеФормы"), Элементы.Товары, Элементы.ТоварыЦена);

НовыйЭлемент.ПутьКДанным = "Объект.Товары.экзаменПорогРентабельности";

НовыйЭлемент.Вид = ВидПоляФормы.ПолеВвода;

НовыйЭлемент.ТолькоПросмотр = Истина;

НовыйЭлемент.Ширина = 8;

КонецПроцедуры // экзаменНастроитьФорму()

&НаСервере

Процедура экзаменЗаполнитьМинимальныеЦены(НомерСтроки = Неопределено)

ДатаСреза = Объект.Дата;

Валюта = Объект.Валюта;

Если НомерСтроки <> Неопределено Тогда

Массив = Новый Массив;

Массив.Добавить(Объект.Товары.Получить(НомерСтроки-1));

Товары = Объект.Товары.Выгрузить(Массив, "НомерСтроки, Номенклатура, Характеристика, Упаковка");

Иначе

Товары = Объект.Товары.Выгрузить(, "НомерСтроки, Номенклатура, Характеристика, Упаковка");

КонецЕсли;

Запрос = Новый Запрос;

Запрос.УстановитьПараметр("ДатаСреза", ДатаСреза);

Запрос.УстановитьПараметр("ВтТовары", Товары);

Запрос.УстановитьПараметр("Валюта", Валюта);

Запрос.Текст =

"ВЫБРАТЬ

| ВтТовары.НомерСтроки,

| ВтТовары.Номенклатура,

| ВтТовары.Характеристика,

| ВтТовары.Упаковка

|ПОМЕСТИТЬ Товары

|ИЗ

| &ВтТовары КАК ВтТовары

|;

|

|////////////////////////////////////////////////////////////////////////////////

|ВЫБРАТЬ

| ЕСТЬNULL(КурсыВалютСрезПоследнихВалютаПорога.Курс, 1) \* ЕСТЬNULL(КурсыВалютСрезПоследнихВалютаДокумента.Кратность, 1) / ЕСТЬNULL(КурсыВалютСрезПоследнихВалютаДокумента.Курс, 1) / ЕСТЬNULL(КурсыВалютСрезПоследнихВалютаПорога.Кратность, 1) КАК КроссКурс

|ПОМЕСТИТЬ КурсыВалют

|ИЗ

| РегистрСведений.КурсыВалют.СрезПоследних(&ДатаСреза, Валюта = &Валюта) КАК КурсыВалютСрезПоследнихВалютаДокумента

| ПОЛНОЕ СОЕДИНЕНИЕ РегистрСведений.КурсыВалют.СрезПоследних(

| &ДатаСреза,

| Валюта В

| (ВЫБРАТЬ

| ВидМинимальноДопустимыхЦенПродажи.Значение

| ИЗ

| Константа.ВидМинимальноДопустимыхЦенПродажи КАК ВидМинимальноДопустимыхЦенПродажи)) КАК КурсыВалютСрезПоследнихВалютаПорога

| ПО (ИСТИНА)

|;

|

|////////////////////////////////////////////////////////////////////////////////

|ВЫБРАТЬ

| Товары.НомерСтроки,

| ВЫБОР

| КОГДА Товары.Упаковка <> ЗНАЧЕНИЕ(Справочник.УпаковкиЕдиницыИзмерения.ПустаяСсылка)

| ТОГДА &ТекстЗапросаКоэффициентУпаковки1

| ИНАЧЕ 1

| КОНЕЦ \* ЕСТЬNULL(ЦеныНоменклатурыСрезПоследних.Цена, 0) / ЕСТЬNULL(&ТекстЗапросаКоэффициентУпаковки2, 1) \* КурсыВалют.КроссКурс КАК Цена

|ИЗ

| Товары КАК Товары

| ЛЕВОЕ СОЕДИНЕНИЕ РегистрСведений.ЦеныНоменклатуры.СрезПоследних(

| &ДатаСреза,

| (Номенклатура, Характеристика) В

| (ВЫБРАТЬ

| Товары.Номенклатура,

| Товары.Характеристика

| ИЗ

| Товары КАК Товары)

| И ВидЦены В

| (ВЫБРАТЬ

| ВидМинимальноДопустимыхЦенПродажи.Значение

| ИЗ

| Константа.ВидМинимальноДопустимыхЦенПродажи КАК ВидМинимальноДопустимыхЦенПродажи)) КАК ЦеныНоменклатурыСрезПоследних

| ПО Товары.Номенклатура = ЦеныНоменклатурыСрезПоследних.Номенклатура

| И Товары.Характеристика = ЦеныНоменклатурыСрезПоследних.Характеристика

| ЛЕВОЕ СОЕДИНЕНИЕ КурсыВалют КАК КурсыВалют

| ПО (ИСТИНА)";

Если НЕ ПолучитьФункциональнуюОпцию("ИспользоватьУпаковкиНоменклатуры") Тогда

ТекстЗапроса = Запрос.Текст;

ЗаменяемыйТекст =

" ВЫБОР

| КОГДА Товары.Упаковка <> ЗНАЧЕНИЕ(Справочник.УпаковкиЕдиницыИзмерения.ПустаяСсылка)

| ТОГДА &ТекстЗапросаКоэффициентУпаковки1

| ИНАЧЕ 1

| КОНЕЦ \* ЕСТЬNULL(ЦеныНоменклатурыСрезПоследних.Цена, 0) / ЕСТЬNULL(&ТекстЗапросаКоэффициентУпаковки2, 1) \* КурсыВалют.КроссКурс КАК Цена";

Если СтрНайти(ТекстЗапроса, ЗаменяемыйТекст) = 0 Тогда

ВызватьИсключение НСтр("ru = 'Некорректный текст запроса'");

КонецЕсли;

ТекстЗамены =

" ЕСТЬNULL(ЦеныНоменклатурыСрезПоследних.Цена, 0) \* ЕСТЬNULL(КурсыВалют.КроссКурс, 1) КАК Цена";

ТекстЗапроса = СтрЗаменить(ТекстЗапроса, ЗаменяемыйТекст, ТекстЗамены);

КонецЕсли;

ТекстЗапроса = Запрос.Текст;

ТекстЗапроса = СтрЗаменить(ТекстЗапроса, "&ТекстЗапросаКоэффициентУпаковки1",

Справочники.УпаковкиЕдиницыИзмерения.ТекстЗапросаКоэффициентаУпаковки(

"Товары.Упаковка",

"Товары.Номенклатура"));

ТекстЗапроса = СтрЗаменить(ТекстЗапроса, "&ТекстЗапросаКоэффициентУпаковки2",

Справочники.УпаковкиЕдиницыИзмерения.ТекстЗапросаКоэффициентаУпаковки(

"ЦеныНоменклатурыСрезПоследних.Упаковка",

"ЦеныНоменклатурыСрезПоследних.Номенклатура"));

Запрос.Текст = ТекстЗапроса;

РезультатЗапроса = Запрос.Выполнить();

Выборка = РезультатЗапроса.Выбрать();

Пока Выборка.Следующий() Цикл

Если Выборка.Цена = 0 Тогда

Продолжить;

КонецЕсли;

НайденныеСтроки = Объект.Товары.НайтиСтроки(Новый Структура("НомерСтроки", Выборка.НомерСтроки));

Если НайденныеСтроки.Количество() Тогда

НайденныеСтроки[0].экзаменПорогРентабельности = Выборка.Цена;

КонецЕсли;

КонецЦикла;

КонецПроцедуры

&НаСервере

Процедура экзаменПроверитьОтклоненияОтУсловийПродаж(Отказ)

Если Пользователи.РолиДоступны("ПолныеПрава, ОтклонениеОтУсловийПродаж", Пользователи.ТекущийПользователь()) Тогда

Возврат;

КонецЕсли;

Для каждого СтрокаТоваров Из Объект.Товары Цикл

Если СтрокаТоваров.экзаменПорогРентабельности > СтрокаТоваров.Цена Тогда

ОбщегоНазначенияКлиентСервер.СообщитьПользователю("Цена продажи меньше минимальной",,"Объект.Товары["+Формат(СтрокаТоваров.ИсходныйНомерСтроки-1, "ЧГ=")+"].Цена",,Отказ);

КонецЕсли;

КонецЦикла;

КонецПроцедуры // экзаменПроверитьОтклоненияОтУсловийПродаж()

#КонецОбласти

В процедуру ОбработкаПроверкиЗаполненияНаСервере

// {Экзамен

экзаменПроверитьОтклоненияОтУсловийПродаж(Отказ);

// }Экзамен