# Теория

### Прайс-чекер

Простыми словами, прайс-чекер (Price Checker) — это устройство с предустановленным программным обеспечением, позволяющее покупателю узнать цену и дополнительную информацию о товаре путём сканирования штрих-кода. Под дополнительной информацией подразумевается набор метрик, а именно: код товара, наименование, количество на складе, артикул, краткое описание и т.д. В нашем случае — это отдельный бизнес-процесс, который будет показывать нам любую, необходимую в рамках наших задач, информацию.

## Для чего это нужно

Пример: все мы ходим в магазин за покупками, но не всегда можем найти к ним актуальный ценник. На помощь приходят мобильные устройства, позволяющие сканировать штрих-код и узнать поподробнее о товаре: цену, остаток, характеристику, акции. В качестве аналога — использование интерактивного Web-киоска, который точно так же берет на себя функционал информирования клиентов/покупателей. Но у них есть одно отличие, а именно наличие обширной памяти и дополнительного оборудования, что значительно расширяет возможности самого магазина. Принцип действия такого устройства прост — покупатель точно таким же способом, что и в случае прайс-чекера, подносит штрих-код интересующего товара к сканеру и получает всю подробную информацию. Но вдобавок он может получить предложение по подобным товарам той же категории, марки и фирмы. При желании даже с картинками.

Пример 2: сотрудник магазина осуществляет контроль цен путем проведения проверки штрих-кодов. Это исключает неактуальную информацию о товаре в торговом зале, снижает риски неверного расположения ценников и т.д. Выполнение подобных задач значительно упростит наличие у проверяющего ТСД с заложенным в нем процессом проверки цен.



# Практика

#### Задача

Нам необходимо создать бизнес-процесс, в результате которого мы будем получать информацию о товаре следующего образца:

- код товара
- наименование товара
- артикул
- остаток
- цена

Дополнительно реализуем функционал двух кнопок.

## Создание бизнес-процесса

Переходим в «Мастер конфигураций», открываем вкладку «Бизнес-процессы» и нажимаем кнопку «Новый бизнес-процесс» и начинаем его создавать:

- 1. В открывшемся окне вводим имя нового Бизнес-процесса (даем ему название «Прайс-чекер») и нажимаем «Сохранить».
- 2. Создаем в области компоновки визуальных элементов для экрана ТСД (то есть, что в итоге будет выводиться на экран) следующий шаблон:

Machinal yound automation with the second of the secon	
Kog toseps {product.code } * * * * * * * * * * * *	×
{ product.code }	
	×
Hawkenobashie ToBapa { product.description } * * * * * * * * * * * *	×
{ product.description }	
Aptrusyn       { characteristic.product.article }       & Koresleng       & Koresleng       & Toser meetra       Ocratox       { product.rest }       & Toser meetra       Liene       { product.price.rur }	
	×
Aprusys { characteristic.product.article } & Romeniesy There resers Coraross { product.rest } & Romeniesy Konneliesy K	×
{ characteristic.product.article } & Korreling & Korreling & There reserve Corarook { product.rest } & There reserve	
Romeniang	////
Coratox Coratox {product.rest}  Tuens {product.price.rur}  Konneloge Konnel	
Coratox {product.rest} Product.price.rur} Romeshatter,12 X @ Rear texts X Rearest hatter,12 X	×
{ product.rest }	
Konnelong      K	
Lena { product.price.rur } @ Konnology @ Konnology & K	×
{ product.price.rur }	
Konnelsep     Konnels butten, 12 X	
Konneirez     Konneir butten, 12 X     Kinne merze X     Konneir butten, 12 X	
Konneinep     Konnex butten, 12 X     Konnex butten, 12 X	
Konner: button, 12 X     Konner: button, 13 X	
Keonea: button_12 X     Keonea: button_13 X	
ВЫХОД КАМЕРА	
КАМЕРА	

Для построения такой конструкции используем 2 отдельных друг от друга контейнера, 5 полей текста, 2 кнопки и разделители.

Легкий лайфхак: чтобы сделать на экране ТСД расстояние между кнопками более приемлемым для глаз, поместите между кнопками пустое поле текста. В настройках этого поля текста оставляем пустыми **Подпись** и **Основной текст**, **Динамические данные** не указываем.



Разница до/после добавления поля текста между кнопками:



В качестве домашнего задания предлагается создать новый бизнес-процесс и визуализировать его, используя различные комбинации кнопок, контейнеров и полей в конструкторе.

3. Настраиваем форму БП следующим образом:

Настроить форму	×
Название формы Главная форма	
Описание	
Data Таблица штрихкодов	•
Тип формы Запись таблицы	•
СОХРАНИТЬ СБРОСИТЬ ЗАКРЕ	ить

4. Немного украсим наш бизнес-процесс, присвоив цвета кнопкам, тексту и подложке (фону):





5. После изменения цветовой гаммы переходим на холст и начинаем составлять логику бизнес-процесса. Блок-схема процесса выглядит так:



Сперва составляем логику работы запроса данных:

Форма: Главная фор	ома	$\ \ \square$	<u>+</u>
{Load } {Resume }	{ Click }	{ Sca	in }
<ul> <li>Запрос даннь</li> </ul>	іх через R	est A	×
Тип запрашиваемых д	анных		
Классификаторы			]?
{ Failure }	{ Suc	cess }	
		/	

В типах данных указываем «Классификаторы». На этом этапе происходит запрос в ERP-систему.

Просьба обратить внимание: когда мы планируем присоединить одну инструкцию к другой, возможные соединения система подсветит нам зеленым цветом.

6. Настраиваем подтверждение наших действий:

	∧ Запрос дан	ных через Rest A $ imes$
	Тип запрашиваемь	іх данных
	Классификаторы	
	{ Failure }	{ Success }
^	Подтверждение	действия 🗙
Текст В 1( о ш	r сообщения С есть подготовлен трихкодах	ные данные 🕜
Текст	гкнопки	
Заг	рузить	?
	{ Failure }	{ Success }

На этом этапе мы создаем отображение всплывающих окон на ТСД, которые запрашивают подтверждение наших дальнейших действий, и текст кнопки подтверждения.

7. Далее – оформляем запись в базу и обновление данных после записи:

∧ Подтвержµ	цение действия	×
Текст сообщения		
В 1С есть подгот о штрихкодах	говленные данные	0
Текст кнопки		
Загрузить		] ⑦
{ Failure }	{Success}	
∧ Запись дан	ных в базу	×
{ Failure }	{ Success }	
	∧ Обновление,	данных ×
	{ Failure }	{Success}

Эта инструкция обрабатывает и записывает входящие данные в базу.

8. Формируем логику работы кнопки «Выход»

	Кнопка: button	.12	
	{ Cl	ick }	
^	Закрытие фор	мы	×
	{ Failure }	{ Success }	

9. Определяем поведение формы при сканировании – возвращаемся к первой инструкции на макете «Главная форма», добавляем новую инструкцию «Поиск объекта в базе данных» и соединяем параметр «Scan» формы с этим поиском объекта в базе:

	Форма: Г	лавная фор	ма	Ē ±	
$\vdash$	{ Load }	{ Resume }	{ Click }	{ Scan }	

<ul> <li>Поиск объекта</li> </ul>	а в базе данных 🗙
Таблица для поиска об	ъекта
Штрихкоды	× • ?
Имя реквизита для пои	ска
Значение штрихкод	a × • ?
{ Failure }	{ Success }

10. Правильно настраиваем поиск в базе. Для этого создаем две новые инструкции и соединяем Поиск с сообщением об ошибке и обновлением данных, как на скриншоте:

1	∧ Поиск объекта в б	азе данных 🗙	
Та	аблица для поиска объект	a	
	Штрихкоды	× • ?	
и :	мя реквизита для поиска Значение штрихкода	× • ?	
	{ Failure }	{ Success }	
	2		
<ul> <li>Сообщение об</li> </ul>	ошибке 🛛 🗙	∧ Обновление да	анных ×
Текст сообщения Штрихкод \$ не найд	ен	{ Failure }	{ Success }
{ Failure }	{ Success }		

Сама кнопка сканирования настраивается так:



И соединяется с инструкцией «Поиск объекта в базе данных» из предыдущего пункта. Таким образом, при нажатии кнопки «Камера» ТСД запустит встроенную камеру.

11. Если наш терминал не подключен к товаро-учетной системе, итогом сканирования тестового штрих-кода будет сообщение:

15:01		Θ 🕯 🗋
	Код товара	:
-	аименование т	овара:
	Man	
	Артикул:	
	and the second	
	All and the	
	Остаток:	
	Цена:	
	0	
Toppor		10048
найден	штрихкоду %090	KAMEPA
A CONTRACTOR		