# Пишем стих.

Ну кто может. Кто нет – перепечатывает.

### Немного теории.

Мобильная аппликация состоит из 3-х взаимосвязанных «слоев»:

- внешнего вида (пользовательского интерфейса). Его мы набираем из визуальных компонентов в конструкторе, его видит пользователь при работе и с ним взаимодействует.
- Внутренняя база данных. Структура, отдельная и изолированная для каждого бизнес-процесса и состоящая из набора таблиц, необходимых для работы. Например, таблицы «Товары» и «Штрих-коды» нужны для прайс-чекера, Для инвентаризации могут потребоваться еще «Таблица документов» и «Таблица строк в документах»
- Инструкции (цепочки инструкций) они определяют поведение каждой формы, ее реакции на нажатие кнопок, сканирование, они получают, отдают и изменяют данные.

В любом бизнес-процессе, сконструированном в системе, первичны именно данные. Форма отображает их, запускает инструкции. Инструкции модифицируют данные. Форма показывает результат. Показали данные -> Изменили данные -> Показали измененные данные.

В этой части, мы будем выводить на экран Хокку поэта Эномото Кикаку:

Мошек легкий рой Вверх летит - плавучий мост Для моей мечты.

Помните, что, самовольно изменяя текст, вы действуете на свой страх и риск. Разработчики не несут ответственности за последствия.

## Экранные формы

Для выполнения, будем использовать 2 формы 2-х разных типов: на первой мы будем вносить, редактировать или удалять текст строку за строкой, на второй – любоваться результатом. Первая форма будет иметь тип «Объект» - то есть она оперирует одним конкретным объектом данных, одной записью. Вторая – тип «Список». Выводит список объектов, соответствующих критерию.

Так как мы для работы не получаем никаких данных из товароучетной системы и ничего не отправляем, берем первую подходящую структуру, из доступных в конструкторе. Пусть это

будет «Таблица товаров», каждой строке нашего Хокку будет соответствовать название «Товара», которое мы создадим в процессе.

Создаем новый бизнес-процесс, называем его «Хокку» и заходим в него.

Объектная форма «Строки»

Создаем первую форму и устанавливаем ее параметры:

	-				
	D	Контейнер			
	••••	Поле ввода			
ń.	Т	Поле текста			
с процессов	***	Кнопка		Настроить форму	×
ураций	Ŧ	Разделитель		пастроять форму	
				Название формы Строки	
un sident)			-		
wbi(dev)			<	Операции со строками	
вание			1	/	
			1	Таблица товаров	•
				С Тип формы	
		/		Запись таблицы	-
				СОХРАНИТЬ СБРОСИТЬ ЗА	крыть
				🗸 ФОРМЫ В БИЗНЕС ПРОЦЕСС	e 🗸
	н	аздание формы:			
		Строки Добавит новую фор	ь млу		
	-	- +			
	Ň				

Добавляем нужные нам элементы на форму:

	×
Ю Поле текста	
- Затоловок 	
	×
<b>Ха</b> Поле ввода	>
âaroneek	×
<b>18</b> Контейнер	>
В Кнопка: button_9 X     Кнопка: button_10 X     Кнопка: button_11 X       СОХРАНИТЬ     УДАЛИТЬ     НОВАЯ	
	×
Kikonika: button_12	

**Поле текста** – в нем мы будем отображать текст после того, как наберем его и сохраним в базе данных.

Текс	тово	ре по	ле							×
Подпись Стока			-							
×					A	Α	А	E	Ŧ	∃
Данные 	по умолч	нанию		Динамич Наимен	еские да НОВАНИ	нные 1е това	ра			•
								0	OXPAH	ИТБ

Настраиваем.

Устанавливаем текст и расположение подписи, указываем, из какого поля базы данных мы хотим видеть информацию.

Поле ввода – в него мы будем набирать каждую строку текста.



Отключаем отображение подписи и также указываем связь с полем базы данных.

Блок кнопок управления:

Контейнер			×
🔯 Кнопка: button_9 🗙	🔯 Кнопка: button_10 🗙	🐯 Кнопка: button_11 🗙	
ок		+	
Кнопка: button_12			

**Внимание!** Подписи кнопок **button\_X** — это системные имена, они могут отличаться от тех, которые будут в вашем конструкторе бизнес-процесса!

Состоит из вложенных один в другой контейнеров с разными режимами размещения содержимого. Общий для этого блока контейнер распределяет свое содержимое (кнопку и второй контейнер) в столбец, вложенный контейнер с 3-мя кнопками - в строку.

Кнопка «Ок» будет сохранять данные с экрана в БД, «-» - удалять из БД и с экрана строку, «+» - добавлять новую пустую запись в БД и отображать ее на экране для редактирования.

### Списочная форма «Хокку»

170.7%				
areno		Добавить новую форму		×
	for secar	Название формы		
¢		Описание формы		
	🔯 Контеймер			•
	ф Контейно ф Кнотса	Тип формы Список записей		•
	КНС	добавить форму	СБРОСИТЬ	ЗАКРЫТЬ
	B Keonka: button_12		×	
		кнопка		
		•	ФОРМЫ В БИЗНЕС ПР	оцессе 🗸
Добавить новую форму				
+				

Добавляем форму и устанавливаем ее параметры:

Эта форма также будет оперировать данными из Таблицы товаров, но отображать эти данные форма будет в виде списка записей.

Все, что нам потребуется на этой форме, это – текстовое поле, которое будет показывать «Наименование товара». Строка за строкой. Без заголовка, чтобы ничто не отвлекало.

<b>10</b> Поле текста	МАКСИМАЛЬНАЯ ШИРИНА ТЕРМИНАЛ	1A, 480PX	×	
{ description }				
	-			
	Текстовое поле			>
	Подпись Заголовок			
			Δ ≡	-
			• =	
	Данные по умолчанию Д	инамические данные Наименование товара		
			<b>a</b>	охранить

#### Логика поведения.

#### Форма «Строки»

При открытии формы нам надо продолжать творить, если в прошлом подходе муза внезапно нас покинула. Поэтому, загружая форму, выводим первую запись из базы данных на экран:

Форма: Строки	Ē ±
{Load} {Resume	<pre>{Click} {Scan}</pre>
Лоиск объект.	а в базе данных 🗙
Таблица для поиска об	іъекта
Товары	× • ?
Имя реквизита для по	ncka
{ Failure }	{Success}
	2
<ul> <li>Обновление д</li> </ul>	анных ×
{ Failure }	{ Success }

Если на вход инструкции «Поиск объектов в базе данных» мы не передаем значение, которое надо найти – инструкция нам покажет первую запись из таблицы.

#### Немного про кнопки.

Каждая помещенная на форму кнопка, имеет свой уникальный идентификатор на этой форме:

*	
<b>163</b> Контейнер	×
Киопка: button_9 X Киопка: button_10	KHORKA: buttor_11 ×
ок -	
Кнопка: button_12	×
хокку	

Точно такие же идентификаторы имеют инструкции на холсте действий. С этих инструкций всегда начинаются цепочки действий, которые программа выполняет в ответ на нажатие кнопки:

Форма: Строки 📋 🛓	Кнопка button_11	Кнопка button_9	Кнопка button_12	
{Load} {Resume} {Click} {Scan}	( Click )	{ Click }	{ Click }	
	∧ Создание объекта ×	🗠 Запись данных в базу 🛛 🗙		
∧ Поиск объекта в базе данных ×	Таблица базы данных	(Failure) (Success)	Переход к форме	
Габлица для поиска объекта	Товары Х • 🧿	(conset)	Имя формы для перехода	
Товары 🗙 💌 🔿	{Failure} {Success}		Хокку Х -	
		<u> </u>	Вариант открытия	
Наименование Х т		<ul> <li>Обновление данных</li> <li>×</li> </ul>	Отображение всех да 🗙 👻 🕐	

Для того, чтобы начинать творить, нам нужна пустая запись в базе данных, в которую мы запишем нетленку. Это должно произойти при нажатии на кнопку «+». Рисуем, не забываем обновить связь между данными и формой!

Кног	ika: button_11		
	{ Click }		
	5		
^	Создание объе	кта	$\times$
Габл	ица базы данных		
Toe	ары	X	• ?
	{ Failure }	{ Succes	is }
~	Обновление да	инных	×
	{ Failure }	{ Succes	is }

После окончания ввода, строку надо сохранить в базе данных. После выполнения этой инструкции, введенная строка будет сохранена и отобразится в текстовом поле в верхней части формы. Рисуем:

Кнопка: button_9 (Click)	
<ul> <li>Запись данных в базу</li> <li>(Failure)</li> <li>(Success)</li> </ul>	Мошек легкий рой
∧ Обновление данных ×	Мошек легкий рой
(Failure) (Success)	
	хокко

Данные (текст в нашем случае) можно редактировать сколько угодно раз, не забывая нажимать ОК для сохранения. Мы работаем с конкретной записью, и пока мы с нее не ушли кнопкой «удалить» или «создать» - эта запись будет у нас перед глазами.

Если то, что мы видим перед собой на экране, не удовлетворяет нас, безжалостно удаляем запись.

Кнопка: button_10		
{ Click }		
5		Crowa
<ul> <li>Подтверждение дейс</li> </ul>	твия ×	Мошек легкий рой
Текст сообщения		
Точно удаляем?	?	
Текст кнопки		and the second s
Без сомнений	?	
{ Failure } {	Success }	мошек легкии рои
<ul> <li>Удаление объекта</li> </ul>	×	
{ Failure } {	Success }	Точно удаляем?
<ul> <li>Обновление данных</li> </ul>	x ×	Без сомнений!
{ Failure }	{ Success }	Отмена

Для того, чтобы прочитать получившийся шедевр, переходим к форме «Хокку», отображая на ней все доступные данные, без дополнительной фильтрации:

{ Click }	Для моей мечты.
Лереход к форме	
Имя формы для перехода Хокку Х т 🕐	
Вариант открытия Отображение всех да Х  (Failure) (Success)	

## Форма «Хокку»

Так как форма предназначена только для отображения результата наших усилий, нам потребуется единственная возможность – редактировать ошибки ввода. Для этого, при клике на любую строку, мы открываем предыдущую форму и загружаем эту строку в нее:

орма: Хокку		⊡ ≛	
Load } {Resume }	{ Click }	{Scan}	
л Перехо	од к форме	•	×
Имя формы д	ля перехода		
Строки		× •	?
Вариант откры	ытия		
Отображен	ие с прим	× •	?
{ Failure	• }	{ Success	}