



**Формы оплаты для универсального модуля
ввода данных.**

Программное обеспечение

«Процессинговый центр Pay-logic»

Руководство пользователя

АННОТАЦИЯ

Описывает процесс создания и конфигурирования сервисов с использованием форм оплаты универсального модуля ввода данных программного обеспечения «Процессинговый Центр Pay-logic»

Версия руководства: 5.9

Руководство актуально для кабинета агента версий 5.8.x, ТПО версий 5.1xx, 7.1xx, РМА версии 6.6x, мобильных приложений приема платежей версий 3.xx

2008–2025 ООО «Софт-Лоджик», г. Барнаул, Россия

Данный документ входит в комплект поставки программных продуктов.

Права использования данного документа предусмотрены соответствующим лицензионным договором.

ООО «Софт-Лоджик»

656006, г. Барнаул, Малахова ул., дом 146в

Тел: (3852) 72-27-27

© Soft-logic

Web: <https://soft-logic.ru/>

Mail: info@soft-logic.ru

ОГЛАВЛЕНИЕ

ИСТОРИЯ ИЗМЕНЕНИЙ.....	6
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 5.4.....	6
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 5.5.....	6
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 5.6.....	7
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 5.7.....	7
ИЗМЕНЕНИЕ В ВЕРСИИ ДОКУМЕНТА 5.8.....	7
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 5.9.....	8
1 ОСНОВНЫЕ ТЕРМИНЫ.....	9
2 ОСНОВНЫЕ СОКРАЩЕНИЯ.....	12
3 ПРЕДПОЛАГАЕМАЯ АУДИТОРИЯ.....	13
4 ОБЩИЕ СВЕДЕНИЯ О МОДУЛЕ ВВОДА ДАННЫХ.....	14
5 ЭЛЕМЕНТЫ ФОРМЫ ДЛЯ УНИВЕРСАЛЬНОГО МОДУЛЯ ВВОДА ДАННЫХ.....	18
5.1 ОБЩАЯ ИНФОРМАЦИЯ.....	18
5.2 ФАЙЛ ФОРМ (<FORMS>).....	20
5.2.1 НАЗНАЧЕНИЕ, СТРУКТУРА КОРНЕВОГО ТЕГА ФАЙЛА ФОРМ.....	20
5.2.2 ФОРМА (<FORM>).....	21
5.2.3 ПОЛУЧЕНИЕ ДАННЫХ ИЗ СПРАВОЧНИКА ТОЧКИ (<PREFILL-ITEMS>).....	29
5.2.4 ОБЪЕДИНЕНИЕ ПОЛЕЙ В СТРОКУ (<ROW>).....	38
5.3 КЛАВИАТУРЫ.....	40
5.3.1 КЛАВИАТУРЫ В 7 ВЕРСИИ ТПО.....	40
5.4 ПОЛЕ ВВОДА.....	43
5.4.1 ОБЩИЕ ПОЛОЖЕНИЯ.....	43
5.4.2 ФЛАГИ.....	48
5.4.3 МЕХАНИЗМЫ ЛОКАЛИЗАЦИИ ФОРМ.....	52
5.5 ПРОСТЫЕ ТИПЫ ДАННЫХ ЗНАЧЕНИЙ АТТРИБУТОВ ЭЛЕМЕНТОВ, ИСПОЛЬЗУЕМЫХ НА ЭКРАНАХ ФОРМ.....	55

6 ОСНОВНЫЕ ТЕГИ	57
6.1 ТЕКСТОВОЕ ПОЛЕ ВВОДА (<TEXT-FIELD>)	57
6.1.1 НАЗНАЧЕНИЕ И СТРУКТУРА ТЕКСТОВОГО ПОЛЯ ВВОДА (<TEXT-FIELD>).....	57
6.1.2 ВАЛИДАЦИЯ ДАННЫХ (<VERIFY>).....	67
6.1.3 ФОРМАТИРОВАНИЕ ДАННЫХ (<SPLIT>).....	73
6.1.4 ЗАМЕНА ЗНАЧЕНИЯ (<REPLACES>).....	76
6.1.5 ФИЛЬТРАЦИЯ ДАННЫХ (<FILTER>).....	79
6.2 ПОЛЕ ВВОДА ЧИСЛОВОГО ЗНАЧЕНИЯ (<NUMERIC-FIELD>)	83
6.3 ФЛАГ/ПЕРЕКЛЮЧАТЕЛЬ (<CHECKBOX-FIELD>)	94
6.4 СЕЛЕКТОРЫ	99
6.4.1 ОБЩИЕ ПОЛОЖЕНИЯ.....	99
6.4.2 ТЕКСТОВЫЙ СЕЛЕКТОР (<TEXT-SELECTOR>).....	100
6.4.3 ГРАФИЧЕСКИЙ СЕЛЕКТОР (<IMAGE-SELECTOR>).....	109
6.5 ВЫБОР ДАТЫ (<DATE-FIELD>)	114
6.6 ПРИСВОЕНИЕ ЗНАЧЕНИЯ (<SET-FIELD>)	119
6.7 ИНФОРМАЦИОННЫЙ ЭКРАН (<INFO-FIELD>)	122
6.8 АВТОЗАПОЛНЕНИЕ (<AUTOCOMPLETE-FIELD>)	128
6.9 ГРУППОВОЙ ЭКРАН (<GROUP-FIELD>)	137
6.10 ЭКРАН ПОДТВЕРЖДЕНИЯ С КОМИССИЕЙ (<COMMISSION-CONFIRM>)	143
7 ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ	145
7.1 ПОДСКАЗКА (<HELP>).....	145
7.2 СЧИТЫВАНИЕ ДАННЫХ СО ШТРИХ-КОДОВ (<BARCODE-SCANNER>).....	148
7.2.1 НАЗНАЧЕНИЕ, СТРУКТУРА <BARCODE-SCANNER>.....	148
7.2.2 ПРАВИЛА РАСПОЗНАВАНИЯ ШТРИХ-КОДА (<BARCODE-PARSER>).....	150
7.3 ВКЛЮЧЕНИЕ ДАННЫХ ИЗ ВНЕШНЕГО ФАЙЛА (<XI:INCLUDE>).....	155
7.4 НАСТРОЙКА ГОЛОСОВОГО СОПРОВОЖДЕНИЯ.....	157
7.5 ИСПОЛЬЗОВАНИЕ ПЕРЕМЕННОЙ, ОБОЗНАЧАЮЩЕЙ КОРНЕВОЙ КАТАЛОГ, ПРИ УКАЗАНИИ ПУТЕЙ.....	159
8 ЦЕПОЧКИ ЭКРАНОВ И ВЕТВЛЕНИЯ	161
9 ТИПОВЫЕ ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ ЯЗЫКА ОПИСАНИЯ ФОРМ	162
10 ОТЛИЧИЯ ТПО 5 И 7 ВЕРСИИ	185
11 ОСОБЕННОСТИ СИНТАКСИСА ДЛЯ ПРИЛОЖЕНИЯ «ПРИЕМ ПЛАТЕЖЕЙ ANDROID»	186

ПРИЛОЖЕНИЕ А. ПРИМЕР ФОРМЫ С ВЛОЖЕННЫМ СЕЛЕКТОРОМ.....187

ИСТОРИЯ ИЗМЕНЕНИЙ

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 5.4

Дата публикации: 30.09.2019.

Изменение	Раздел
Улучшения в версии 11.1.0 универсального обработчика:	
Добавлен атрибут force-external-range , который позволяет задать ограничение на значение, вводимое в числовое поле <code><numeric-field></code> . Ограничение устанавливается в атрибутах min , max	6.2

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 5.5

Дата публикации: 03.12.2019.

Изменение	Раздел
Общие улучшения в документе:	
Исправлен пример использования элемента <code><set-field></code>	6.6

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 5.6

Дата публикации: 26.05.2023.

Изменение	Раздел
Улучшения в версии 5.136:	
Добавлено маскирование номера карты в логах ТПО	5.4.2

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 5.7

Дата публикации: 04.09.2024.

Изменение	Раздел
Улучшения в документации:	
Добавлена возможность настраивать перенаправление пользователя при использовании шаблонов платежей.	5.2.2

ИЗМЕНЕНИЕ В ВЕРСИИ ДОКУМЕНТА 5.8

Дата публикации: 31.03.2025.

Изменение	Раздел
Улучшения в документации:	
Добавлено описание параметра trailing-widget в структуре текстового поля ввода.	6.1.1

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 5.9

Дата публикации: 05.11.2025.

Изменение	Раздел
Улучшения в документации:	
Добавлено описание флагов	5.4.2

1 ОСНОВНЫЕ ТЕРМИНЫ

Агент — юридическое лицо, владеющее точками приема платежей или действующее как дистрибьютор — не имея собственных точек приема платежей и выполняющее функции по приему платежей или предоставлению возможности другим лицам проводить платежи через себя.

Атрибуты платежа — информация, по которой можно идентифицировать операцию платежа в платёжной системе.

Карта сдачи — представляет собой виртуальные денежные средства, которые сохраняются на сервере, при внесении пользователем денежных средств в сумме большей, чем требуется для проведения платежа. Один из возможных вариантов реализации механизма сдачи.

Личный кабинет агента — специализированный веб-сайт, который предназначен для управления сетью точек приема платежей и просмотра финансовой статистики.

Номерная емкость — файл с указанием привязки номера к оператору связи региона.

Объект процессинга — модель данных процессинга содержит в себе описание структур объектов и их отношений. Каждый объект обладает набором определенных свойств и для него могут быть указаны значения этих свойств.

Переменная — поименованная область памяти, адрес которой можно использовать для осуществления доступа к данным. Данные являются значением этой переменной.

Платёж — расчёт за услугу, осуществляемый абонентом (клиентом) при помощи различных платежных инструментов: наличных денежных средств, банковской карты, карт сдачи, ваучеров.

Платежная система — юридическое лицо, самостоятельно предоставляющее возможность оплачивать свои услуги, либо юридическое лицо, предоставляющее возможность оплачивать услуги других компаний.

Тип точки — вид программно-аппаратного устройства, с помощью которого осуществляется прием платежа (терминал, РМА, мобильный клиент и т.д.)

Провайдер платёжных сервисов (от англ. payment service provider) — компания, которая предоставляет онлайн-сервисы по осуществлению электронных платежей различными способами, включая смарт-карты, банковские платежи, такие как банковские операции и другие. Некоторые провайдеры платёжных сервисов предоставляют различные инновационные сервисы: платёжные системы, включая платежи наличными, электронные кошельки, предоплаченные карты, ваучеры и т. д.

Проведение платежа — обработка данных о платеже сервером платежной системы.

Произвольное свойство объекта — определяется объекту дополнительно независимо от заданной структуры и взаимосвязей в модели данных. Недоступно в фильтрах, выводится в отчетах по SQL запросу к БД процессинга.

Сервис — услуга, по которой принимаются платежи в платёжной системе.

Сервис провайдера — сервис в процессинге, жестко привязанный к провайдеру по идентификационному коду в системе провайдера.

Собственный провайдер — провайдер, заведенный на уровне текущего агента, имеющий договор с этим агентом.

Субагент — дочерний агент, находящийся в иерархии агентской сети ниже по отношению к текущему агенту.

Сценарий оплаты по сервису — xml-документ, который описывает процесс оплаты услуги и определяет последовательность полей ввода данных, экранов и правил

переходов между ними, допускает как ручной ввод данных, так и получение данных из вне (запросы к шлюзам и оборудованию), при всем этом сценарии поддерживают операции над введенными данными, логические условия, циклы, сложные валидаторы и другие возможности.

Точка — программно-аппаратное устройство, с помощью которого (через которую) осуществляется прием платежа.

Форма оплаты по сервису — xml-документ, который описывает процесс оплаты услуги и определяет последовательность заполняемых пользователем полей ввода данных без возможности выполнять действия над этими данными (атрибутами платежа).

2 ОСНОВНЫЕ СОКРАЩЕНИЯ

ПО — программное обеспечение.

ПС — платежная система.

РМА — рабочее место агента.

ТПП — точка приема платежей.

3 ПРЕДПОЛАГАЕМАЯ АУДИТОРИЯ

Руководство предназначено администраторам и пользователям ПО «Процессинговый центр Pay-logic», имеющим опыт работы с ПС, знакомым с принципами работы в ОС Linux и Windows. Предполагается, что специалист, создающий формы оплаты по сервису, обладает общими знаниями в области языков разметки.

Документ содержит описание процесса создания и конфигурирования сервисов с использованием форм оплаты универсального модуля ввода данных, а также описание справочной и системной информации ПО «Процессинговый Центр Pay-logic».

Язык форм, описываемый в руководстве, является разработкой компании Soft-logic и в ПО других производителей не поддерживается.

Руководство является полным по созданию и конфигурированию сервисов с использованием форм оплаты универсального модуля ввода данных.

Процесс создания сервисов состоит из 5 шагов:

1. Создание нового элемента в справочнике сервисов.
2. Связывание сервиса с одним или несколькими провайдерами.
3. Описание процесса оплаты в файле .xml для универсального модуля ввода данных.
4. Создание логотипа.
5. Обновление терминалов и РМА.

Процедуры создания сервисов описаны также в документе [«Сценарии оплаты для усовершенствованного модуля ввода данных. Программное обеспечение «Процессинговый центр Pay-logic». Руководство пользователя»](#).

4 ОБЩИЕ СВЕДЕНИЯ О МОДУЛЕ ВВОДА ДАННЫХ

Модули ввода данных по сервису (или тип обработчика данных по сервису) — это модули, реализованные во всех платёжных инструментах, описывающие логику оплаты и последовательность смены экранов переходов на терминале при оплате услуги, служат для реализации формы или сценария оплаты.

Модуль ввода данных задает последовательность смены экранов на терминале при оплате по сервису. На каждом экране выводятся поля ввода текстовой, цифровой информации, выбора даты, поля с автозаполнением либо поля с predetermined значениями, из которых пользователь может выбрать необходимое. Навигация по экранам на терминале осуществляется кнопками: «Назад», «Далее», «Выход».

В ПО «Процессинговый центр Pay-logic» используются три основных модуля ввода данных:

1. Мобильный.
2. Универсальный.
3. Усовершенствованный.

Существуют и другие модули ввода данных, разработанные под отдельные функциональные особенности работы сервисов, в данном руководстве не рассматриваются.

Мобильный модуль ввода данных служит для сервисов мобильной связи, которые работают на основе номерных ёмкостей операторов. Ёмкости операторов содержатся в файле *capacity.xml.gz* (более подробно описано в документе [«Провайдеры. Программное обеспечение «Процессинговый центр Pay-logic». Руководство пользователя»](#)). Для генерации данного файла разработан отдельный шлюз. Рекомендуется обновлять данный файл не реже, чем раз в месяц.

Для усовершенствованного модуля ввода данных создается сценарий оплаты по сервису. Особенности создания сценариев описаны в документе [«Сценарии оплаты для усовершенствованного модуля ввода данных. Программное обеспечение «Процессинговый центр Pay-logic». Руководство пользователя»](#).

Для универсального модуля ввода данных создается форма оплаты по сервису.

Формы оплаты по сервисам представляют собой файлы xml. В настоящее время формы оплаты по сервисам могут располагаться в общем файле *forms.xml*, в составном файле *Nxx.xml* (где N — число полученное от целочисленного деления номера формы на 100) и отдельный для конкретного сервиса (в наименовании файла формы указывается код сервиса, например, *192.xml*).

Разбивка файлов форм на составные файлы *Nxx.xml* используется для увеличения производительности при поиске и для снижения объема трафика при обновлении.

В текущих версиях ТПО файл *forms.xml*, как правило, не используется.

Кроме того, предусмотрена возможность создания нескольких копий файлов оплаты форм в зависимости от языка пользователя, то есть поддерживается локализация. Более подробная информация о локализации приведена в разделе [5.4.3](#).

При выборе сервиса для оплаты поиск подходящей формы осуществляется в следующем порядке: первоначально ищется локализованный xml-файл, наименование которого соответствует коду формы, например, в случае выбора сервиса с кодом 192 на англоязычном интерфейсе первоначально ищется файл *192_en.xml*. Если такой файл не найден, то ищется xml-файл, наименованием которого является код формы — *192.xml*. Если такой файл также не найден, то ТПО ищет составной файл *1xx.xml*, в котором должны располагаться формы для сервисов с кодами 100-199. Если и в составном файле *1xx.xml* форма для сервиса с кодом 192 не найдена, то поиск осуществляется в файле *forms.xml*.

В ТПО 5 версии формы располагаются в каталоге `<корень ТПО>/resources/forms/`, в ТПО 7 версии — в каталоге `<корень ТПО>/res/module/input/universal/`. Для добавления на терминал новых форм, обновления текущих, используется система обновлений, в которой размещаются новые файлы. Настройка системы обновлений описана в документе [«Кабинет агента. Программное обеспечение «Процессинговый центр Pay-logic». Руководство пользователя»](#). Для того, чтобы терминал получил обновления, необходимо создать команду на обновление из раздела «Мониторинг — Состояние оборудования» или раздела «Мониторинг — Обновление терминалов» или установить флаг обновления в разделе «Мониторинг — Состояние оборудования» (подробнее в документе [«Кабинет агента. Программное обеспечение «Процессинговый центр Pay-logic». Руководство пользователя»](#)). В системе обновлений формы располагаются в каталогах `<корень системы обновлений>/v5/<операционная система>/resources/forms/` в 5 версии ТПО и `<корень системы обновлений>/v7/<операционная система>/res/module/input/universal/` в 7 версии ТПО, в качестве ОС могут быть указаны linux или windows. Для каждой ОС используется свой каталог в системе обновлений, подробнее в документе [«Система обновлений терминального ПО. Программное обеспечение «Процессинговый центр Pay-logic». Руководство пользователя»](#).

В PMA файлы форм располагаются в каталоге `<корневой каталог PMA>/resources/forms/`. В системе обновлений располагаются в каталоге `<корневой каталог PMA>/rma/resources/forms/`.

Для Andorid-приложений формы содержатся в каталоге `../android/forms/universal/`.

Кроме того, формы могут использоваться в кабинете агента для построения форм при добавлении платежей через раздел «Диспетчерская — Поиск платежа», «Кассир — Платежи», корректировке платежей, а также валидации введенных параметров. Добавление форм в кабинет возможно с использованием двух вариантов:

1. Ручная загрузка в кабинет с использованием раздела «Справочники — Системные параметры — Загрузка файлов» (подробнее в документе [«Программное обеспечение «Процессинговый центр Pay-logic». Руководство администратора»](#)).

2. Автоматическая загрузка из системы обновлений. Настройка автоматической загрузки форм в кабинет из системы обновлений описана в документе [«Система обновлений терминального ПО. Программное обеспечение «Процессинговый центр Pay-logic». Руководство пользователя»](#).

В кабинет возможно загрузить форму по умолчанию с именем *default.xml*. Обработка формы по умолчанию поддерживается только при создании/корректировке платежей в кабинете. Если форма для сервиса не задана, то при создании/корректировке ищется форма по умолчанию, а если и она не найдена, то используется форма оплаты номера телефона.

Основными отличиями усовершенствованного модуля ввода данных от универсального являются:

1. Возможность создания правил перехода на другие экраны (для универсального модуля ввода данных можно задать только последовательность смены экранов).
2. Возможность совершения операций с данными (для универсального модуля ввода данных возможна только проверка данных и форматирование).

5 ЭЛЕМЕНТЫ ФОРМЫ ДЛЯ УНИВЕРСАЛЬНОГО МОДУЛЯ ВВОДА ДАННЫХ

5.1 ОБЩАЯ ИНФОРМАЦИЯ

Форма состоит из одного или нескольких элементов ввода различных типов. Каждый из типов элементов ввода предназначен для выполнения определенной функции (ввод текста, ввод числа, выбор и т. д.).

Основными типами элементов, используемых в формах являются:

1. Поле ввода текста.
2. Числовое поле ввода.
3. Поля для ввода даты.
4. Флаг/переключатель — элемент, позволяющий управлять параметром с двумя состояниями (вкл/выкл).
5. Селекторы различных разновидностей (выбор варианта из предложенных списков, в том числе динамических).
6. Информационные поля различных типов (обеспечивают вывод разного рода информации без возможности изменения — значения атрибутов, информацию о комиссии).
7. Другие специализированные типы элементов ввода.

Поля на форме могут быть сгруппированы в соответствии с логикой ввода данных либо отображаться индивидуально. Если группировка не используется, каждый визуальный элемент формы вводится последовательно и по одному: на экране пользователю предлагается ввести ровно одно поле, группировка позволяет на одном экране заполнить значения нескольких полей.

Дополнительно существуют элементы, которые так же являются частью формы, но не отображаются на экране — они предназначены для программного добавления атрибутов в платеж, одним из таких элементов является `<set-field>`.

Для каждого из элементов ввода определен набор атрибутов и дочерних элементов. Часть атрибутов является общей для всех типов элементов, другая часть атрибутов специфична для конкретного типа элемента.

Элемент формы, его атрибуты и дочерние элементы (при их наличии) описываются в файле формы соответствующими тегами. Файлы форм должны быть сохранены в кодировке UTF-8 в файле с расширением `.xml`. XML-файл хранит информацию о кодировке в XML-декларации (по умолчанию UTF-8 или UTF-16). Кодировка явно указывается следующим образом:

```
<?xml version="1.0" encoding="кодировка"?>
```

Тег кодировки необязателен для указания.

Корневой элемент файла форм описывается тегом `<forms>`. Тег `<forms>` объединяет в себе один или несколько элементов `<form>`.

5.2 ФАЙЛ ФОРМ (<FORMS>)

5.2.1 НАЗНАЧЕНИЕ, СТРУКТУРА КОРНЕВОГО ТЕГА ФАЙЛА ФОРМ

Корневой элемент `<forms>` указывает, что xml-документ содержит описания форм для универсального модуля ввода данных. В одном файле может быть описано неограниченное количество сервисов, для удобства элементы `<form>` в общем файле располагаются по возрастанию значения кода сервиса.

Атрибутов не имеет.

```
<forms>
  <form    ...  >
  ...
</form>
  <form    ...  >
  ...
</form>
  ...
</forms>
```

Дочерние элементы: `<form>`.

Пример:

В приведенном ниже примере файла содержится описание двух форм для сервисов с ID=111 и ID=112.

```
<forms>
  <form service-id="111"> ... </form>
  <form service-id="112"> ... </form>
</forms>
```

В качестве значения атрибута **service-id** используется код сервиса.

5.2.2 ФОРМА (<FORM>)

Форма оплаты задается в элементе `<form>`, который определяет логику и последовательность смены экранов при оплате. При активации оплаты по сервису определяется модуль ввода данных, по коду сервиса — инициализируется переменная в ТПО со значением кода сервиса. Данный механизм позволяет определять одну форму для нескольких сервисов оплаты, для которых атрибуты платежа идентичны. Такую возможность можно реализовать, поменяв КОД сервиса в кабинете для выбранных сервисов. Структура элемента:

```
<form service-id=<srvId>
  confirm-screen=<notEmptyString>
  min-sum-limit=<notEmptyString>
  max-sum-limit=<notEmptyString>
  sum=<notEmptyString>
  sum-view=<notEmptyString>
  chnum-name=<notEmptyString>
  single-check=[true|false]
  allow-return=[true|false]
  redirect-params=<notEmptyString>>
  ...
</form>
```

Дочерние элементы: `<prefill-items>` (раздел [5.2.3](#)), `<row>` (раздел [5.2.4](#)).
Атрибуты формы приведены в таблице 5.2.2.1.

Таблица 5.2.2.1 — Атрибуты формы

Атрибут	Описание	Обяз.
service-id	Код сервиса	Да
confirm-screen	Тип экрана подтверждения. На случай, если на экране подтверждения требуется отобразить более 1 поля используется запись	Нет

Атрибут	Описание	Обяз.
	confirm-screen="confirm/multiple" . Используется только в 7 версии ТПО	
chnum-name	Название атрибута, который хранит номер телефона, на который будет сохранена сдача	Нет
min-sum-limit	Ключ переменной контекста, из которой будет браться значение минимально допустимой суммы платежа	Нет
max-sum-limit	Ключ переменной контекста, из которой будет браться значение максимально допустимой суммы платежа	Нет
sum	Код альтернативного экрана суммы, который будет использоваться при оплате по сервису вместо стандартного. Экран должен существовать в используемом интерфейсе. Пример: <pre><form service-id="1123" sum="sum-banner"/></pre>	Нет
sum-view	Переменная, которая хранит значение заголовка для экрана вноса средств, по умолчанию id1	Нет
single-check	Определяет какой тип чека будет использован при печати. Возможные значения: <ul style="list-style-type: none"> • true — печатает общий чек на все платежи; • false — печатает отдельный чек на каждый платеж. Значение по умолчанию (если параметр не задан явно) 	Нет
allow-return	Позволяет осуществлять возврат к предыдущим шагам ввода данных или в главное меню после того, как денежные средства уже внесены, сохраняя сумму для дальнейшей оплаты. Возможные значения: <ul style="list-style-type: none"> • true — возврат возможен. Значение по 	Нет

Атрибут	Описание	Обяз.
	<p>умолчанию;</p> <ul style="list-style-type: none"> • false — возврат невозможен <p>В случае, когда возврат возможен, пользователь может выбрать в модуле ввода:</p> <ol style="list-style-type: none"> 1) либо выход из модуля с соответствующим формированием сдачи (выбором варианта); 2) либо вход в платежный модуль, но с новыми данными и прежним состоянием кеерг-а (то есть с сохраненной суммой внесенных денежных средств) 	
redirect-params	<p>Для передачи параметров при переходе на другой сервис используется атрибут redirect-params="..." в теге <code><form></code>.</p> <p>Для передачи всех параметров используйте значение «\$all» параметра redirect-params="...". Для того, чтобы явно указать передаваемые параметры, укажите их ключи через «,». Если redirect-params="..." не указан, но перенаправление происходит (указан redirect в <code><item></code>, раздел 6.4.2), то не будет передано ни одно значение</p>	
begin-template	<p>Позволяет указать на какой шаг будет перенаправлен пользователь при использовании шаблона платежа. Возможные значения:</p> <ul style="list-style-type: none"> • confirm — переход на шаг подтверждения платежа; • pay — переход на шаг оплаты. <p>Пример:</p> <pre data-bbox="687 1688 1374 1756"><form service-id="106" begin-template="pay"></pre> <p>Если begin-template="..." не указан, или указано</p>	Нет

Атрибут	Описание	Обяз.
	значение отличное от приведенных выше, внесение платежа будет производиться стандартно, с первого шага.	

Форма содержит элементы ввода или их группы, которые реализуют ввод данных, то есть экраны. Экраны могут быть с одним полем ввода или несколькими.

**Предупреждение!**

Объекты формы (экраны, поля) должны иметь уникальные идентификаторы.

Для ввода пользователем данных применяются клавиатуры, ПО поддерживает несколько типов клавиатур с разными возможностями ввода и различными раскладками. Тип и вид клавиатуры определяется атрибутом `keyboard` (раздел [5.3](#)). Экраны с одним полем ввода выводятся последовательно, введенное значение добавляется в набор атрибутов платежа при переходе на другой экран. Каждый атрибут платежа, определяемый для тега, имеет ключ (или наименование переменной, раздел [5.4.1](#)) и значение (введенное значение является значением переменной), в таком виде обрабатывается в процессинге. Атрибутами платежа, заданными по умолчанию в процессинге, являются ключи `id1` и `id2`. По умолчанию `id1` не может быть пустым и в качестве значения берутся первые 40 символов, `id2` — может быть пустым, ограничение на длину аналогично `id1`. Ключ `id1` в форме должен присутствовать всегда. Ограничения на другие атрибуты задаются в форме (раздел [6.1.2](#)).

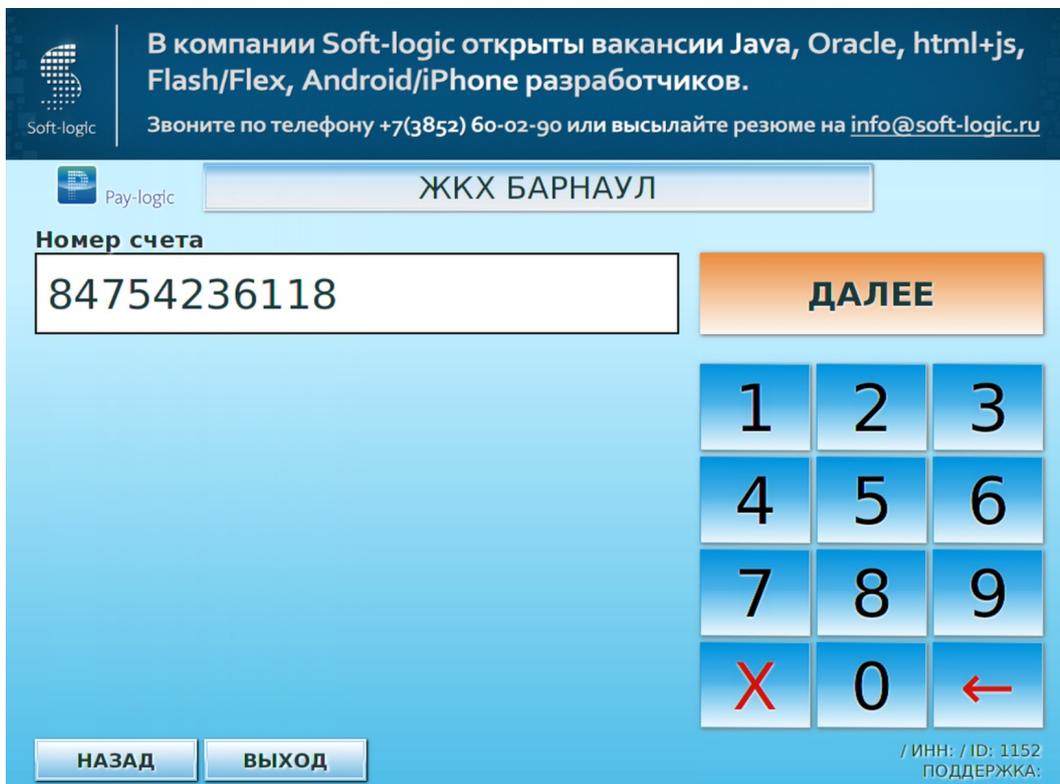
Элементы ввода: `<text-field>`, `<numeric-field>`, `<checkbox-field>`, `<text-selector>`, `<image-selector>`, `<date-field>`, `<set-field>`, `<info-field>`, `<autocomplete-field>`, `<group-field>`, `<commission-confirm>`.

Пример простой формы:

```
<?xml version="1.0" encoding="UTF-8"?>
<forms xmlns="http://pay-logic.ru">
```

```
<form service-id="1152">
  <!--Создание поля ввода, со следующими параметрами: активна цифровая
клавиатура, максимальное количество вводимых символов 11. Название поля
ввода: "Номер счета"-->
  <text-field name="account" keyboard="Digital" max-len="11"
    message="Введите номер лицевого счета" title="Номер счета">
    <!--Регулярное выражение для валидации вводимого номера-->
    <verify><rule regex="^\d{6,11}$"/></verify>
    <!--Подсказка, отображается на терминале-->
    <help><![CDATA[<html>Введите номер лицевого счета]]></help>
  </text-field>
</form>
</forms>
```

Внешний вид приведен на рисунке 5.2.2.1.



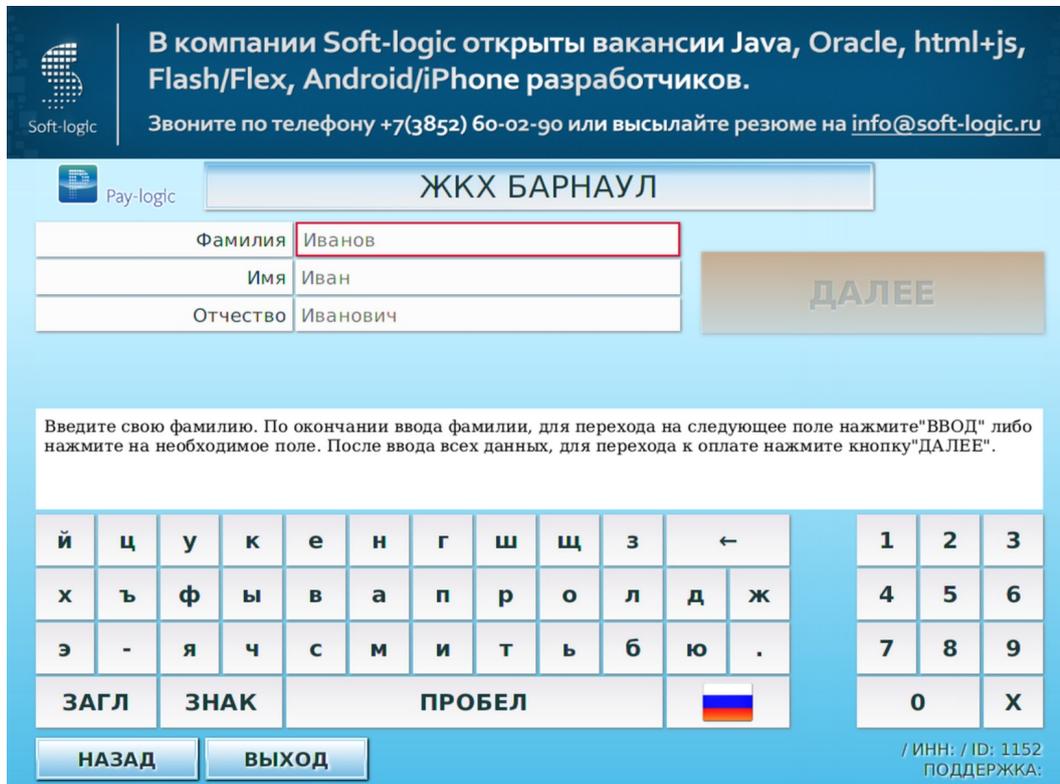
The screenshot shows a web-based payment form. At the top, there is a dark blue banner with the Soft-logic logo and text: "В компании Soft-logic открыты вакансии Java, Oracle, html+js, Flash/Flex, Android/iPhone разработчиков. Звоните по телефону +7(3852) 60-02-90 или высылайте резюме на info@soft-logic.ru". Below this, the Pay-logic logo is visible next to the text "ЖКХ БАРНАУЛ". The main form area has a light blue background. On the left, there is a label "Номер счета" above a text input field containing the number "84754236118". To the right of the input field is an orange button labeled "ДАЛЕЕ". Below the input field is a numeric keypad with buttons for digits 1-9, 0, a red 'X' for delete, and a red arrow for backspace. At the bottom left, there are two buttons: "НАЗАД" and "ВЫХОД". At the bottom right, there is small text: "/ ИНН: / ID: 1152" and "ПОДДЕРЖКА:".

Рисунок 5.2.2.1 — Пример простой формы

Пример формы с групповым экраном:

```
<?xml version="1.0" encoding="UTF-8"?>
<forms xmlns="http://pay-logic.ru">
  <form service-id="1152">
    <!--Создание экрана, отображающего несколько полей ввода
одновременно-->
    <group-field>
      <!--Создание поля ввода, со следующими параметрами: активна
цифровая, символьная и буквенная клавиатура, язык ввода - русский, без
возможности смены, максимальное количество вводимых символов 40. Название
поля ввода: "Фамилия"-->
      <text-field name="id1" keyboard="any:[ru,symb]:lower:true"
        max-len="40" example="Иванов"
        message="Введите фамилию" title="Фамилия">
        <!--Регулярное выражение для валидации вводимых данных-->
        <verify> <rule regex="^{2,}$"/> </verify>
      </text-field>
      <!--Создание поля ввода, со следующими параметрами: активна
цифровая, символьная и буквенная клавиатура, язык ввода - русский, без
возможности смены, максимальное количество вводимых символов 30. Название
поля ввода: "Имя"-->
      <text-field name="name" keyboard="any:[ru,symb]:lower:true"
        max-len="30" example="Иван"
        message="Введите имя" title="Имя">
        <!--Регулярное выражение для валидации вводимых данных-->
        <verify> <rule regex="^{2,}$"/> </verify>
      </text-field>
      <!--Создание поля ввода, со следующими параметрами: активна
цифровая, символьная и буквенная клавиатура, язык ввода - русский, без
возможности смены, максимальное количество вводимых символов 30. Название
поля ввода: "Отчество"-->
      <text-field name="patr" keyboard="any:[ru,symb]:lower:true"
        max-len="30" example="Иванович"
        message="Введите отчество" title="Отчество">
        <!--Регулярное выражение для валидации вводимых данных-->
        <verify> <rule regex="^{3,}$"/> </verify>
      </text-field>
    </group-field>
  </form>
</forms>
```

Внешний вид приведен на рисунке 5.2.2.2.



В компании Soft-logic открыты вакансии Java, Oracle, html+js, Flash/Flex, Android/iPhone разработчиков.
Звоните по телефону +7(3852) 60-02-90 или высылайте резюме на info@soft-logic.ru

Pay-logic **ЖКХ БАРНАУЛ**

Фамилия	Иванов
Имя	Иван
Отчество	Иванович

ДАЛЕЕ

Введите свою фамилию. По окончании ввода фамилии, для перехода на следующее поле нажмите "ВВОД" либо нажмите на необходимое поле. После ввода всех данных, для перехода к оплате нажмите кнопку "ДАЛЕЕ".

й	ц	у	к	е	н	г	ш	щ	з	←	1	2	3	
х	ъ	ф	ы	в	а	п	р	о	л	д	ж	4	5	6
э	-	я	ч	с	м	и	т	ь	б	ю	.	7	8	9
ЗАГЛ	ЗНАК	ПРОБЕЛ									0	X		

НАЗАД **ВЫХОД**

/ ИНН: / ID: 1152
ПОДДЕРЖКА:

Рисунок 5.2.2.2 — Пример простой формы

5.2.3 ПОЛУЧЕНИЕ ДАННЫХ ИЗ СПРАВОЧНИКА ТОЧКИ (<PREFILL-ITEMS>)

Элемент, описываемый тегом `<prefill-items>`, предназначен для добавления данных из справочника точки в контекст сценария.

Структура элемента:

```
<prefill-items>
  <item from=<notEmptyString25>
    to=<notEmptyString25>
    title=<notEmptyString40/>
  </prefill-items>
```

Дочерние элементы: `<item>`.

Атрибуты для ТПО:

1. **dealer.id** — идентификатор агента.
2. **dealer.name** — наименование агента.
3. **dealer.inn** — ИНН агента.
4. **dealer.address** — адрес агента.
5. **dealer.support** — контакты поддержки.
6. **dealer.supportHours** — время работы поддержки.
7. **dealer.contractNo** — номер договора.
8. **dealer.contractDate** — дата договора.
9. **dealer.account** — расчетный счет.

-
10. **dealer.bank** — банк.
 11. **dealer.bik** — БИК банка.
 12. **dealer.kpp** — КПП агента.
 13. **dealer.okato** — ОКАТО агента.
 14. **dealer.ks** — корреспондентский счет.
 15. **service.id** — ID сервиса.
 16. **service.code** — код сервиса.
 17. **service.barcodePrefix** — префикс штрих-кода.
 18. **service.name** — сервис.
 19. **service.printName** — название для печати.
 20. **service.altName** — альтернативное название.
 21. **service.invitation** — текст приглашения.
 22. **service.minSum** — минимальная сумма платежа.
 23. **service.maxSum** — максимальная сумма платежа.
 24. **service.maxInputSum** — максимальная вложенная сумма платежа.
 25. **service.divideSum** — сумма разбивки платежа.
 26. **service.verifyType** — тип верификации.
 27. **service.processingType** — тип проведения.
 28. **service.minComm** — минимальная комиссия.
 29. **service.paymentSources** — источники оплаты.

-
30. **service.getMaxComm()** — комиссия, такие коды связаны с особенностями velocity (отсутствует стандартный геттер для поля). Есть альтернативное название «commission».
31. **service.getOutCurrency()** — валюта провайдера, альтернативное название «out-currency».
32. **service.getInCurrency()** — валюта сервиса, альтернативное название «in-currency».
33. **point.id** — номер точки.
34. **point.code** — код точки.
35. **point.name** — название точки.
36. **point.address** — адрес точки.
37. **point.city** — город, в котором находится точка.
38. **point.cityCode** — телефонный код города, в котором находится точка.
39. **point.area** — регион, в котором находится точка.
40. **point.params** — параметры точки.
41. **point.incassLimit** — лимит точки до следующей инкассации.
42. **Point.pointType** — тип точки.

В 5 версии ТПО возможно вывести произвольные свойства объектов **dealer** (агент в соответствии с направлением проведения) и **point** (точка), используя синтаксис:

```
dealer.properties.<код произвольного свойства>
```

и

```
point.attributes.<код произвольного свойства>
```

соответственно.

**Внимание!**

Для отправки произвольного свойства на точку в настройках произвольного свойства в разделе «Справочники — Система — Типы свойств объектов» установите флаг **«Включать в справочники точки (отправка на точку)»**.

Атрибуты элемента `<item>` приведены в таблице 5.2.3.1.

Таблица 5.2.3.1 — Атрибуты элемента `<item>`

Атрибут	Описание	Обяз.
<code>from</code>	Указывает, какой элемент берется из справочника берем	Да
<code>to</code>	Указывает какому атрибуту платежа присваивается значение	Да
<code>title</code>	Указывает заголовок для полученного атрибута, если атрибут не указан, то будет использовано название по умолчанию	Нет

Пример:

Название города из справочника присваивается атрибуту `city`.

```
<prefill-items>
  <item from="point.city"
    to="city"
    title="Город"/>
</prefill-items>
```

Пример формы для вывода произвольных свойств (5 версия ТПО):

```
<form service-id="1152">
  <prefill-items>
    <item from="dealer.properties.recipient_name" to="d1"/>
    <item from="point.attributes.fp" to="d2"/>
  </prefill-items>
```

```
<!--Создание поля ввода, со следующими параметрами: активна цифровая,  
символьная и буквенная клавиатура с возможностью переключения языка,  
максимальное количество вводимых символов 10. Название поля ввода: "Номер  
счета"-->  
<text-field name="id1" keyboard="any:[ru,en]:upper:true" max-len="10"  
  message="Введите номер счета" title="Номер счета">  
<!--Регулярное выражение для валидации вводимого номера-->  
<verify>  
  <rule regex="^.*$"/>  
</verify>  
<!--Подсказка, отображается на экране-->  
<help>  
  Введите номер счета  
</help>  
</text-field>  
</form>
```

Для ТПО 5 и РМА также была добавлена обработка в **<prefill-items>** атрибута **point.pointType**. Данный атрибут позволяет передавать тип точки с сервера в составе справочника точки. Возможные типы точек и соответствующие им значения кодов приведены в таблице 5.2.3.2.

Таблица 5.2.3.2 — Значения атрибута **point.pointType**

Тип точки	Наименование типа точки	Код
APPARAT	Терминал	0
OFFICE	Офис	1
MOBILE_POINT	Мобильная точка	2
GATE	Шлюз	3
RMA	РМА	4
GATE2	Шлюз по сертификату	5
GATE_CYBER	Шлюз по протоколу 2	6
CASH	Касса с выдачей	7
POS	POS-терминал	8
VPS_POINT	Точка ВПС	9

Тип точки	Наименование типа точки	Код
EXCHANGER	Обменник	10
CELLPHONE	Мобильный телефон	11
WEB_KIOSK	Web Kiosk	12

Пример конструкции для получения значения типа точки в сценарии:

```
<item from="point.pointType" to="id2" title="Тип точки"/>
```

Пример конструкции для отображения полученного значения типа точки в сценарии:

```
<screen type="confirm" title="Подтверждение данных платежа" id="confirm">  
<!--Список атрибутов, отображаемых на экране подтверждения-->  
<fields list="id2"/>  
</screen>
```

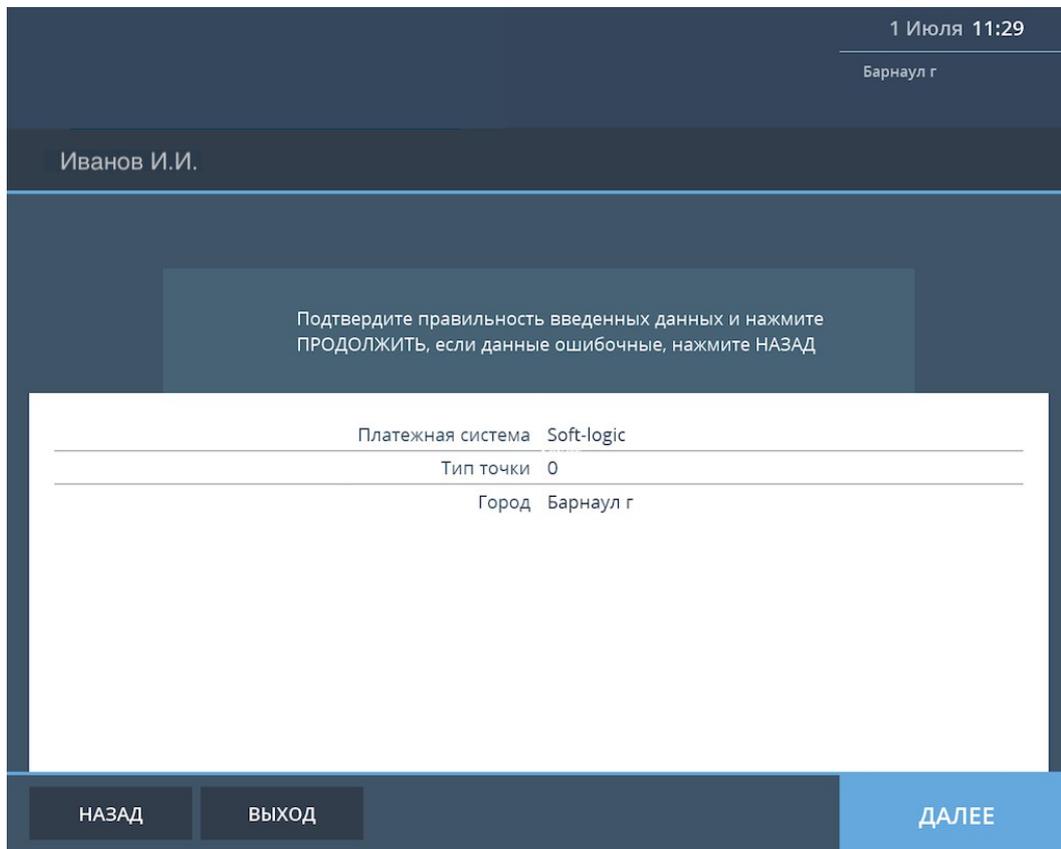
Пример сценария с использованием данной конструкции:

```
<scenario begin="1screen">  
<!--Добавление данных из справочников точки в контекст сценария-->  
<prefill-items>  
<item from="paymentSystem.name" to="id1"  
key-title-id="Платежная система"/>  
<item from="point.pointType" to="id2" key-title-id="Тип точки"  
flags="HIDE_ON_CONFIRM"/>  
<item from="point.city" to="id3" key-title-id="Город"/>  
<item from="menuItem.service.properties(account1)" to="id4"  
key-title-id="Расчетный счет"/>  
</prefill-items>  
<!--Создание экрана ввода данных с несколькими полями-->  
<screen type="group" title="Введите данные" id="1screen">  
<fields>  
<!--Создание поля ввода, со следующими параметрами: активна цифровая,  
символьная и бувенная клавиатура, язык ввода - русский без возможности  
переключения языка, максимальное количество вводимых символов 60.  
Название  
поля ввода: "ФИО плательщика". Атрибут отправляется в маскированном  
виде-->  
<text-field id="fio" title="ФИО плательщика" max-len="60"
```

```
keyboard="any:[ru,symb]:upper:true" flags="FLAG_MASK_IN_LOGS">
<!--Регулярное выражение для валидации вводимых данных-->
<validator type="regex">
<rules>
<rule regex=".*"/>
</rules>
</validator>
<!--Подсказка, отображается на экране-->
<help> <![CDATA[Введите ФИО плательщика]]> </help>
</text-field>
<!--Создание поля ввода, со следующими параметрами: активна цифровая,
символьная и бувенная клавиатура, язык ввода – русский без возможности
переключения языка, максимальное количество вводимых символов 254.
Название
поля ввода: "Адрес плательщика"-->
<text-field id="address" title="Адрес плательщика" max-len="254"
keyboard="any:[ru,symb]:upper:true">
<!--Регулярное выражение для валидации вводимых данных-->
<validator type="regex">
<rules> <rule regex="^.{3,254}$"/> </rules>
</validator>
<!--Подсказка, отображается на экране-->
<help> <![CDATA[Введите ФИО плательщика]]> </help>
</text-field>
<!--Создание поля выбора типа оплаты-->
<selector-field id="type" title="Тип оплаты">
<items type="static">
<item value="21" title="Паспорт РФ"/>
<item value="22" title="Заграничный паспорт"/>
<item value="31" title="Паспорт иностранного гражданина"/>
</items>
</selector-field>
</fields>
<actions>
<!--Описание поведения сценария оплаты при нажатии кнопки "Далее"-->
<action type="Next" title="Далее">
<goto target="confirm"/>
</action>
<!--Описание поведения сценария оплаты при нажатии кнопки "Назад"-->
<action type="Prev" title="Назад">
<goto target="previos"/>
</action>
<!--Описание поведения сценария оплаты при нажатии кнопки "Далее"-->
```

```
<action type="Exit" title="Далее">
<goto target="exit"/>
</action>
</actions>
</screen>
<!--Создание экрана подтверждения-->
<screen type="confirm" title="Подтверждение данных платежа"
id="confirm">
<!--Список атрибутов, отображаемых на экране подтверждения-->
<fields list="id1,id2,id3,id4"/>
<actions>
<!--Описание поведения сценария оплаты при нажатии кнопки "Далее"-->
<action type="Next" title="Далее">
<goto target="pay"/>
</action>
<!--Описание поведения сценария оплаты при нажатии кнопки "Назад"-->
<action type="Prev" title="Назад">
<goto target="lscreen"/>
</action>
<!--Описание поведения сценария оплаты при нажатии кнопки "Выход"-->
<action type="Exit" title="Выход">
<goto target="exit"/>
</action>
</actions>
</screen>
</scenario>
```

При выводе значения на экране ТПО отображается код, характеризующий тип данной точки (значения типов точек и соответствующие им коды приведены в таблице 5.2.3.2). Пример отображения полученных данных о типе точки представлен на рисунке 5.2.3.1.



1 Июля 11:29
Барнаул г

Иванов И.И.

Подтвердите правильность введенных данных и нажмите ПРОДОЛЖИТЬ, если данные ошибочные, нажмите НАЗАД

Платежная система	Soft-logic
Тип точки	0
Город	Барнаул г

НАЗАД ВЫХОД ДАЛЕЕ

Рисунок 5.2.3.1 — Пример вывода типа точки

5.2.4 ОБЪЕДИНЕНИЕ ПОЛЕЙ В СТРОКУ (<ROW>)

Элемент `<row>` позволяет разместить указанные внутри него поля ввода в одну строку. В текущей версии реализовано только для «Платформы электронных кошельков Smartkeeper».

Синтаксис:

```
<form service-id="1152">
  <row orientation=[PORTRAIT|LANDSCAPE]>
    <text-field width=<nonNegativeInteger>
      ...
    </text-field>
    ...
    <placeholder width=<nonNegativeInteger>/>
    ...
  </row>
</form>
```

Значения `PORTRAIT` и `LANDSCAPE` задают для какой ориентации используется размещение в одну строку: `PORTRAIT` — вертикальная, `LANDSCAPE` — горизонтальная.

При таком варианте использования для всех полей модели доступны атрибуты **width**, **rowId**.

Атрибут **width** определяет ширину поля, для которого он указан, в текущей строке в процентах. Если сумма значений **width** всех полей внутри `<row>` превышает 100, то возникнет ошибка при обработке сценария. Если сумма значений **width** всех полей внутри `<row>` превышает 100, но **width** указан не для всех полей, то возникнет ошибка при обработке сценария. Если значение не указано, то считается, что такое поле занимает все свободное место. Если таких полей несколько, то они делят все свободное место поровну.

Атрибут **rowId** определяет идентификатор строки, в которой будет расположено поле.

Элемент `<placeholder>` является полем ввода, но не несет никакой полезной нагрузки кроме значения атрибута **width**.

5.3 КЛАВИАТУРЫ

5.3.1 КЛАВИАТУРЫ В 7 ВЕРСИИ ТПО

В ТПО 5 и 7 версии предусмотрены следующие типы клавиатур:

1. Цифровая клавиатура — позволяет вводить пользователю только цифры, и возможно — десятичный разделитель. Цифровая клавиатура задается кодом Digital (рисунок 5.3.1.1).

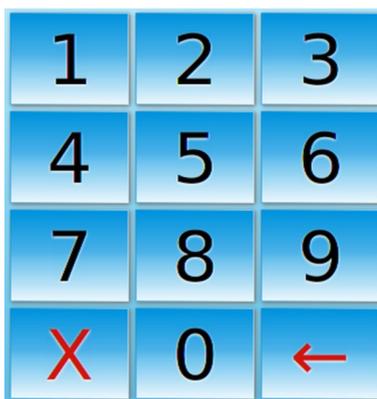


Рисунок 5.3.1.1 — Digital-клавиатура

2. Вместо функциональной кнопки «X» (очистить) возможно указать символ из символьной или буквенной раскладки клавиатуры. Например, отображение клавиатуры, с косой чертой вместо символа «X» (очистить) приведено на рисунке 5.3.1.2.

3. Полноценная (полноразмерная клавиатура). Внешний вид клавиатуры определяется опциями (any|letter|digital):(\([[a-z]{1,5},)*[a-z]{1,5}\])(:(lower|upper):(true|false)), которые позволяют настраивать отображение различных символов на клавиатуре, язык, регистр букв, возможность смены регистра:

1) часть (any|letter|digital) задает используемые символы, возможно использовать только одно из значений:

- a) any — все символы;
- b) letter — буквы и цифры;
- c) digital — только цифры.

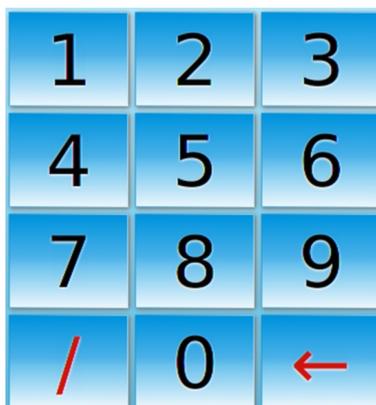


Рисунок 5.3.1.2 — Тип клавиатуры Digital с заменённой функциональной кнопкой «Стереть»

2) часть (\([[a-z]{1,5},)*[a-z]{1,5}\]) задает алфавит, наличие переключателя на символную раскладку, возможность использования только цифровой раскладки на групповом экране:

- a) ru — русский алфавит, en — английский алфавит и т. п.;
- b) symb — наличие переключателя на символную раскладку.

3) часть (lower|upper) задает регистр букв в раскладке по умолчанию (используется с letter раскладкой). Возможно использовать только одно из значений:

a) lower — нижний регистр;

b) upper — верхний регистр;

с) для интерфейса «Smoke» добавлен вариант клавиатуры «mixed» — смешанный регистр. Первая буква и каждая буква после пробела прописные, остальные — строчные. По запросу может быть внедрена поддержка типа клавиатуры в других типах интерфейсов.

4) часть (true|false) задает возможность смены регистра при вводе данных. Возможно использовать только одно из значений:

a) true — допустимо менять регистр символов при вводе;

b) false — невозможна смена регистра.

Части разделяются символом ":". Пример:

```
keyboard="any:[ru,symb]:lower:true"
```

Отображение клавиатуры, заданной записью «any:[ru,symb]:lower:true» приведено на рисунке 5.3.1.3.

й	ц	у	к	е	н	г	ш	щ	з	←		1	2	3
х	ъ	ф	ы	в	а	п	р	о	л	д	ж	4	5	6
э	-	я	ч	с	м	и	т	ь	б	ю	.	7	8	9
ЗАГЛ	ЗНАК	ПРОБЕЛ									0	X		

Рисунок 5.3.1.3 — Клавиатура any:[ru,symb]:lower:true

5.4 ПОЛЕ ВВОДА

5.4.1 ОБЩИЕ ПОЛОЖЕНИЯ

Поле ввода — элемент формы, описывающий поле, заполняемое пользователем на экране. Общие для всех элементов ввода атрибуты можно разделить на две группы: основные атрибуты и атрибуты для интернационализации (перевода). Эти атрибуты являются общими для всех без исключения типов полей. В дальнейшем, при рассмотрении синтаксиса, описывающего конкретные типы полей, общие атрибуты подробно не разбираются. Основные атрибуты приведены в таблице 5.4.1.1.

Таблица 5.4.1.1 — Общие атрибуты элементов ввода

Атрибут	Описание	Обяз.
name	Код атрибута платежа (идентификатор поля)	Да
title	Заголовок поля ввода. Предназначен для вывода пользователю названия вводимого поля (например, «Номер телефона»)	Да
message	Информационное сообщение (подсказка) для вводимого поля	Нет
on-top	Позволяет редактировать значение поля ввода во всплывающем окне	Нет
group-id	Значение, по которому происходит группировка полей на экране, то есть поля, для которых указано одно и то же значение group-id , визуально будут объединены на экране в один блок. Поддерживается некоторыми интерфейсами	Нет
flags	Флаги, дополнительные свойства данного поля. Поддерживаемые флаги приведены в разделе 5.4.2	Нет

Атрибут	Описание	Обяз.
width	Ширина данного поля в текущей строке в процентах, если используется конструкция <code><row></code> . Если сумма значений width всех полей внутри <code><row></code> превышает 100, то возникнет ошибка при обработке сценария. Если сумма значений width всех полей внутри <code><row></code> превышает 100, но width указан не для всех полей, то возникнет ошибка при обработке сценария. Если значение не указано, то считается, что такое поле занимает все свободное место. Если таких полей несколько, то они делят все свободное место поровну. Поддержка атрибута зависит от интерфейса. В текущей версии реализовано только для «Платформы электронных кошельков Smartkeeper».	Нет
rowId	Идентификатор строки, в которой будет расположено поле. Поддержка атрибута зависит от интерфейса. В текущей версии реализовано только для «Платформы электронных кошельков Smartkeeper».	Нет

Формы могут быть локализованы. Программное обеспечение поддерживает несколько механизмов локализации (раздел [5.4.3](#)).

Пример использования `on-top`:

```
<?xml version="1.0" encoding="UTF-8"?>
<form service-id="1152">
  <group-field>
    <!--Создание поля ввода, со следующими параметрами: активна
    символная и буквенная клавиатура с возможностью переключения языка
    смены, максимальное количество вводимых символов 50. Название поля ввода:
    "E-mail"-->
    <text-field name="id1" keyboard="letter:[ru,en,symb]:lower:true"
      max-len="50" message="Введите E-mail" title="E-mail">
```

```
<!--Регулярное выражение для валидации вводимых данных-->
<verify>
  <rule regex="^{1,50}$"/>
</verify>
<!--Подсказка, отображается на экране-->
<help>
  <![CDATA[Введите E-mail ]]>
</help>
</text-field>
<!--Создание поля ввода, со следующими параметрами: активна цифровая
клавиатура, максимальное количество вводимых символов 7. Название поля
ввода: "Номер карты"-->
<text-field name="wc_cardnumber" keyboard="Digital" max-len="7"
  message="Номер бонусной карты" title="Номер карты" >
  <!--Регулярное выражение для валидации вводимого номера-->
  <verify>
    <rule regex="^\d{4,7}$|"/>
  </verify>
  <!--Подсказка, отображается на терминале-->
  <help>
    Введите номер Вашей бонусной карты. Поле не обязательно для
    заполнения
  </help>
</text-field>
</group-field>
</form>
```

В этом примере при редактировании поля ввода E-mail будет отображаться всплывающее окно (рисунок 5.4.1.1).

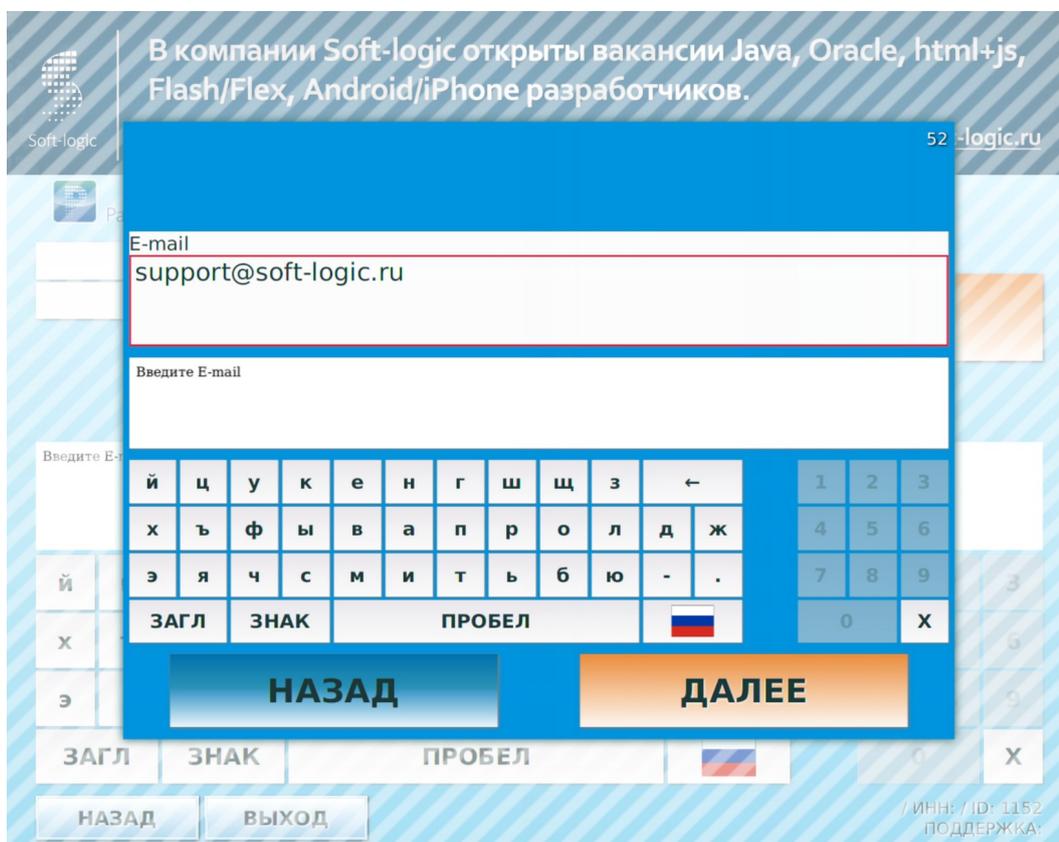


Рисунок 5.4.1.1 — Редактирование значения поля ввода во всплывающем окне

Для числового поля необходимо указывать `on-top="numeric/popup"` (рисунок 5.4.1.2).

Пример:

```
<?xml version="1.0" encoding="UTF-8"?>
  <form service-id="1152">
    <numeric-field decor="popup" name="id3"
      message="Введите сумму платежа" title="Сумма" min="0"
      max="15000" on-top="numeric/popup"/>
  </form>
```

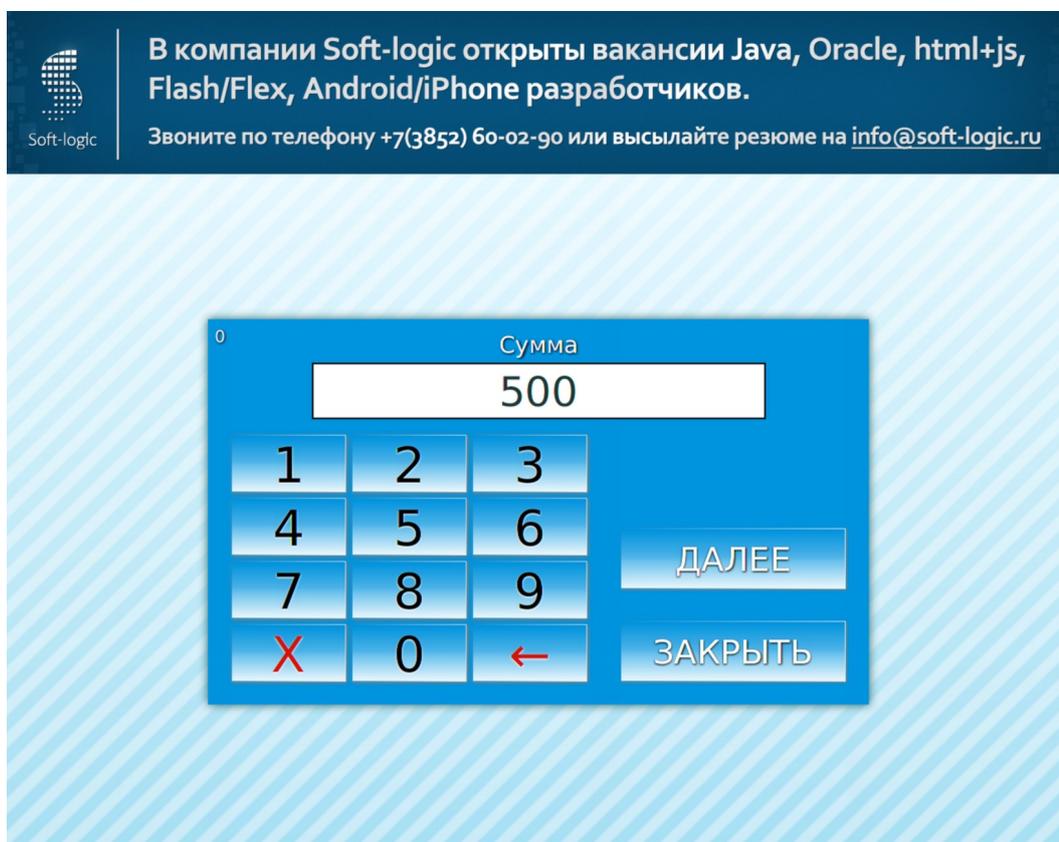


Рисунок 5.4.1.2 — Редактирование значения числового поля ввода во всплывающем окне

5.4.2 ФЛАГИ

Флаги могут задаваться как числом, так и символьными константами. Может указываться одновременно несколько флагов через символ «|» (вертикальная черта). Доступные флаги приведены в таблице 5.4.2.1.

Таблица 5.4.2.1 — Флаги

Символьная константа	Числовое значение	Описание
HIDE_ON_CONFIRM	0x01	Не показывать на экране подтверждения. Доступен на ТПО и РМА
HIDE_ON_PRINT	0x02	Не выводить на печать в чеке. Используется, в том случае когда атрибут платежа, не нужно печатать на чеке. В шаблоне чека необходимо использовать цикл: <pre>#foreach(\$key in \$operation.inputElements) \$key.keyTitle: \$key.value #end</pre> key — ключ атрибута. Доступен на ТПО и РМА
HIDE_ON_SERVER	0x04	Не показывать в атрибутах платежа в кабинете. Распространяется на атрибуты id1 , id2 . Администратору ПС атрибут отображается вне зависимости от значения флага. Доступен на ТПО и РМА
HELP_KEEPER_ANDROID	0x08	Подсказка для Android клиентов
HIDE_FROM_EXTERNAL	0x100	Запрет отдачи атрибута внешним агентам

Символьная константа	Числовое значение	Описание
HIDE_ON_FRONTEND	0x200	Не отображать атрибут плательщику. Используется, в том случае когда для совершения платежа передается служебный атрибут, который не нужно показывать пользователю. Доступен в электронном кошельке. Для роли пользователя «Администратор» атрибут не скрывается
USE_CONTACTS	0x10	Возможность использования контактов из телефонной книги. Используется на мобильных клиентах электронного кошелька
VALIDATE_BY_CAPACITY	0x20	Использовать валидацию по ёмкостям. Предназначен для полей, содержащие телефонные номера
HELP_MOBILE_ANDROID	0x40	Подсказка для терминалов Android
HELP_MOBILE_IOS	0x80	Подсказка для терминалов IOS
RECEIVED_FROM_SERVER	0x300	Атрибут был получен от сервера. Используется при онлайн-запросах. Для использования в коде сценария недоступен. Используется разработчиками
VALUE_IS_UNCHANGED	0x400	Позволяет пометить input-элементы, как не редактируемые
FLAG_MASK_IN_LOGS	0x800	Позволяет отправлять атрибуты в логах в маскированном виде. При этом originalValue , valueTitle не передаются, а в value каждый оригинальный символ значения заменяется на символ «*». На атрибуты

Символьная константа	Числовое значение	Описание
		id1, id2 не действует
FLAG_MASK_CARD_IN_LOGS	0x8000	Позволяет шифровать в логах номера карт по правилу $^{\backslash d\{6\} \backslash d\{4\}\backslash s\backslash d\{2\}}.*(\backslash d\{4\})\$ \Rightarrow \$1*****\$2$
FLAG_DO_NOT_SAVE_TO_DB	0x40000	Атрибут не требует сохранения в БД. При сохранении результатов операции, атрибут с таким флагом будет удален из БД
FLAG_ATTRIBUTE_ENCRYPTED	0x40000000	Позволяет шифровать атрибуты на стороне сервера

Пример сценария с маскированием значения атрибута в логах (7 версия ТПО):

```
<?xml version="1.0" encoding="UTF-8"?>
<forms>
  <form service-id="1152">
    <!--Создание поля ввода, со следующими параметрами: активна цифровая
    клавиатура, максимальное количество вводимых символов 11. Название поля
    ввода: "Телефон"-->
    <text-field name="id1" keyboard="Digital" max-len="11" prefix="7"
      example="903333333"
      message="Введите номер своего сотового телефона"
      title="Телефон" flags="0x800">
      <!--Регулярное выражение для валидации вводимого номера-->
      <verify>
        <rule regex="7\d{10}$"/>
      </verify>
      <!--Маска для форматирования вводимого номера, применяется по умолчанию-->
      <split default="+*-***-***-***"/>
    </text-field>
    <numeric-field name="sum"
```

```
message="Введите сумму платежа (от 10р до 15000р)"  
title="Сумма платежа" allow-decimal="true" min="10"  
max="15000" flags="FLAG_MASK_IN_LOGS"/>  
</form>  
</forms>
```

В маскированном виде будут переданы **value** атрибутов **id1**, **sum**.

Фрагмент network.log:

```
16:05:41,236 INFO [JavaFX Application Thread] Add operation:  
Operation[service=Лотерея, inputElements={id1=InputElement{key=id1,  
keyTitle=Телефон, keyTitleId=null, value=*****, flags=100000000000},  
sum=InputElement{key=sum, keyTitle=Сумма платежа, keyTitleId=null, value=***,  
flags=100000000000}}, form={id1=79634567663, sum=500}, views={id1=+7-963-456-  
7663, sum=500}, titles={id1=Телефон, sum=Сумма платежа}, id=1, check=22964,  
pinCode=null, pinCodeNominal=null, date=Wed Mar 22 16:05:29 NOVT 2017,  
change-date=null, fraudControlRule=null]
```

5.4.3 МЕХАНИЗМЫ ЛОКАЛИЗАЦИИ ФОРМ

Существуют два механизма локализации форм. Первый предполагает создавать для каждого языка отдельный файл форм с кодом локали, например, *110_ru.xml*, *110_en.xml* и т. д. Язык подгружаемой формы сопоставляется с выбранным языком пользователя. Такой же механизм локализации поддерживается и для утилитных форм (копия чека, проверка статуса платежа и другие), лежащих в каталоге **<корень ТПО>/resources/utility**.

Другой механизм локализации предполагает выносить коды текстовок в отдельный файл. Ресурсные файлы с текстовками в ТПО 7 версии расположены в каталоге **<корень ТПО>/res/module/i18n/** в файлах *input.properties*, *input_<код локали>.properties*, в ТПО 5 версии — **<корень ТПО>/resources/i18n/** в файлах *input.properties*, *input_<код локали>.properties*. Код текстовки для того или иного языка указывается с помощью определенных атрибутов. Замена осуществляется автоматически при смене пользователем языка интерфейса.

Атрибуты, используемые для интернационализации, приведены в таблице 5.4.3.1, а также в описании элементов формы в соответствующих разделах (такие атрибуты содержат в названии «-id»).

Таблица 5.4.3.1 — Атрибуты, используемые для интернационализации

Атрибут	Описание	Обяз.
message-id	Идентификатор текстовки для сообщения (подсказки). Если для текущего выбранного языка текстовка в ресурсном файле отсутствует — используется текстовка из атрибута message	Нет
title-id	Идентификатор текстовки для заголовка (титла)	Нет

Атрибут	Описание	Обяз.
	поля. Если для текущего выбранного языка текстовка в ресурсном файле отсутствует — используется текстовка из атрибута title	

Фрагмент локализованной формы:

```
<!--Создание поля ввода, со следующими параметрами: активна цифровая,
символьная и буквенная клавиатура, язык ввода - русский, без возможности
смены, максимальное количество вводимых символов 80. Название поля ввода:
"Фамилия, имя, отчество"-->
<text-field name="id3" keyboard="any:[ru,symb]:upper:true" max-len="80"
      message="Петров Петр Петрович" message-id="m.1" title="Фамилия,
имя, отчество">
  <!--Регулярное выражение для валидации вводимых данных-->
  <verify>
    <rule regex="^[А-Яа-яёЁ]{3,}(\s+[А-Яа-яёЁ]{1,}\.?)\{0,2\}$"/>
  </verify>
</text-field>
<!--Создание поля ввода, со следующими параметрами: активна цифровая,
символьная и буквенная клавиатура, язык ввода - русский, без возможности
смены, максимальное количество вводимых символов 80. Название поля ввода:
"Населенный пункт"-->
<text-field name="id4" keyboard="any:[ru,symb]:upper:true" max-len="80"
      message="г. Барнаул" default="г. Барнаул" message-id="m.2"
      title="Населенный пункт">
  <!--Регулярное выражение для валидации вводимых данных-->
  <verify>
    <rule regex="^[А-Яа-яёЁ\d \.-]{5,}$"/>
  </verify>
</text-field>
<!--Создание поля ввода, со следующими параметрами: активна цифровая,
символьная и буквенная клавиатура, язык ввода - русский, без возможности
смены, максимальное количество вводимых символов 80. Название поля ввода:
"Адрес"-->
<text-field name="id5" keyboard="any:[ru,symb]:upper:true" max-len="80"
      message="ул. 10-я Западная 5 кв 1" message-id="m.3" title="Адрес">
  <!--Регулярное выражение для валидации вводимых данных-->
  <verify>
    <rule regex="^\([А-Яа-яёЁ\d-]{1,}\.?\)(\s*[А-Яа-яёЁ\d-]{1,}\.?)\*$"/>
  </verify>
</text-field>
```

```
</verify>  
</text-field>
```

В файле с текстовками данные записываются в формате <код текстовки>=<значение>.

5.5 ПРОСТЫЕ ТИПЫ ДАННЫХ ЗНАЧЕНИЙ АТТРИБУТОВ ЭЛЕМЕНТОВ, ИСПОЛЬЗУЕМЫХ НА ЭКРАНАХ ФОРМ

Формат входных данных, обрабатываемых формами, описывается input-моделью. Модель содержит описание доступных в формах элементов, их атрибутов и типов данных значений атрибутов. В модели используются простые и комплексные типы данных. Комплексные типы содержат набор предопределенных значений. Простые типы, как правило, описывают требования к содержанию значения атрибута. Однако они так же могут содержать предопределенные значения. Ниже приведен список простых типов данных, используемых в input-модели, не имеющих предопределенных значений:

1. **resourceIdType** — идентификатор. Идентифицирует сущность, определенную в другом месте. Представлен строкой, содержащей от 2 до 30 символов, первый символ обязательно строчная буква латинского алфавита, последующие символы: цифры от 0 до 9, строчные буквы латинского алфавита и символы «.», «_», «-».
2. **srvId** — идентификатор сервиса, целое положительное число, минимальное значение: «0», максимальное: «32767».
3. **floatSum** — десятичное число с двумя цифрами в дробной части. Минимальное значение «0.00».
4. **notEmptyString(n)** — текстовая строка длиной до n символов. Если n не указано, то ограничение на максимальную длину отсутствует.
5. **positiveFloat** — положительное вещественное число, минимальное значение «0».
6. **fileNameType** — имя файла. Например, «resources/ewalletoffer.html».

7. **messageType** — непустая строка от 1 до 40 символов. Первый символ — прописная буква, или цифра, или символ «№», следующие: любые. Последний: обязательно непробельный символ.

8. **titleType** — непустая строка от 1 до 30 символов. Первый символ: прописная буква, или цифра, или символ «№», следующие: любые. Последний: обязательно непробельный символ.

9. **selectorTitleType** — непустая строка от 1 до 35 символов. Первый символ: прописная буква, или цифра, или символ «№», следующие: любые за исключением пробелов, табуляций и переносов строк. Последний: обязательно непробельный символ.

Комплексные типы данных и простые типы с predefined значениями будут описаны при рассмотрении атрибутов с соответствующими типами данных.

6 ОСНОВНЫЕ ТЕГИ

6.1 ТЕКСТОВОЕ ПОЛЕ ВВОДА (<TEXT-FIELD>)

6.1.1 НАЗНАЧЕНИЕ И СТРУКТУРА ТЕКСТОВОГО ПОЛЯ ВВОДА (<TEXT-FIELD>)

Текстовое поле предназначено для ввода цифровой и буквенной информации. Описывается тегом `<text-field>`. Внешний вид представлен на рисунках 6.1.1.1, 6.1.1.2.

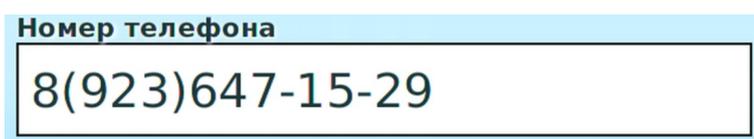


Рисунок 6.1.1.1 — Текстовое поле ввода

Структура текстового поля ввода:

```
<text-field name=<string>
  title=<titleType>
  message=<messageType>
  message-id=<resourceIdType>
  title-id=<resourceIdType>
  on-top=<notEmptyString>
  group-id=<nonNegativeInteger>
  flags=<flags_specification>
  keyboard=<keyboard_specification>
  max-len=<short>
  prefix=<notEmptyString25>
  postfix=<notEmptyString25>
  read-only=[true|false]
  default=<notEmptyString>
  example=<string>
  xbutton=<notEmptyString1>
  secure=[true|false]
  fail-regex=<string>
```

```

trailing-widget=<string>
width=<nonNegativeInteger>
rowId=<nonNegativeInteger>>
<verify> ... </verify>
<help> ... </help>
<split> ... </split>
<replaces> ... </replaces>
<filter >... </filter>
<barcode-scanner> ... </barcode-scanner>
</text-field>
    
```

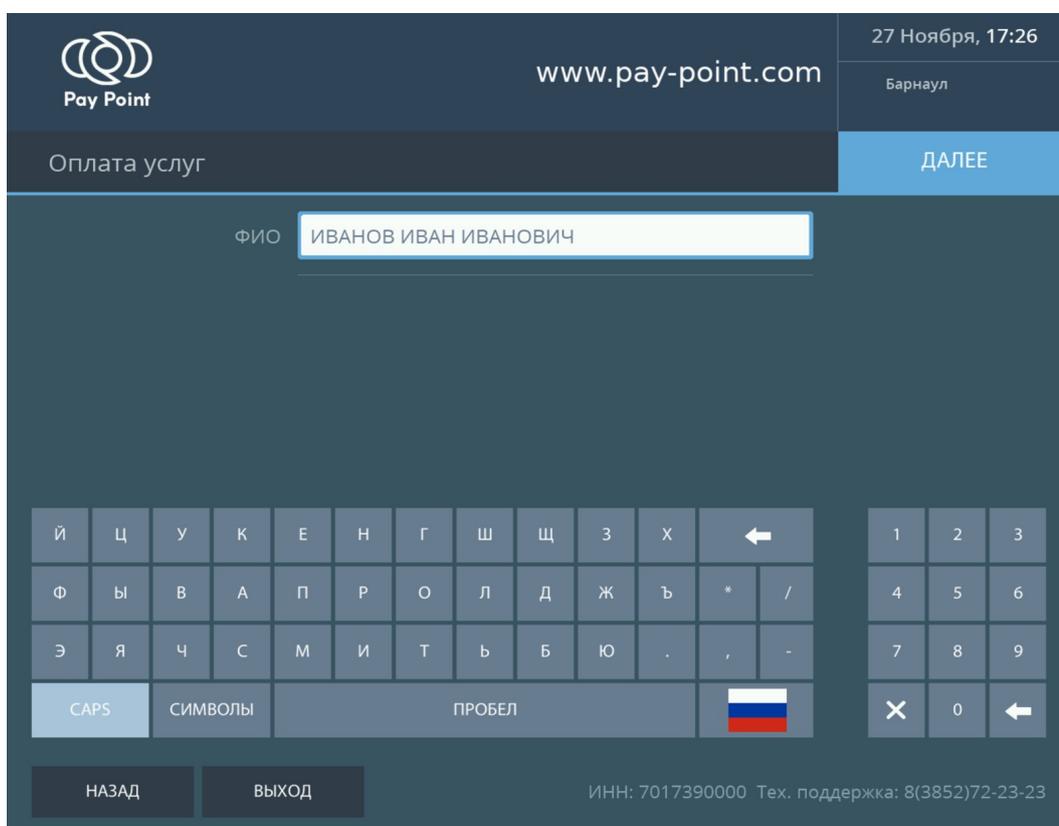


Рисунок 6.1.1.2 — Пример текстового поля ввода для интерфейса «Smoke»

Дочерние элементы: <verify> (раздел [6.1.2](#)), <help> (раздел [7.1](#)), <split> (раздел [6.1.3](#)), <replaces> (раздел [6.1.4](#)), <filter> (раздел [6.1.5](#)), <barcode-scanner> (раздел [7.2](#)).

Описание атрибутов приведено в таблице 6.1.1.1.

Таблица 6.1.1.1 — Атрибуты текстового поля ввода

Атрибут	Описание	Обяз.
name	Код атрибута платежа (идентификатор поля)	Да
title	Заголовок поля ввода. Предназначен для вывода пользователю названия вводимого поля (например, «Номер телефона»)	Да
title-id	Идентификатор текстовки для заголовка (титла) поля. Если для текущего выбранного языка текстовка в ресурсном файле отсутствует — используется текстовка из атрибута title	Нет
message	Информационное сообщение (подсказка) для вводимого поля	Нет
message-id	Идентификатор текстовки для сообщения (подсказки). Если для текущего выбранного языка текстовка в ресурсном файле отсутствует — используется текстовка из атрибута message	Нет
on-top	Позволяет редактировать значение поля ввода во всплывающем окне	Нет
group-id	Значение, по которому происходит группировка полей на экране, то есть поля, для которых указано одно и то же значение group-id , визуально будут объединены на экране в один блок. Поддерживается некоторыми интерфейсами	Нет
flags	Флаги, дополнительные свойства данного поля. Поддерживаемые флаги приведены в разделе 5.4.2	Нет
width	Ширина данного поля в текущей строке в процентах, если используется конструкция <code><row></code> . Если сумма значений width всех полей внутри <code><row></code> превышает 100, то возникнет ошибка при обработке сценария. Если сумма значений width всех полей внутри <code><row></code> превышает 100, но width указан	Нет

Атрибут	Описание	Обяз.
	не для всех полей, то возникнет ошибка при обработке сценария. Если значение не указано, то считается, что такое поле занимает все свободное место. Если таких полей несколько, то они делят все свободное место поровну. Поддержка атрибута зависит от интерфейса. В текущей версии реализовано только для «Платформы электронных кошельков Smartkeeper».	
keyboard	Указывается тип клавиатуры, отображаемой на экране терминала. Типы клавиатур описаны в разделе 5.3	Да
max-len	Максимальное число вводимых символов, задает ограничение на длину поля. Максимальная длина текста рассчитывается с учётом префикса и постфикса	Да
prefix	Используется для указания префикса вводимых данных. Префикс не редактируется. Введённые данные будут переданы на сервер вместе с префиксом	Нет
postfix	Используется для указания постфикса вводимых данных, не редактируется, добавляется к введённому значению, передается на сервер	Нет
read-only	Флаг «только чтение». Необязательный атрибут. Принимает одно из двух значений: <ul style="list-style-type: none">• true — только чтение, поле для ввода неактивно;• false — поле для ввода активно, используется по умолчанию	Нет
default	Используется для указания значения, выводимого в поле ввода по умолчанию, можно редактировать	Нет
example	Пример ввода информации, отображается на экране	Нет
xbutton	Указывается один символ из символьной или	Нет

Атрибут	Описание	Обяз.
	буквенной раскладки клавиатуры, который будет отображаться вместо функциональной кнопки «Стереть» (или сброс введенного значения) для цифровой раскладки. Данный атрибут используется только с цифровой раскладкой. При указании клавиатуры в 5 версии ТПО важен регистр (digital — неверно)	
secure	Используется для возможности безопасного ввода данных, определены значения: <ul style="list-style-type: none">• true — в поле ввода текстовой информации символы при вводе подменяются знаком «*». Например, при вводе пароля;• false — в поле ввода текстовой информации при вводе символы не подменяются знаком «*»	Нет
fail-regex	Содержит регулярное выражение для проверки. Используется в тех случаях, когда нельзя гарантировать, что элемент <code><verify></code> валидирует все допустимые значения. Если введенные данные не удовлетворяют <code><verify></code> , но удовлетворяют fail-regex , то к атрибутам платежа добавляется элемент Key.INVALID=«true» . Если данный атрибут присутствует, то генерируется событие с типом «Неизвестный номер» (раздел кабинета «Мониторинг — События»), а в pay.log появляется запись «Invalid input data. Ask about payment». Для клиента процедура оплаты предполагает вывод диалогового окна с запросом подтверждения оплаты. Доступен только в ТПО 7 версии	Нет
trailing-widget	Добавление на экран специальных кнопок ввода данных. Например «Камера», «Список контактов». Поддержка функционала обеспечивается в рамках дизайна GUI.	Нет

Пример:

```
<text-field name="id1" keyboard="Digital" max-len="11"
  message="Введите номер лицевого счета"
  title="Номер лицевого счета" default="24546784586"
  xbutton="/">
  <!--Регулярное выражение для валидации вводимого номера-->
  <verify> <rule regex="^\d{6,11}$"/> </verify>
</text-field>
```

В данном примере символ «X» на клавиатуре заменен на косую черту, установлено значение по умолчанию в поле ввода. Экран, соответствующий данному примеру, приведен на рисунке 6.1.1.3.

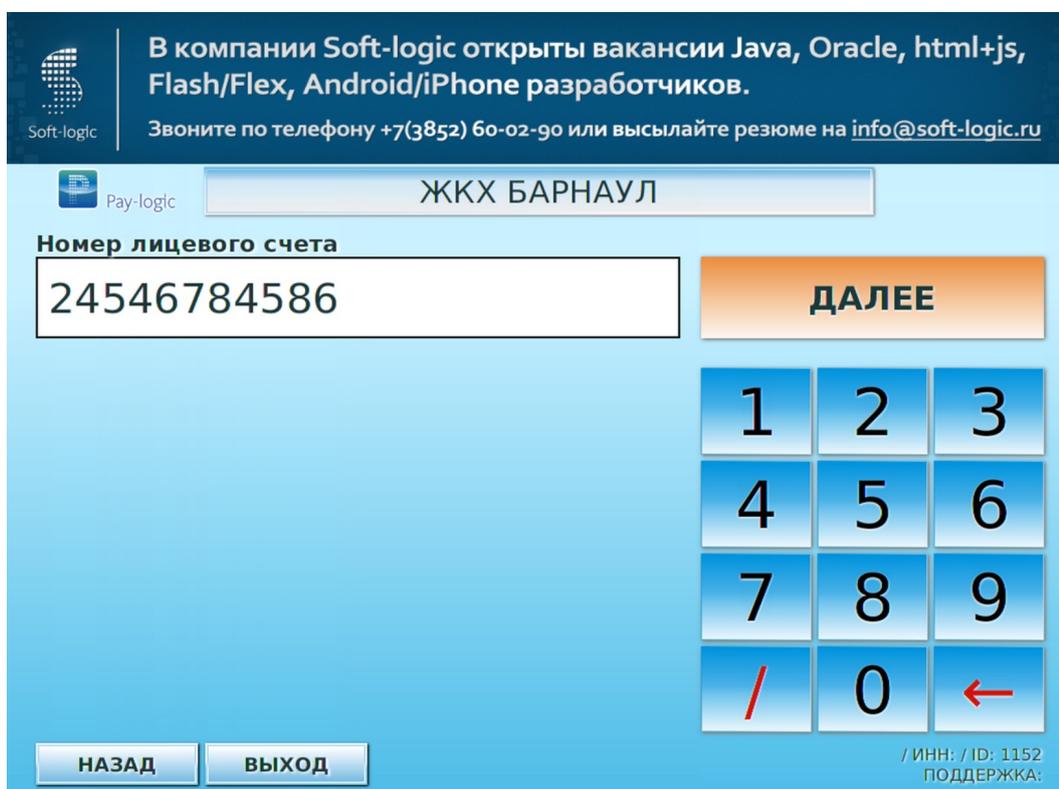


Рисунок 6.1.1.3 — Текстовое поле ввода с цифровой клавиатурой

Пример использования текстового поля ввода с `fail-regex`:

В рассматриваемом примере валидный номер лицевого счета содержит 10 цифр, в атрибуте `fail-regex` указано, что по номерам, состоящим из 8 цифр, так же необходимо принимать оплату.

```
<forms>
  <form service-id="1152">
    <!--Создание поля ввода, со следующими параметрами: активна цифровая
    клавиатура, максимальное количество вводимых символов 10. Название поля
    ввода: "Лицевой счет"-->
```

```
<text-field name="id1" keyboard="Digital" max-len="10"
  message="Введите номер лицевого счета"
  title="Лицевой счет" fail-regex="^\d{8}$">
<!--Подсказка, отображается на экране-->
<help> Введите номер лицевого счета </help>
  <!--Регулярное выражение для валидации вводимого номера-->
  <verify>
    <rule regex="^\d{10}$"/>
  </verify>
  <!--Маска для форматирования вводимого номера, применяется по
умолчанию-->
    <split default="**** * * *"/>
  </text-field>
</form>
</forms>
```

Процесс оплаты по сценарию приведен на рисунке 6.1.1.4.

Рисунок 6.1.1.4 — Форма оплаты с **fail-regex**

Когда клиент вводит номер лицевого счета, состоящий из 8 цифр, и нажимает кнопку «Далее», то появляется диалоговое окно с запросом подтверждения правильности ввода данных. Если клиент нажимает кнопку «Подтвердить», то он переходит к экрану оплаты.

В разделе «Мониторинг — События» отображается событие с типом «Неизвестный номер» (рисунок 6.1.1.5).

Фильтр Настройки

Дата начала: 11.12.2015 00:00 Дата окончания: 12.12.2015 00:00 Период: Не задано

Агент: Город: Комментарий:

Тип события: Не задано Точка: 768 - Терминал 768 Сервисная группа: Показывать события: Все

Версия ПО: Группировать по точкам:

События

	Время события	Время сервера	Статус	Тип события	Агент	Сервисная группа	Точка	Версия ПО	Комментарий
	11.12.2015 15:53:40	11.12.2015 15:53:40	Обработано	Предупреждение на автомате	"Soft-logic"	Группа Техник	Терминал 768	7.10.3/86	Error on process template: render3, param=\$operation.service.provider.ogrn line: 36, col: 13
	11.12.2015 15:53:40	11.12.2015 15:53:40	Обработано	Предупреждение на автомате	"Soft-logic"	Группа Техник	Терминал 768	7.10.3/86	Error on process template: render3, param=\$operation.service.provider.bank line: 31, col: 14
	11.12.2015 15:53:40	11.12.2015 15:53:40	Обработано	Предупреждение на автомате	"Soft-logic"	Группа Техник	Терминал 768	7.10.3/86	Error on process template: render3, param=\$operation.service.provider.ks line: 25, col: 14
	11.12.2015 15:53:40	11.12.2015 15:53:40	Обработано	Предупреждение на автомате	"Soft-logic"	Группа Техник	Терминал 768	7.10.3/86	Error on process template: render3, param=\$operation.service.provider.account line: 22, col: 6
	11.12.2015 15:53:40	11.12.2015 15:53:40	Обработано	Предупреждение на автомате	"Soft-logic"	Группа Техник	Терминал 768	7.10.3/86	Error on process template: render3, param=\$operation.service.provider.bik line: 17, col: 14
	11.12.2015 15:53:39	11.12.2015 15:53:40	Обработано	Ошибка автомата	"Soft-logic"	Группа Техник	Терминал 768	7.10.3/86	Specific collection template is not found: render3
	11.12.2015 15:53:29	11.12.2015 15:53:30	Обработано	Неизвестный номер	"Soft-logic"	Группа Техник	Терминал 768	7.10.3/86	Оплата услуг Барнаул: [InputElement(key=id1, keyTitle=Лицевой счет, keyTitleId=null, originalValue=22457896, value=22457896, valueTitle=2245 789 6 , valueTitleId=null, flags=0, alternatives=[])]

Рисунок 6.1.1.5 — «Мониторинг — События»

6.1.2 ВАЛИДАЦИЯ ДАННЫХ (<VERIFY>)

Правила проверки задают ограничение на вводимое пользователем значение текстового или цифрового выражения. Если введенные данные удовлетворяют параметрам валидации, значение элемента считается прошедшим проверку, что влияет на соответствующее поведение формы (в том числе, возможность перехода к дальнейшим шагам — оплате).

Для проверки правильности введенных данных используется элемент валидатор, описываемый тегом `<verify>`.

Дочерние теги: зависят от типа валидатора.



Примечание!

Внешний валидатор — это валидатор, который расположен не в библиотеке `input-model.jar`, а какой либо другой. Так, например, для использования js-валидатора необходимо присутствие библиотеки `js.jar`

Элемент `<verify>` имеет один атрибут **type**, который и определяет функционал валидатора. Все доступные типы приведены в таблице 6.1.2.1.

Таблица 6.1.2.1 — Типы валидаторов

Тип	Описание	Пример
regex	Используется по умолчанию. Осуществляет проверку на соответствие регулярным выражениям. Дочерние элементы: <ul style="list-style-type: none"><code><rule></code> — в атрибуте regex описывает разрешающее правило;	<pre><verify type="regex"> <rule regex="^12.*\$"/> <block regex="^.*7\$"/> </verify></pre>

Тип	Описание	Пример
	<ul style="list-style-type: none"> <code><block></code> — в атрибуте regex описывает запрещающее правило. Данные будут валидны, если они удовлетворяют хотя бы одному разрешающему правилу и при этом не удовлетворяют всем запрещающим правилам 	
range	Осуществляется проверка на вхождение в заданный интервал. Недоступно для <code><text-field></code>	<pre><verify type="range"> <range begin="0" end="9999999.99"/> </verify></pre>
cbc	Валидатор номеров карт сдачи. Возвращает true , когда введен верный номер карты сдачи. Поддерживаются 16-значные пин-коды	<pre><verify type="cbc"/></pre>
voucher	Валидатор ваучеров. Возвращает true , когда введен 12-ти значный пин	<pre><verify type="voucher"/></pre>
or	Логический валидатор, возвращает true , если хотя бы один вложенный валидатор вернул true	<pre><verify type="or"> <verify type="regex"> <rule regex="^2.*"/> </verify> <verify type="regex"> <rule regex="^1.*"/> </verify> </verify></pre>
and	Логический валидатор, возвращает true , если все вложенные валидаторы вернули true	<pre><verify type="and"> <verify type="regex"> <rule regex="^2.*"/> </verify> <verify type="regex"></pre>

Тип	Описание	Пример
		<pre><rule regex="^1.*"/> /verify> </verify></pre>
luhn	Используется для контроля правильности ввода PAN (номера пластиковой карты)	<pre><verify type="luhn"/></pre>
yandex	Валидатор номера кошелька Яндекс.деньги. В 7 версии ТПО не поддерживается	<pre><verify type="yandex"/></pre>
custom	Любой внешний валидатор, тип внешнего валидатора определяется атрибутом name	<pre><verify type="custom" name="имя валидатора"/></pre>
js	Валидатор, позволяющий использовать для проверки данных JavaScript	<pre><verify type="js"> <![CDATA[код на JavaScript]]> </verify></pre>
empty	Валидатор, позволяющий либо оставлять поле пустым, либо проверять его на соответствие вложенному валидатору	<pre><verify type="or"> <verify type="empty"/> <verify type="luhn"/> </verify></pre>
mod11	Валидатор по модулю 11	<pre><verify type="mod11"/></pre>
ibaniso7064	Валидатор номеров банковских счетов IBAN (<i>International Bank Account Number</i>), реализованный в соответствии со стандартом ISO 7064	<pre><verify type="ibaniso7064"/></pre>

В зависимости от типа доступны различные дочерние теги и атрибуты.

Дочерние элементы `<verify>` с `type="regex"`: `<rule>` — определяет правильность ввода, `<block>` — определяет невалидный ввод.

Например, запись:

```
<rule regex="1.*"/>
```

говорит о том, что номер должен начинаться с «1», но

```
<block regex="1.*"/>
```

запрещает вводить все строки, которые начинаются с символа «1».

Валидатор с типом **regex** использует язык регулярных выражений. Используется для локальной проверки введенных данных.

При объявлении используют элементы `<rule>`, `<block>`.

Элемент `<verify>` с типом **regex** для проверки введенного значения с помощью регулярного выражения:

```
<verify type="regex">  
  <rule regex=<string>/>  
</verify>
```

Правило задает условие на введенное значение.

Дочерние элементы `<verify>` с `type="range"`: `<rule>`.

Атрибуты элемента `<verify>` с `type="range"` приведены в таблице 6.1.2.2.

Таблица 6.1.2.2 — Атрибуты валидатора с типом **range**

Атрибут	Описание	Обяз.
begin	Начальное значение для проверки вводимого	Да
end	Конечное значение для проверки вводимого	Да

Пример использования валидатора с `type="js"`:

```
<verify name="js" type="js" >
  <![CDATA[
    !!!Валидатор!!!
  ]]>
</verify>
```

Примеры валидаторов:

В следующем примере определяется функция, внутри которой сначала происходит проверка длины `number` — это вводимое в поле значение, в самом поле атрибут может называться как угодно `id1/number/phone` и другие, но в функции вводимый номер ВСЕГДА `number`. Если длина `number` равна 13, то вычисляется значение `m` путем сложения произведения каждого символа номера и заданного числа. Затем определяется значение переменной `c` — остаток от деления `m` на 11. Переменной `k` присваивается разность между 11 и `c`. Если `k` равняется 11 или 10, то присваивается значение 0. Затем возвращается `k`, равное первым 12 символам `number`. Если обработка скрипта завершается успешно (не `false`), то номер считается валидным.

```
function f(number) {
  if (number.length == 13) {
    var m = number[0] * 7 + number[1] * 6 + number[2] *
      5 + number[3] * 4 + number[4] * 3 + number[5] * 2 +
      number[6] * 7 + number[7] * 6 + number[8] * 5 +
      number[9] * 4 + number[10] * 3 + number[11] * 2;
    var c = m % 11;
    var k = 11 - c;
    if (k == 11 || k == 10) {
      k = 0;
    }
    return k == number[12];
  }
  return false;
}
f(number);
```

Валидные номера: 2709978450144,2204991445007.

```
function f(number) {
  if (number.length != 22) {
```

```
        return false;
    }

    var result = 0;
    var count = 2;
    for (j = 20; j >= 0; j--) {
        result += count * number[j];
        count += 1;
        if (count > 7) {
            count = 2;
        }
    }

    step2 = result % 11;
    var check = number[21];
    if (step2 == 0 && step2 == check) {
        return true;
    }

    step2 = 11 - step2;
    if (step2 > 9 && step2 % 10 == check) {
        return true;
    }

    return step2 == number[21];
}
f(number)
```

Валидные номера: 9100108510009616090595, 9100111510003038743850.

6.1.3 ФОРМАТИРОВАНИЕ ДАННЫХ (<SPLIT>)

Элемент `<split>` используется для форматирования введенных данных. Позволяет задавать правила разбивки текста или числа на экране ввода для удобства восприятия. Правило разбивки не может быть изменено пользователем при вводе. Алгоритм работы предполагает сначала обход всех правил разбивки. Если будет найдено правило, которое подходит для введенных данных, то будет выбран вариант разбивки из данного правила. Если ни одно правило не подошло, то используется разбивка по умолчанию из атрибута **default**. Наличие секции не является обязательным. Структура элемента:

```
<split default=<string>
  by-step=[true|false]>
  <rule regex=<string> value=<string>/>
</split>
```

Дочерние элементы: `<rule>`.

Описание атрибутов приведено в таблице 6.1.3.1.

Таблица 6.1.3.1 — Описание атрибутов элемента `<split>`

Атрибут	Описание	Обяз.
default	Задаёт правило разбивки вводимых данных по умолчанию	Нет
by-step	Указывает, что маска в текстовом поле будет отображаться по мере ввода символов	Нет

Элемент `<rule>` определяет правило разбивки введенных данных. В случае, если введённое значение соответствует регулярному выражению, заданному в атрибуте **regex**, то применяется правило разбивки, указанное в **value**.

Дочерние теги: отсутствуют.

Описание атрибутов элемента `<rule>` приведено в таблице 6.1.3.2.

Таблица 6.1.3.2 — Описание атрибутов элемента `<rule>`

Атрибут	Описание	Обяз.
<code>regex</code>	Регулярное выражение, на соответствие которому осуществляется проверка	Да
<code>value</code>	Правило разбивки данных	Да

Пример:

В примере ниже задано форматирование введенного значения из 6 символов по правилу — разбивка символов по три через один дефис. В случае, если длина введенного выражения от 7 до 12 символов, то применяется другое правило разбивки — разбивка символов по три с дефисами между ними. Максимально возможно ввести 12 символов.

```
<split default="***-***">  
  <rule regex="^[0-9]{7,12}$" value="***-***-***-***"/>  
</split>
```

В примере ниже задано форматирование введенного значения из 6 символов по правилу — разбивка символов по три через один дефис.

Пример:

```
<split default="***-***"/>
```

В примере ниже, если не введен ни один цифровой символ, форматирования фактически не осуществляется. Если введен хотя бы один цифровой символ, то по мере ввода символов выполняется форматирование в соответствии с правилом `8(***)**_**_**`.

Пример:

```
<split by-step="true">  
  <rule regex="^\d{0}" value="8*****"/>  
  <rule regex="^\d{1,}" value="8 (**) ***-**-**"/>  
</split>
```

**Внимание!**

Программное обеспечение «Платформа электронных кошельков SmartKeeper» не поддерживает возможность указания нескольких масок, т. е. вложенных элементов `<rule>`.

6.1.4 ЗАМЕНА ЗНАЧЕНИЯ (<REPLACES>)

Для замены вводимого клиентом значения используется элемент `<replaces>`. Он позволяет разбивать вводимое значение на группы для последующей модификации (добавление произвольных символов, удаления части значения, перестановка групп). Внутри данного тега описываются правила модифицирования `<rule>`, количество правил не ограничено. Структура элемента:

```
<replaces>
  <rule regex=<notEmptyString>
    value=<string>/>
</replaces>
```

Дочерние элементы: `<rule>`.

Атрибуты: отсутствуют.

Описание атрибутов элемента `<rule>` приведено в таблице 6.1.4.1.

Таблица 6.1.4.1 — Описание атрибутов элемента `<rule>`

Атрибут	Описание	Обяз.
regex	Регулярное выражение, на соответствие которому осуществляется проверка	Да
target	Правила для разбития значения на группы путем указания ссылок на эти группы. Разбивка происходит с помощью круглых скобок, каждая пара скобок считается ссылкой. В случае, если отсутствует, то вместо него будет использоваться regex	Нет
value	Указывается ссылка на номер группы при отображении данных. Выражение value может	Да

Атрибут	Описание	Обяз.
	содержать любые символы и правила форматирования	

Пример:

```
<!--Создание поля ввода, со следующими параметрами: активна цифровая
клавиатура, максимальное количество вводимых символов 6. Название поля
ввода: "Период платежа"-->
<text-field name="period"
  keyboard="Digital"
  max-len="6"
  message="Введите период платежа"
  title="Период платежа">
  <!--Подсказка, отображается на экране-->
  <help>
  <![CDATA[<html>Пример 01-2011<br>По окончании ввода нажмите
    клавишу <b>"Далее"</b>]]>
  </help>
  <!--Регулярное выражение для валидации вводимых данных-->
  <verify>
  <rule regex="^(0[1-9]|1[0-2]) (201[0-5])$"/>
  </verify>
  <replaces>
  <rule regex="\d{6}$" target="^(0[1-9]|1[0-2]) (201[0-5])$"
    value="$1-$2"/>
  </replaces>
  <!--Маска для форматирования вводимого номера, применяется по
умолчанию-->
  <split default="**-****"/>
</text-field>
```

В данном примере в атрибут платежа **period** записывается период платежа, состоящий из 6 цифр. Введенное выражение форматируется и на экране отображается: 2 цифры через дефис 4 цифры, для него задано правило валидации первые две цифры периода соответствуют месяцу ("(0[1-9]|1[0-2])"), следующие четыре — соответствуют году ("(201[0-5])"). В теге `<replaces>` в атрибуте **target** задается правило разбивки введенного выражения для передачи этого атрибута. Атрибут **period** будет передаваться не как строка из 6 цифр, а как месяц — первая группа (\$1), указанная в атрибуте **value**, и год — вторая группа (\$2). Экран, соответствующий примеру, приведен на рисунке 6.1.4.1.

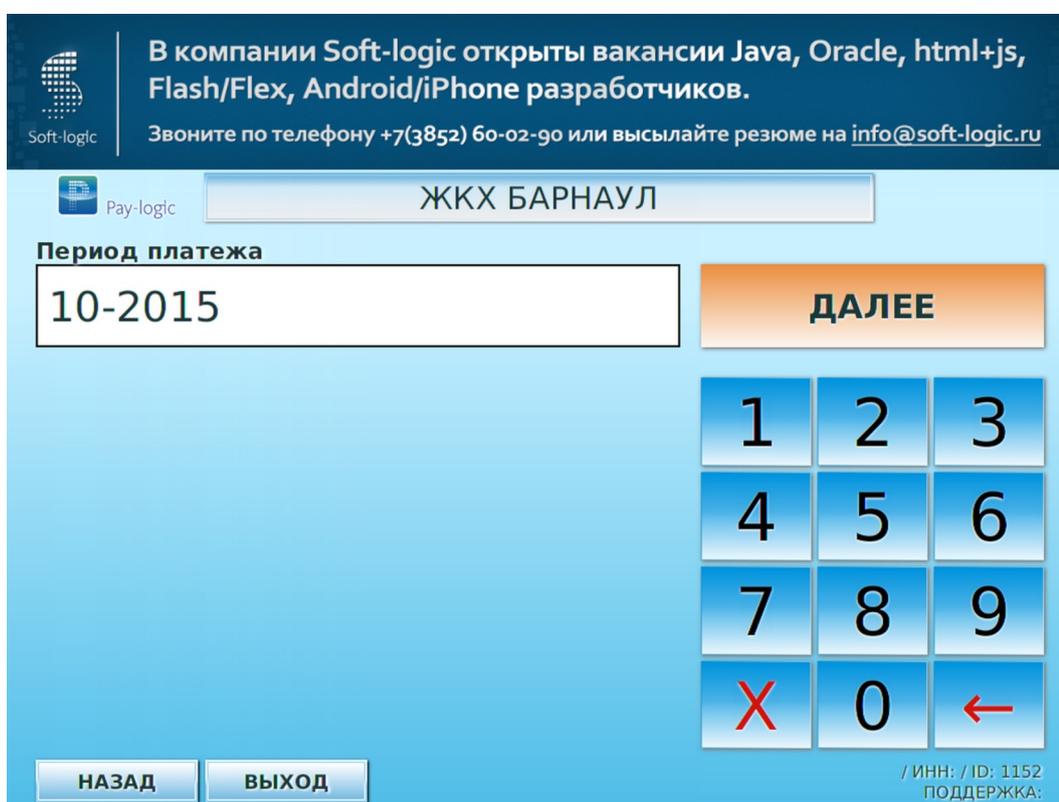


Рисунок 6.1.4.1 — Разбивка значения атрибута по правилу, описанному в теге `<replaces>`

6.1.5 ФИЛЬТРАЦИЯ ДАННЫХ (<FILTER>)

Определить какие символы допустимы для ввода в какое-либо поле возможно с использованием правил фильтраций. Для указания правил фильтрации данных используется элемент `<filter>`. Для элемента определены два варианта фильтрации: по маске и по правилам. В случае ввода символа, не подходящего под правила, описанные внутри `<filter>`, символ не вводится. ТПО игнорирует нажатия.

Структура тега:

1 вариант:

```
<filter type="mask" mask=<notEmptyString>/>
```

2 вариант:

```
<filter type="char-permission">  
  <allow pos=<integer> value=<notEmptyString>/>  
</filter>
```

При использовании первого варианта фильтрация осуществляется согласно маске, указанной в атрибуте **mask**. Подробное описание правил составления масок приведено по адресу:

<http://docs.oracle.com/javase/7/docs/api/javax/swing/text/MaskFormatter.html>.

Атрибут **mask** описывает маску ввода. Символ «d» в маске означает, что в данной позиции разрешены только цифры, «с» — буквы, «*» — любые символы.

При использовании второго варианта фильтрация осуществляется по специальным правилам, определяемым в дочерних элементах.

Дочерние элементы: `<allow>`, `<deny>` — определяют разрешающее/запрещающее правила фильтрации.


```
<filter type="mask" mask="cdddddddddddddddddddddddddd"/>  
</text-field>
```

Экран, соответствующий данному примеру, приведен на рисунке 6.1.5.1.

В примере переход к следующему экрану может быть осуществлен только при вводе выражения длиной до 26 символов, первый из которых «a» или «b», последующие — цифры.

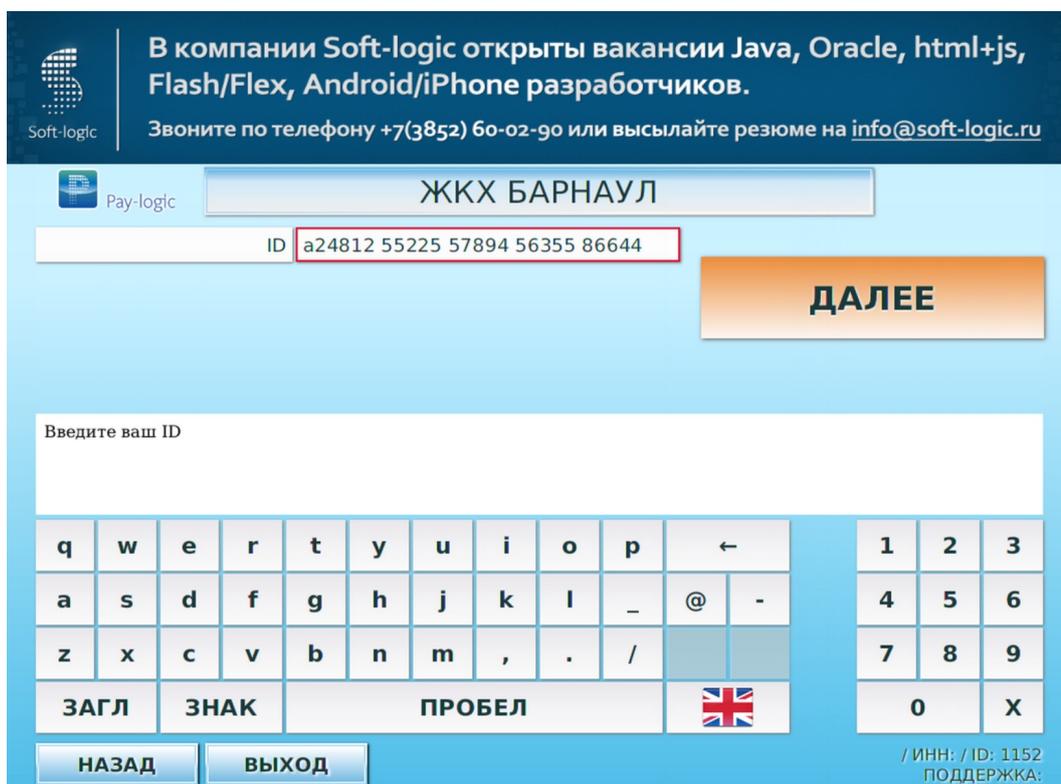


Рисунок 6.1.5.1 — Ввод данных, соответствующих правилу фильтрации

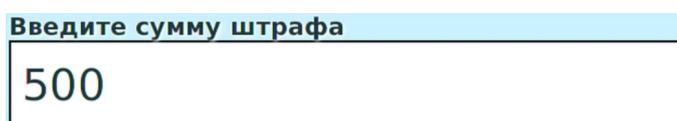
В примере, представленном ниже, номер не может начинаться с 8 или 7.

Пример:

```
<!--Создание поля ввода, со следующими параметрами: активна цифровая
клавиатура, максимальное количество вводимых клиентом символов 10.
Название поля ввода: "Номер телефона"-->
<text-field name="idl" keyboard="Digital" max-len="10"
  title="НОМЕР ТЕЛЕФОНА">
  <!--Регулярное выражение для валидации вводимого номера-->
  <verify> <rule regex="^9\d{9}$" /> </verify>
  <!--Подсказка, отображается на экране-->
  <help>
  Введите номер телефона (10 цифр) БЕЗ ведущей 8.
  </help>
  <filter type="char-permission">
  <deny pos="0" value="^8|7$" />
  </filter>
  <!--Маска для форматирования вводимого номера, применяется по
умолчанию-->
  <split default="8*****" by-step="true">
  <rule regex="^\d{1,}" value="8(**)***-**-***" />
  </split>
</text-field>
```

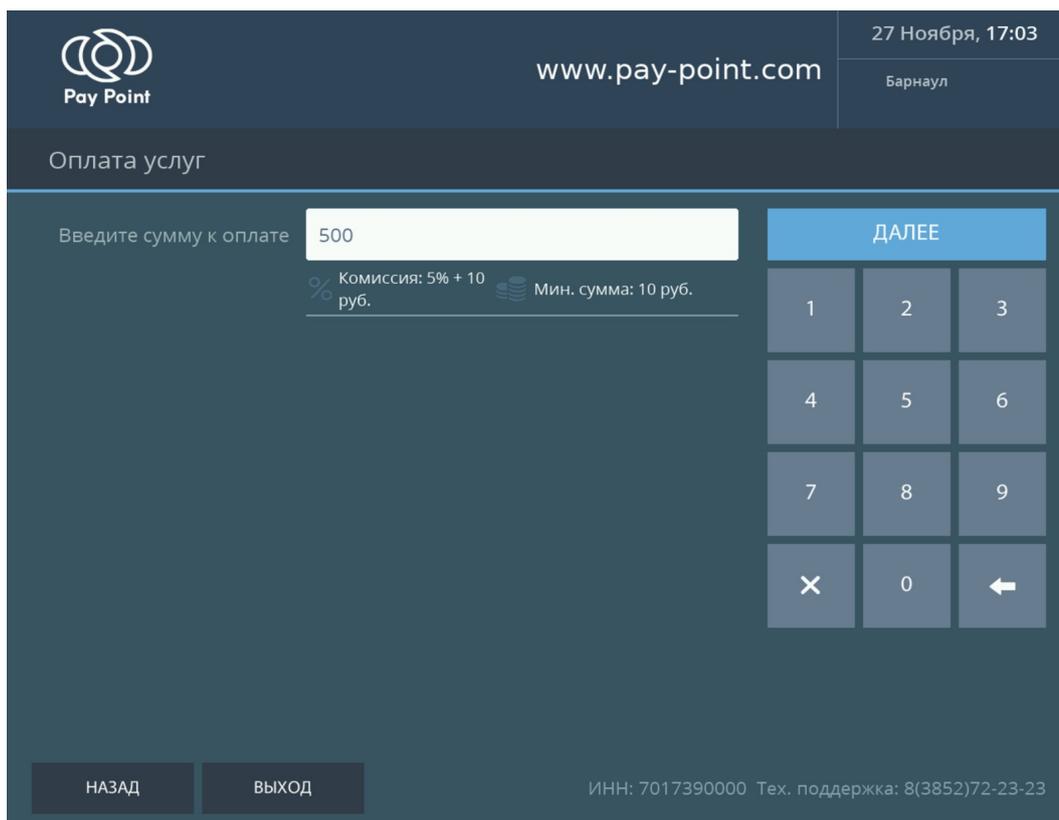
6.2 ПОЛЕ ВВОДА ЧИСЛОВОГО ЗНАЧЕНИЯ (<NUMERIC-FIELD>)

Поле ввода числового значения позволяет отображать цифровую информацию в форматированном виде. Задается тегом <numeric-field>. Внешний вид приведен на рисунках 6.2.1, 6.2.2.



Введите сумму штрафа

Рисунок 6.2.1 — Пример элемента <numeric-field> для интерфейса «Blues»



Pay Point www.pay-point.com 27 Ноября, 17:03
Барнаул

Оплата услуг

Введите сумму к оплате 500

ДАЛЕЕ

Комиссия: 5% + 10 руб. Мин. сумма: 10 руб.

1	2	3
4	5	6
7	8	9
×	0	←

НАЗАД ВЫХОД

ИНН: 7017390000 Тех. поддержка: 8(3852)72-23-23

Рисунок 6.2.2 — Пример элемента <numeric-field> для интерфейса «Smoke»

Возможности поля ввода:

1. Позволяет вводить целые положительные и отрицательные числа.
2. Позволяет задать диапазон допустимых значений для вводимых данных.
3. Позволяет задать сумму оплаты по сервису с возможностью изменения суммы на экране оплаты.

Структура поля ввода числового значения:

```
<numeric-field decor=<notEmptyString>
  name=<string>
  title=<titleType>
  title-id=<resourceIdType>
  message=<messageType>
  message-id=<resourceIdType>
  on-top=<notEmptyString>
  group-id=<nonNegativeInteger>
  flags=<flags_specification>
  min=<decimal>
  max=<decimal>
  change-sumexchange=[true|false]
  change-sumpurchase=[true|false]
  default=<decimal>
  allow-decimal=[true|false]
  width=<nonNegativeInteger>
  rowId=<nonNegativeInteger>
  updown=<decimal>>
  <help> ... </help>
  <barcode-scanner> ... </barcode-scanner>
</numeric-field>
```

Дочерние элементы: <verify> (раздел [6.1.2](#)), <help> (раздел [7.1](#)), <barcode-scanner> (раздел [7.2](#)).

Описание атрибутов приведено в таблице 6.2.1.

Таблица 6.2.1 — Атрибуты поля ввода числового значения

Атрибут	Описание	Обяз.
name	Код атрибута платежа (идентификатор поля)	Да
title	Заголовок поля ввода. Предназначен для вывода пользователю названия вводимого поля (например, «Номер телефона»)	Да
title-id	Идентификатор текстовки для заголовка (титла) поля. Если для текущего выбранного языка текстовка в ресурсном файле отсутствует — используется текстовка из атрибута title	Нет
message	Информационное сообщение (подсказка) для вводимого поля	Нет
message-id	Идентификатор текстовки для сообщения (подсказки). Если для текущего выбранного языка текстовка в ресурсном файле отсутствует — используется текстовка из атрибута message	Нет
on-top	Позволяет редактировать значение поля ввода во всплывающем окне	Нет
group-id	Значение, по которому происходит группировка полей на экране, то есть поля, для которых указано одно и то же значение group-id , визуально будут объединены на экране в один блок. Поддерживается некоторыми интерфейсами	Нет
flags	Флаги, дополнительные свойства данного поля. Поддерживаемые флаги приведены в разделе 5.4.2	Нет
min	Используется для указания минимального значения для экрана ввода числового значения. Допустимо указывать отрицательное число. В качестве значения данного атрибута возможно указать минимальную сумму для ввода при использовании атрибута force-external-range	Да
max	Используется для указания максимального	Да

	значения для экрана ввода числового значения. В качестве значения данного атрибута возможно указать максимальную сумму для ввода при использовании атрибута force-external-range	
change-sumpurchase	Используется для указания того, чтобы введенное значение использовалось в качестве суммы платежа. Возможные значения: <ul style="list-style-type: none">• true — указывает, что введенное числовое значение будет интерпретировано как сумма платежа;• false — означает, что поле не является полем ввода суммы платежа, то есть введенные в него данные представляют собой всего лишь какие-то данные, необходимые для совершения платежа. Значение по умолчанию. При установленном false кнопка «Изменить сумму к оплате» становится доступной только, если принятая сумма больше минимальной по сервису с учетом комиссии	Нет
change-sumexchange	Используется для указания того, что вводимая сумма является суммой к обмену. Используется на сервисах обменника для того, чтобы валидировать сумму, которую хочет обменять пользователь в зависимости от наличия купюр валюты, на которую осуществляется обмен	Нет
default	Используется для указания значения, выводимого в поле ввода по умолчанию, можно редактировать	Нет
allow-decimal	Используется для задания возможности ввода дробных чисел. Возможные значения: <ul style="list-style-type: none">• true — позволяет вводить дробные числа;• false — позволяет вводить только целые числа	Нет

width	<p>Ширина данного поля в текущей строке в процентах, если используется конструкция <code><row></code>. Если сумма значений width всех полей внутри <code><row></code> превышает 100, то возникнет ошибка при обработке сценария. Если сумма значений width всех полей внутри <code><row></code> превышает 100, но width указан не для всех полей, то возникнет ошибка при обработке сценария. Если значение не указано, то считается, что такое поле занимает все свободное место. Если таких полей несколько, то они делят все свободное место поровну. Поддержка атрибута зависит от интерфейса. В текущей версии реализовано только для «Платформы электронных кошельков Smartkeeper».</p>	Нет
rowId	<p>Идентификатор строки, в которой будет расположено поле. Поддержка атрибута зависит от интерфейса. В текущей версии реализовано только для «Платформы электронных кошельков Smartkeeper».</p>	Нет
updown	<p>Используется для возможности ввода с использованием счетчика. Поддерживается только в 7 версии ТПО на групповом экране</p>	Нет
force-external-range	<p>Позволяет принудительно устанавливать ограничение на ввод суммы. Возможные значения:</p> <ul style="list-style-type: none"> • true — возможно установить максимальное и минимальное значения суммы, используя атрибуты min, max; • false — ограничение на ввод суммы устанавливается извне. <p>Пример использования атрибута:</p> <pre data-bbox="710 1713 1372 1809"><numeric-field name="id1" message="Введите фиксированную сумму"</pre>	Нет

	<pre>secure="true" allow-decimal="true" min="50" max="15000" updown="2" change-sumpurchase="true" force-external-range="true"/></pre> <p>В данном примере будут установлено ограничение на вводимую сумму от 50 до 15000, которое задано в атрибутах min, max.</p>	
--	---	--

**Примечание!**

Если задан атрибут **change-sumpurchase** или **change-sumexchange**, то значения атрибутов **min**, **max** игнорируются. Для **change-sumpurchase** **min** станет минимальная сумма по сервису, **max** — сумма, которая с комиссией на себя не превысит установленный максимум для сервиса. Например, установлена минимальная сумма 10, максимальная — 500, минимальная и максимальная комиссия — 10%. В качестве суммы платежа максимально возможно указать 454,55, с учетом комиссии платеж составит 500. Минимальная сумма платежа — останется 10. С учетом комиссии платеж составит 11. Для **change-sumexchange** **min** и **max** — это минимальная и максимальная сумма, которая может быть обменена в зависимости от курса и наличия банкнот в диспенсере купюр

Пример:

```
<numeric-field name="id2"
  message="Введите сумму штрафа"
  allow-decimal="false"
  min="100"
  max="4000"
  change-sumpurchase="true"/>
```

В данном примере в атрибут платежа **id2** будет записано значение суммы штрафа. При вводе используется цифровая клавиатура. Подпись для поля ввода: «Введите сумму штрафа», введённая сумма будет являться суммой платежа (рисунок 6.2.3).

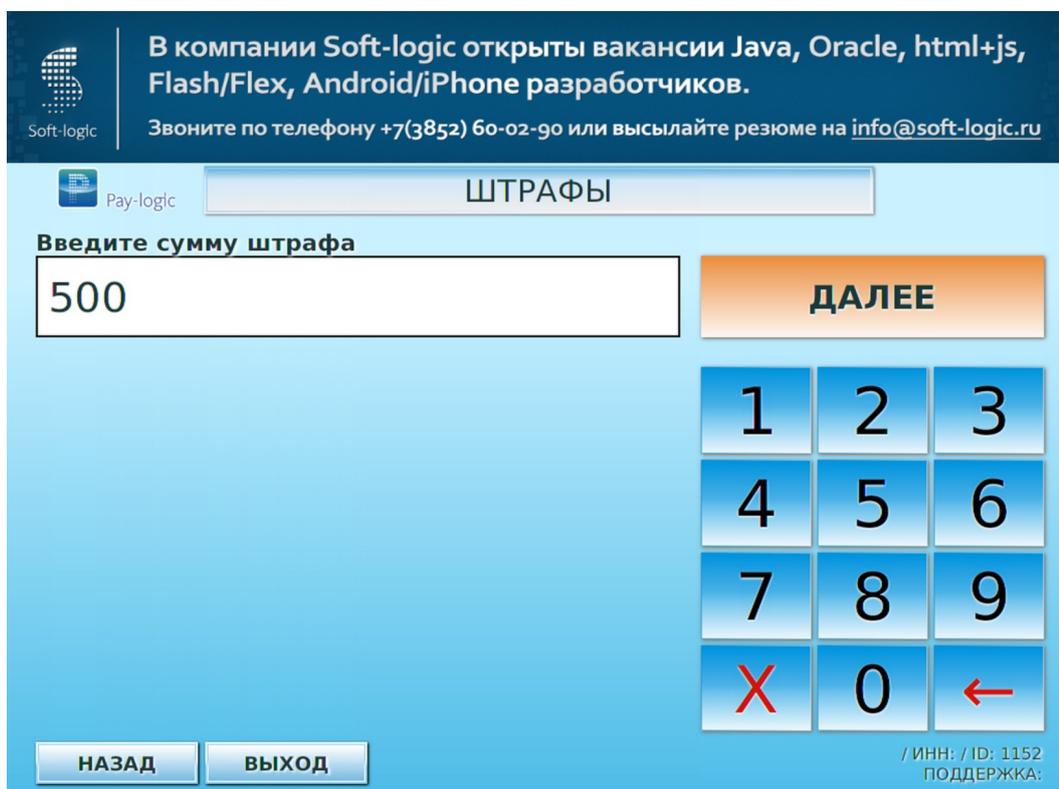


Рисунок 6.2.3 — Экран ввода числового значения по платежу

В 7 версии ТПО на групповом экране возможен ввод с использованием числового счетчика. Для этого используется атрибут **updown** — задает числовой счетчик. Отображает специальные клавиши справа и слева от вводимого значения — "+", "-". Позволяет корректировать значение в поле ввода на величину **updown** (увеличивать, уменьшать значение). При отсутствии данного атрибута специальные клавиши не отображаются. Для **updown="0"** — специальные клавиши не отображаются.

Пример:

```
<group-field>
  <numeric-field name="id1" message="Введите фиксированную сумму"
    secure="true" allow-decimal="true"
    min="50" max="15000" updown="2"
    change-sumpurchase="true"/>
  <date-field name="id4" title="Дата" format="dd.MM.yyyy"
    format-title="dd-MM-yyyy" from="03.03.2013">
```

```
</group-field> to ="03.03.2016" default="01.03.2015"/>
```

Экран, соответствующий данному примеру, приведен на рисунке 6.2.4.

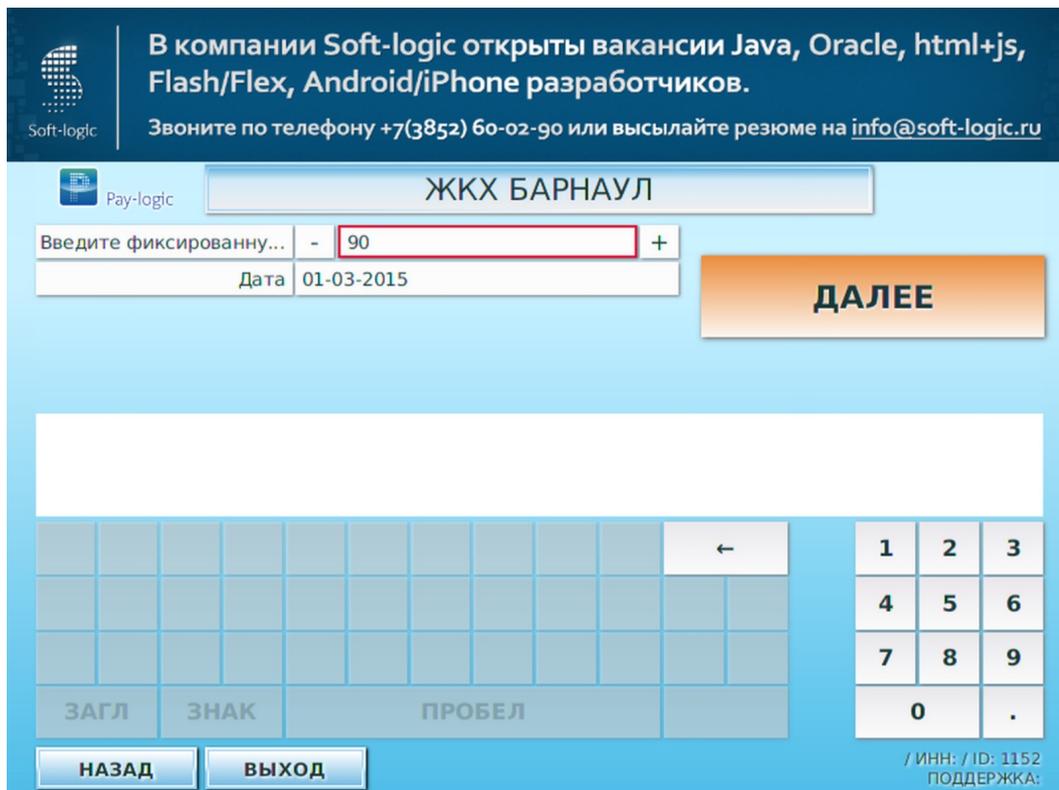


Рисунок 6.2.4 — Ввод с использованием счетчика

В 5 версии ТПО в качестве заголовка экрана отображается название сервиса, в 7 версии — значение атрибута **message** экрана.

Доступны следующие виды **decor** в стандартном интерфейсе: **cbs**, **popup**.

Пример:

```
<forms>  
<form service-id="1152">  
  <numeric-field decor="cbs" name="id3"
```

```
message="Введите пин-код карты сдачи" title="Пин-код"  
min="0" max="9999999999999999"/>  
</form>  
</forms>
```

Экран, соответствующий данному примеру, приведен на рисунке 6.2.5.

Пример:

```
<forms>  
<form service-id="1152">  
<numeric-field decor="popup" name="id3" message="Введите сумму платежа"  
title="Сумма" min="0" max="15000"/>  
</form>  
</forms>
```

Экран, соответствующий данному примеру, приведен на рисунке 6.2.6.

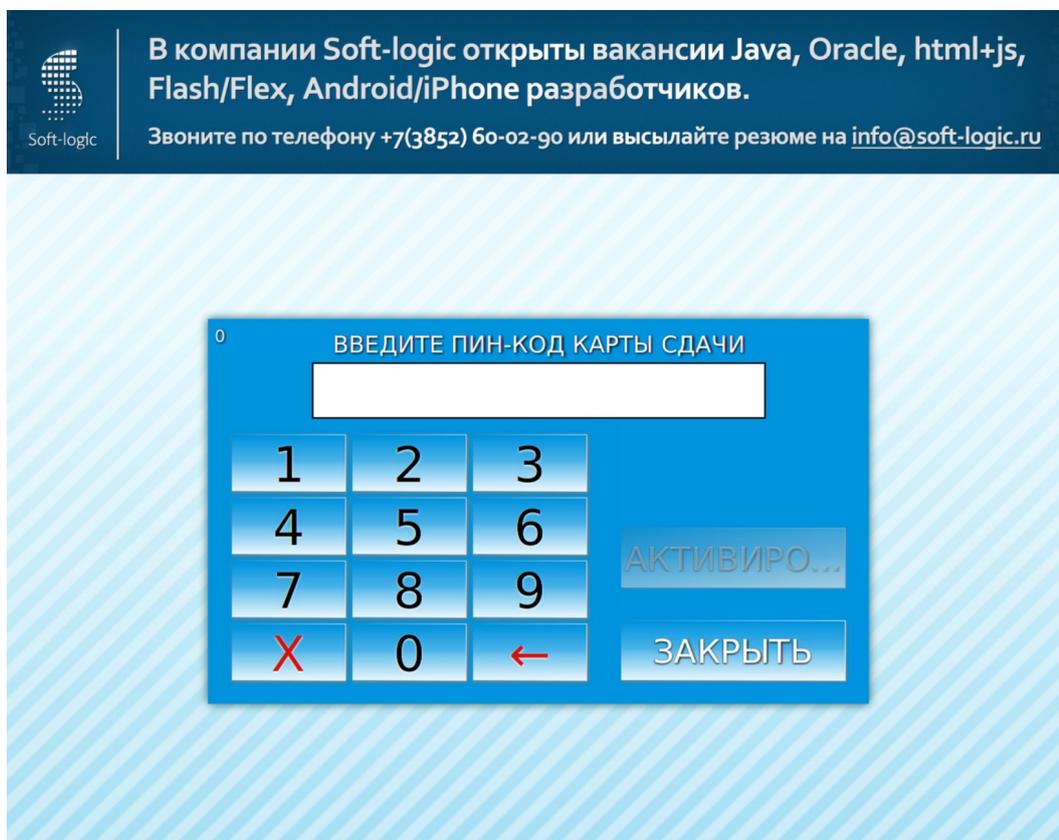


Рисунок 6.2.5 — Пример экрана `numeric-field` с `decor=«cbc»`

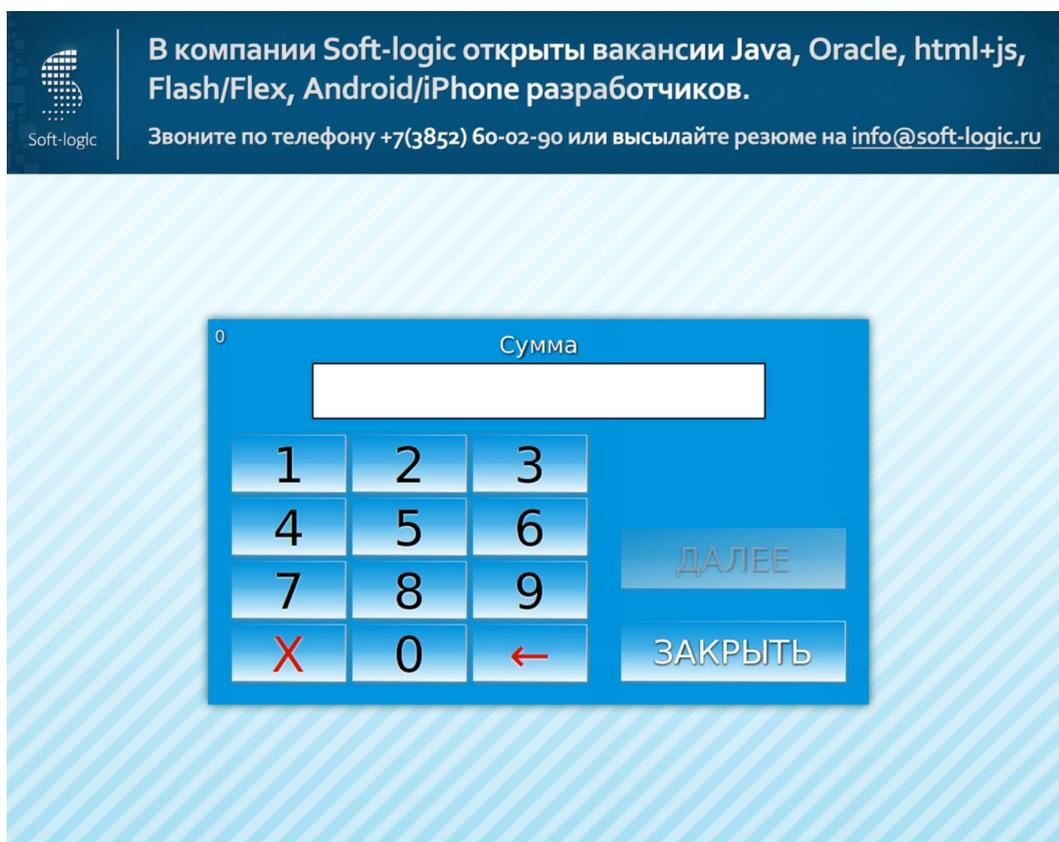


Рисунок 6.2.6 — Пример экрана `numeric-field` с `decor=«popup»`

6.3 ФЛАГ/ПЕРЕКЛЮЧАТЕЛЬ (<CHECKBOX-FIELD>)

Флаг/переключатель — элемент, позволяющий управлять параметром с двумя состояниями: включено/выключено. Задается тегом <checkbox-field>. Например, на экране может отображаться параметр «Согласен с условиями» с вариантами ответов «Да»/«Нет». Используется только на групповом экране. Внешний вид приведен на рисунке 6.3.1.



Рисунок 6.3.1 — Внешний вид элемента <checkbox-field>

Структура флага/переключателя:

```
<checkbox-field name=<string>
  title=<titleType>
  title-id=<resourceIdType>
  message=<messageType>
  message-id=<resourceIdType>
  on-top=<notEmptyString>
  group-id=<nonNegativeInteger>
  flags=<flags_specification>
  default=<notEmptyString40>
  read-only=[true|false]
  text=<notEmptyString40>
  message-off=<notEmptyString40>
  message-off-id=<notEmptyString40>
  required=[true|false]
  width=<nonNegativeInteger>
  rowId=<nonNegativeInteger>>
  <help> ... </help>
  <help-off> ... </help-off>
</checkbox-field>
```

Дочерние элементы: <help> (раздел [7.1](#)), <help-off> (используется для вывода подсказки при выключенном состоянии параметра, аналогичен <help>).

Описание атрибутов приведено в таблице 6.3.1.

Таблица 6.3.1 — Атрибуты элемента `<checkbox-field>`

Атрибут	Описание	Обяз.
name	Код атрибута платежа (идентификатор поля)	Да
title	Заголовок поля ввода. Предназначен для вывода пользователю названия вводимого поля (например, «Номер телефона»)	Да
title-id	Идентификатор текстовки для заголовка (титла) поля. Если для текущего выбранного языка текстовка в ресурсном файле отсутствует — используется текстовка из атрибута title	Нет
message	Информационное сообщение (подсказка) для вводимого поля	Нет
message-id	Идентификатор текстовки для сообщения (подсказки). Если для текущего выбранного языка текстовка в ресурсном файле отсутствует — используется текстовка из атрибута message	Нет
on-top	Позволяет редактировать значение поля ввода во всплывающем окне	Нет
group-id	Значение, по которому происходит группировка полей на экране, то есть поля, для которых указано одно и то же значение group-id , визуальнo будут объединены на экране в один блок. Поддерживается некоторыми интерфейсами	Нет
flags	Флаги, дополнительные свойства данного поля. Поддерживаемые флаги приведены в разделе 5.4.2	Нет
default	Используется для указания значения, выводимого в поле ввода по умолчанию, можно редактировать	Нет
read-only	Флаг «только чтение». Необязательный атрибут. Принимает одно из двух значений: <ul style="list-style-type: none">• true — только чтение, поле для ввода неактивно;	Нет

	<ul style="list-style-type: none"> false — поле для ввода активно, используется по умолчанию 	
text	Указывается наименование элемента списка. В 5 версии ТПО допустимо использовать теги языка разметки html для форматирования выводимого сообщения. Например, "<html>полной массой до 0.75 тонн включительно"	Нет
message-off	Используется для указания заголовка экрана ввода, когда значение чекбокса не выбрано	Нет
message-off-id	Идентификатор текстовки, которая будет использована в качестве сообщения, когда значение чекбокса не выбрано. Ресурсы с текстовками в ТПО 7 версии расположены в каталоге <i><корень ТПО>/res/module/i18n</i>	Нет
required	Используется для указания того, обязательным ли является поле <code><check-field></code> . Возможные значения: <ul style="list-style-type: none"> true — является обязательным; false — не является обязательным 	Нет
width	Ширина данного поля в текущей строке в процентах, если используется конструкция <code><row></code> . Если сумма значений width всех полей внутри <code><row></code> превышает 100, то возникнет ошибка при обработке сценария. Если сумма значений width всех полей внутри <code><row></code> превышает 100, но width указан не для всех полей, то возникнет ошибка при обработке сценария. Если значение не указано, то считается, что такое поле занимает все свободное место. Если таких полей несколько, то они делят все свободное место поровну. Поддержка атрибута зависит от интерфейса. В текущей версии реализовано только для «Платформы электронных кошельков Smartkeeper».	Нет
rowId	Идентификатор строки, в которой будет расположено	Нет

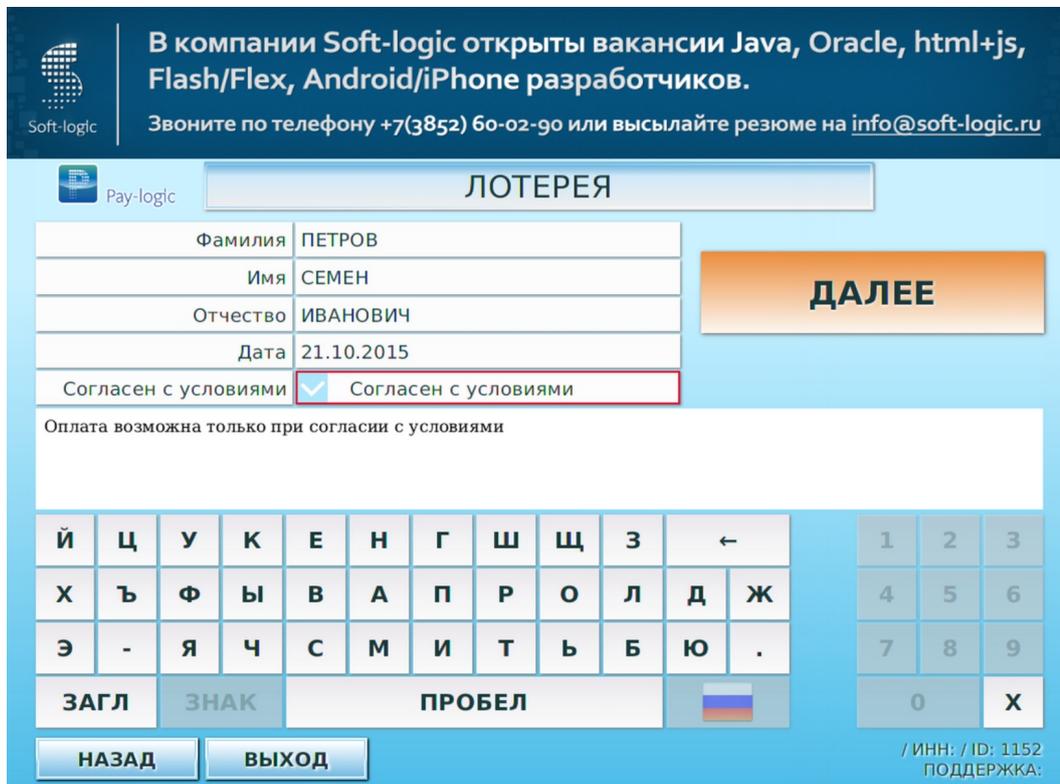
	поле. Поддержка атрибута зависит от интерфейса. В текущей версии реализовано только для «Платформы электронных кошельков Smartkeeper».	
--	---	--

Пример:

```
<checkbox-field name="idcheckbox" title="Согласен с условиями"
  text="Согласен с условиями" message="Согласен"
  message-off="Не согласен" validate="true">
  <help>
  <![CDATA[<html>Оплата возможна только при согласии с
    условиями]]>
  </help>
  <help-off>
  <![CDATA[<html>Для продолжения оплаты необходимо принять
    условия]]>
  </help-off>
</checkbox-field>
```

В данном примере элемент `<checkbox-field>` позволяет согласиться или не согласиться с условиями оплаты.

Экран, соответствующий данному примеру, приведен на рисунке 6.3.2.



Soft-logic

В компании Soft-logic открыты вакансии Java, Oracle, html+js, Flash/Flex, Android/iPhone разработчиков.
Звоните по телефону +7(3852) 60-02-90 или высылайте резюме на info@soft-logic.ru

Pay-logic ЛОТЕРЕЯ

Фамилия	ПЕТРОВ
Имя	СЕМЕН
Отчество	ИВАНОВИЧ
Дата	21.10.2015
Согласен с условиями	<input checked="" type="checkbox"/> Согласен с условиями

Оплата возможна только при согласии с условиями

Й	Ц	У	К	Е	Н	Г	Ш	Щ	З	←	
Х	Ъ	Ф	Ы	В	А	П	Р	О	Л	Д	Ж
Э	-	Я	Ч	С	М	И	Т	Ь	Б	Ю	.
ЗАГЛ	ЗНАК	ПРОБЕЛ									

НАЗАД ВЫХОД

1 2 3
4 5 6
7 8 9
0 X

/ ИНН: / ID: 1152
ПОДДЕРЖКА:

Рисунок 6.3.2 — Поле <checkbox-field>

6.4 СЕЛЕКТОРЫ

6.4.1 ОБЩИЕ ПОЛОЖЕНИЯ

Экраны-селекторы используются для выбора одного значения из списка значений. Информация на экране-селекторе представлена в виде списка элементов, являющихся кнопками с текстовой информацией или кнопками с логотипами. Экран-селектор часто используется для имитации вложенности меню, позволяет создавать группы элементов в меню. Для всех экранов реализован скроллинг списка.

Примером использования экрана-селектора может выступать выбор типа транспортного средства при оплате налога. Еще одним примером, иллюстрирующим использование экрана-селектора, является выбор тарифа за кабельное ТВ из списка, в котором каждый тариф имеет свой код, название и цену.

Также с использованием селекторов возможно осуществлять переход в другую форму оплаты с сохранением уже введенных атрибутов платежа (с использованием атрибута **redirect** тега `<item>` и атрибута **redirect-params**="..." в теге `<form>`, подробнее в разделе [6.4.2](#).

К экранам селекторам относятся текстовый селектор `<text-selector>` (раздел [6.4.2](#)), `<image-selector>` (раздел [6.4.3](#)). Все селекторы включают набор элементов, описываемых тегом `<item>`, **поддерживается неограниченное количество уровней вложенности** (пример вложенного селектора приведен в приложении А). Набор атрибутов `<item>` различен в зависимости от типа селектора. Текстовый и графический селектор типа **«calendar»** включают в себя элементы, описываемые тегом `<init>`.

6.4.2 ТЕКСТОВЫЙ СЕЛЕКТОР (<TEXT-SELECTOR>)

Текстовый селектор позволяет отображать информацию в виде кнопок с текстовыми данными, описывается тегом <text-selector>. Внешний вид приведен на рисунках 6.4.2.1, 6.4.2.3.

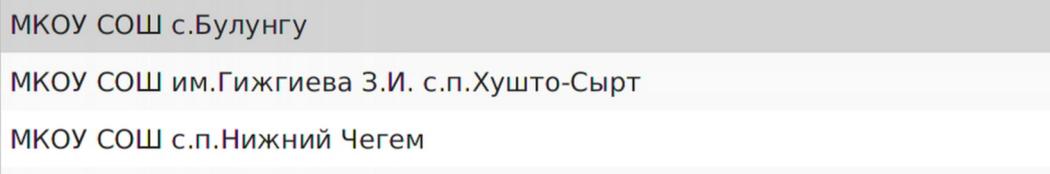


Рисунок 6.4.2.1 — Пример внешнего вида элемента <text-selector> для интерфейса «Blues»

Структура текстового селектора:

```
<text-selector name=<string>
  title=<titleType>
  message=<messageType>
  message-id=<resourceIdType>
  title-id=<resourceIdType>
  on-top=<notEmptyString>
  group-id=<nonNegativeInteger>
  flags=<flags_specification>
  selected-index=<nonNegativeInteger>
  type=[calendar|static]
  decor=[button|list]
  width=<nonNegativeInteger>
  rowId=<nonNegativeInteger>>
  <item> ... </item>
  <barcode-scanner> ... </barcode-scanner>
</text-selector>
```

Дочерние элементы: <init>, <item>, <barcode-scanner> (раздел [7.2.1](#)).

Описание атрибутов приведено в таблице 6.4.2.1.

Таблица 6.4.2.1 — Атрибуты элемента `<text-selector>`

Атрибут	Описание	Обяз.
name	Код атрибута платежа (идентификатор поля)	Да
title	Заголовок поля ввода. Предназначен для вывода пользователю названия вводимого поля (например, «Номер телефона»)	Да
title-id	Идентификатор текстовки для заголовка (титла) поля. Если для текущего выбранного языка текстовка в ресурсном файле отсутствует — используется текстовка из атрибута title	Нет
message	Информационное сообщение (подсказка) для вводимого поля	Нет
message-id	Идентификатор текстовки для сообщения (подсказки). Если для текущего выбранного языка текстовка в ресурсном файле отсутствует — используется текстовка из атрибута message	Нет
on-top	Позволяет редактировать значение поля ввода во всплывающем окне	Нет
group-id	Значение, по которому происходит группировка полей на экране, то есть поля, для которых указано одно и то же значение group-id , визуальны будут объединены на экране в один блок. Поддерживается некоторыми интерфейсами	Нет
flags	Флаги, дополнительные свойства данного поля. Поддерживаемые флаги приведены в разделе 5.4.2	Нет
selected-index	Индекс элемента, который будет выбран изначально	Нет
type	Возможные значения: <ul style="list-style-type: none">• calendar — выведет клиенту список автоматически сгенерированных дат согласно заданным настройкам	Нет

	<ul style="list-style-type: none">• static — значение по умолчанию	
decor	Определяет внешний вид селектора. Может принимать предопределенные значения в зависимости от типа экрана: button , list . Не обязателен для указания, по умолчанию задан decor=«button» — не указывается. В 5 версии ТПО в качестве заголовка экрана decor=«button» отображается название сервиса, в 7 версии — значение атрибута message экрана	Нет
width	Ширина данного поля в текущей строке в процентах, если используется конструкция <code><row></code> . Если сумма значений width всех полей внутри <code><row></code> превышает 100, то возникнет ошибка при обработке сценария. Если сумма значений width всех полей внутри <code><row></code> превышает 100, но width указан не для всех полей, то возникнет ошибка при обработке сценария. Если значение не указано, то считается, что такое поле занимает все свободное место. Если таких полей несколько, то они делят все свободное место поровну. Поддержка атрибута зависит от интерфейса. В текущей версии реализовано только для «Платформы электронных кошельков Smartkeeper».	Нет
rowId	Идентификатор строки, в которой будет расположено поле. Поддержка атрибута зависит от интерфейса. В текущей версии реализовано только для «Платформы электронных кошельков Smartkeeper».	Нет

Для селектора с типом «**static**» элементы селектора определяются с помощью тега `<item>`.

Структура `<item>` для текстового селектора:

```
<item text=<selectorTitleType>
```

```
text-id=<resourceIdType>  
value=<notEmptyString>  
enable=[true|false]  
redirect=<srvId>  
sum=<floatSum>/>
```

Дочерние элементы: отсутствуют, но в качестве вложенных возможно использовать `<text-field>`, `<numeric-field>`, `<text-selector>`, `<image-selector>`, `<set-field>`, `<info-field>`, `<autocomplete-field>`, `<group-field>`.

Атрибуты элемента `<item>` описаны в таблице 6.4.2.2.

Таблица 6.4.2.2 — Атрибуты элемента `<item>`

Атрибут	Описание	Обяз.
text	Указывается наименование элемента списка. В 5 версии ТПО допустимо использовать теги языка разметки html для форматирования выводимого сообщения. Например, "<html>полной массой до 0.75
тонн включительно"	Да
text-id	Идентификатор текстовки, которая будет подгружена в качестве значения text . Подробнее о локализации в разделе 5.4.3	Нет
value	Указывается значение атрибута платежа, добавляется в набор атрибутов при переходе на экран оплаты	Да
enable	Указывается для активации или блокировки элемента списка. Возможные значения: <ul style="list-style-type: none">• true — соответствует активному элементу (элемент доступен для выбора). Значение по умолчанию;• false — применяется для блокировки возможности выбора элемента пользователем. Для текстовых и графических селекторов работает одинаково	Нет

redirect	Указывается для перенаправления платежа на другой сервис, данные по платежу наименование элемента (сервиса) и значение ключа (id сервиса) добавляются в набор атрибутов платежа, а все остальные атрибуты платежа по сервису добавляются по сценарию из вызванного сервиса. Используется, когда оплату нельзя реализовать в рамках одного сервиса, например, при оплате сервиса банковской картой, когда по одному сервису формируются 2 реестра в зависимости от того, используется карта резидента или нерезидента банка. Используется в 7 версии ТПО. Для передачи параметров используется атрибут redirect-params="..." в теге <form> (раздел 5.2.2).	Нет
sum	Атрибут платежа, определенный как сумма по умолчанию, может быть указан дополнительно для отображения на экране фиксированной суммы оплаты по сервису	Нет

Пример:

```
<item text="Ростелеком (лицевой счет)"
value="5"
redirect="68"/>
```

В данном примере на экране текстового селектора по кнопке с наименованием «Ростелеком (лицевой счет)» для атрибута платежа, заданного в <text-selector>, присваивается значение **value** и далее оплата по сервису переводится в другой модуль ввода данных с id-service="68", а значение атрибута платежа, заданного в <text-selector>, добавляется в набор атрибутов.

Пример:

```
<text-selector decor="list" name="sosh"
message="Выберите образовательное учреждение"
```

```
title="Образовательное учреждение">
<item text="МКОУ СОШ с.Булунгу" value="МКОУ СОШ с.Булунгу"/>
<item text="МКОУ СОШ им.Гижгиева З.И. с.п.Хушто-Сырт"
value="МКОУ СОШ им.Гижгиева З.И. с.п.Хушто-Сырт"/>
<item text="МКОУ СОШ с.п.Нижний Чегем"
value="МКОУ СОШ с.п.Нижний Чегем"/>
</text-selector>
```

В данном примере на экране селекторе выводятся три кнопки с текстом, указанным в атрибуте **text** элемента `<item>` (рисунки 6.4.2.2, 6.4.3.3). По выбору в атрибут платежа **sosh** записывается значение, указанное в атрибуте **value**.

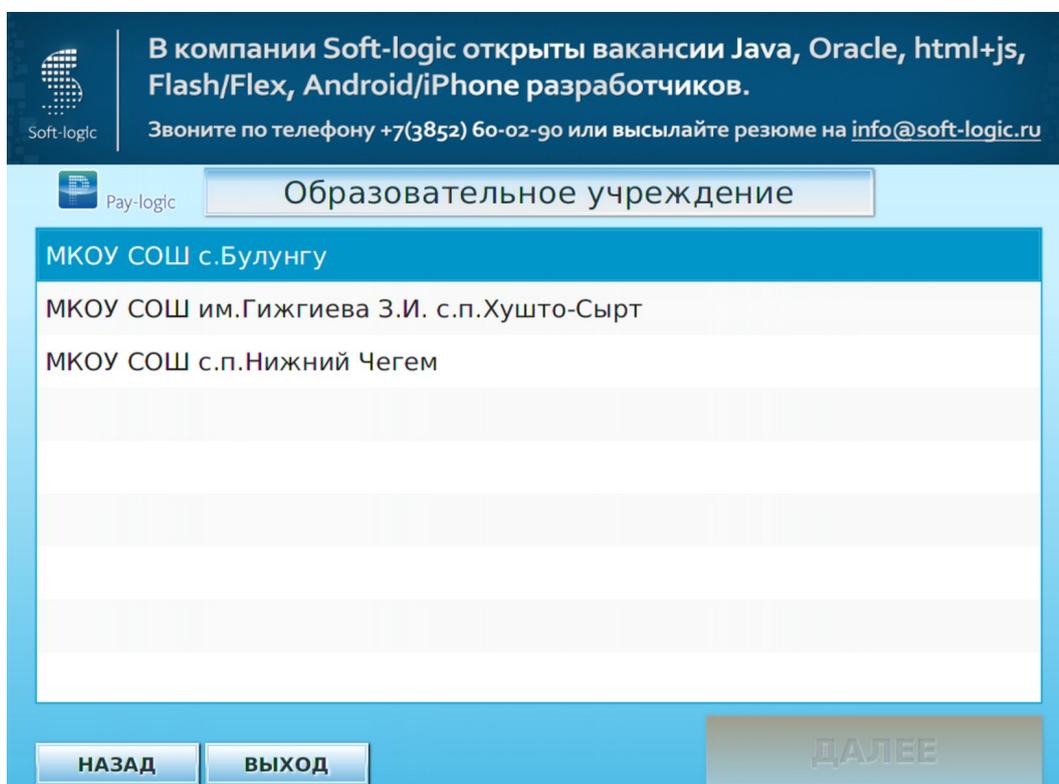


Рисунок 6.4.2.2 — Пример экрана, описанного элементом `<text-selector>`, для интерфейса «Blues»

Дочерний элемент `<init>` используется только для селектора с типом «calendar».

Структура элемента:

```
<init type=[Day|Month|Year]
      key-format=<notEmptyString>
      title-format=<notEmptyString>
      offset=<integer>
      count=<positiveInteger>/>
```

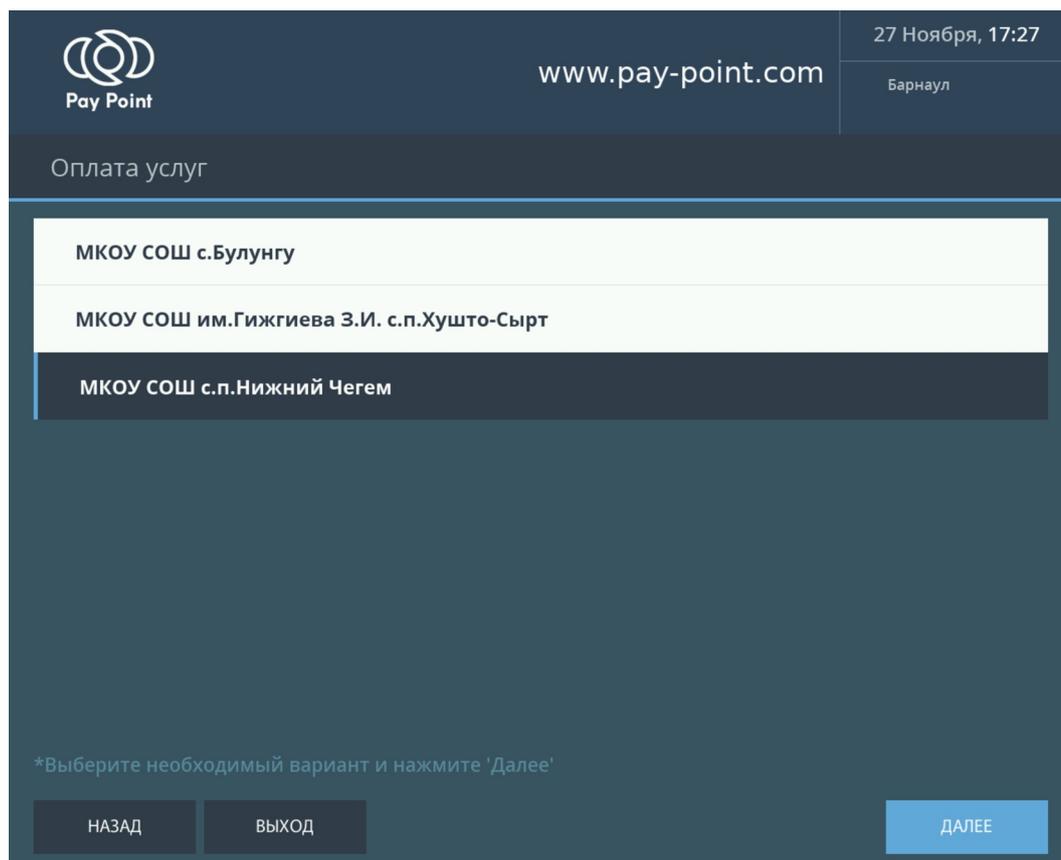


Рисунок 6.4.2.3 — Пример экрана, описанного элементом `<text-selector>`, для интерфейса «Smoke»

Дочерние элементы: отсутствуют.

Атрибуты элемента `<init>` описаны в таблице 6.4.2.3.

Таблица 6.4.2.3 — Атрибуты элемента `<init>` для текстового селектора

Атрибут	Описание	Обяз.
type	Задаёт тип выбираемого в селекторе элемента. Может принимать значения <code>day</code> , <code>month</code> , <code>year</code> (день, месяц, год) — в 7 версии ТПО. В 5 версии ТПО не реализовано. При генерации элементов цикл будет происходить соответственно по дням, месяцам или годам	Да
key-format	Задаёт формат значения даты. Формат даты задается в соответствии с <code>SimpleDateFormat</code> (день месяца — <code>dd</code> , месяц в году — <code>MM</code> , год — <code>yyyy</code>)	Да
title-format	Задаёт формат значения даты, выводимый клиенту	Да
offset	Задаёт смещение от текущей даты (может принимать как положительное, так и отрицательное значение). Единица измерения смещения зависит от значения атрибута type	Да
count	Задаёт количество выводимых записей в селекторе	Да

Пример:

```
<text-selector name="id2"
  message="Введите период начислений"
  title="Период начислений"
  type="calendar">
  <init type="Month"
    key-format="MM.yy"
    title-format="MM.yyyy"
    offset="-6"
    count="6"/>
</text-selector>
```

Экран, реализованный данной формой, приведен на рисунке 6.4.2.4.

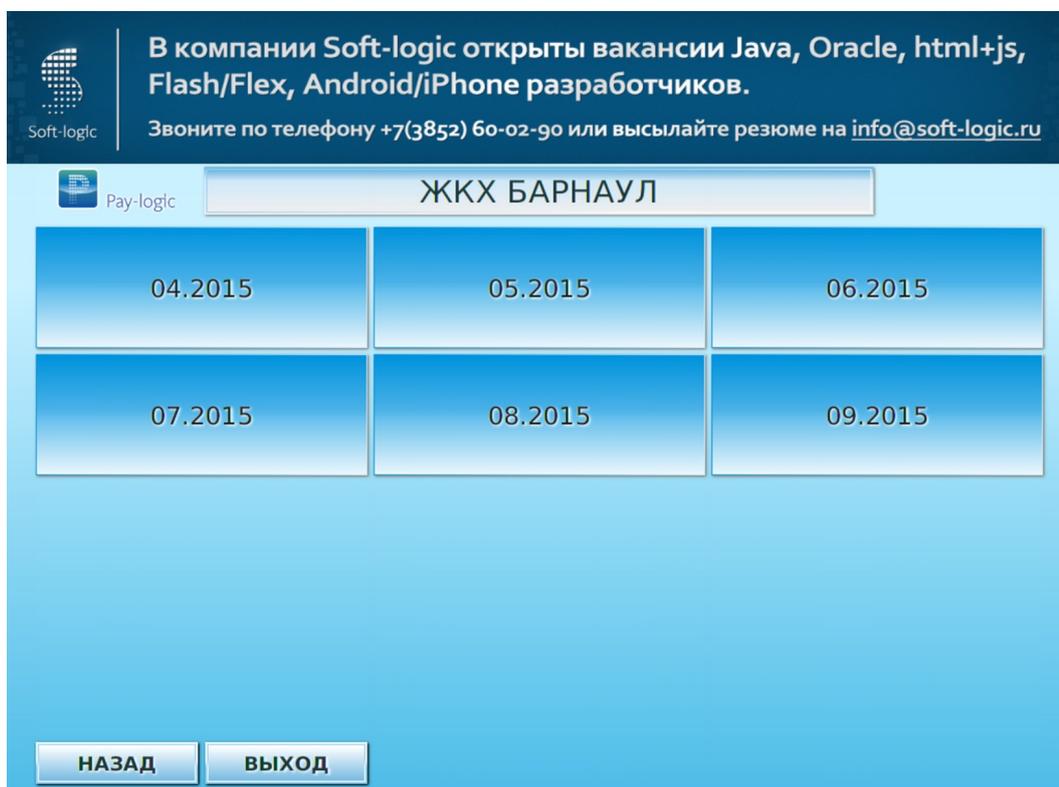


Рисунок 6.4.2.4 — Текстовый селектор с типом «calendar»

6.4.3 ГРАФИЧЕСКИЙ СЕЛЕКТОР (<IMAGE-SELECTOR>)

Графический селектор позволяет отображать информацию в виде кнопок с изображениями, описывается тегом <image-selector>. Внешний вид приведен на рисунке 6.4.3.1.



Рисунок 6.4.3.1 — Кнопки с изображениями

Структура графического селектора:

```
<image-selector name=<string>
  title=<titleType>
  message=<messageType>
  message-id=<resourceIdType>
  title-id=<resourceIdType>
  on-top=<notEmptyString>
  group-id=<nonNegativeInteger>
  flags=<flags_specification>
  selected-index=<nonNegativeInteger>
  type=[calendar|static]
  decor=[image|cards]
  width=<nonNegativeInteger>
  rowId=<nonNegativeInteger>>
  <item> ...</item>
  <barcode-scanner> ...</barcode-scanner>
</image-selector>
```

Дочерние элементы: <item>, <barcode-scanner> (раздел [7.2.1](#)).

Описание атрибутов **name**, **message**, **title**, **on-top**, **group-id**, **flags**, **width**, **rowId** приведено в разделе [5.4.1](#), **message-id**, **title-id** — в разделе [5.4.3](#), описание атрибутов **selected-index**, **type** аналогично атрибутам для текстового селектора. Атрибут **decor** может принимать значения **image** или **cards**. Если атрибут **decor** не задан, то будет отображаться текстовый селектор с **decor=«button»**.

Для графического селектора элементы определяются с помощью элемента `<item>`.

Атрибуты элементы `<item>` соответствуют атрибутам элемента `<item>` для текстового селектора за исключением наличия обязательного атрибута **img**, в котором указывается наименование файла логотипа с расширением **.png**. Размер логотипа зависит от интерфейса. В 7 версии ТПО изображения должны размещаться в каталоге **<корень ТПО>/gui/skin/<наименование интерфейса>/img/**. В 5 версии ТПО — **<корень ТПО>/img/<наименование интерфейса>/icon/service/**. В обеих версиях ТПО реализована возможность создания вложенных папок. То есть в 7 версии при указании папки **image=«dir1/dir2/dir3/image1.png»** следует файл логотипа положить в каталог **<корень ТПО>/gui/skin/<наименование интерфейса>/img/dir1/dir2/dir3/**, в 5 версии — **<корень ТПО>/img/<наименование интерфейса>/icon/service/dir1/dir2/dir3/**. Атрибут является обязательным. Значение атрибутов **text** и **text-id** используется, если файл, указанный в атрибуте **img**, не найден в системе обновления. Если используется **decor = «cards»**, то необходимо указывать атрибут **sum**.

Пример:

```
<image-selector name="id2" decor="image"
  message="Выберите электронный кошелек">
  <item img="1ticket.png" text="1 билет" value="1"/>
  <item img="2ticket.png" text="2 билета" value="2"/>
  <item img="3ticket.png" text="3 билета" value="3"/>
</image-selector>
```

В данном примере на экране селекторе заданы три элемента списка, описанные в тегах `<item>`, отображаются как кнопки с заданным логотипом — *1ticket.png*, *2ticket.png*, *3ticket.png* (рисунок 6.4.3.2, 6.4.3.3).

При выборе одного из элементов (по нажатию кнопки) атрибут платежа **id2** получает значение, указанное в атрибуте **value**, и добавляется к набору атрибутов платежа. Если выбирается элемент с наименованием «2 билета», то атрибут платежа **id2** получит значение «2».



Рисунок 6.4.3.2 — Пример графического селектора `<image-selector>` с `decor=«image»` для интерфейса «Blues»

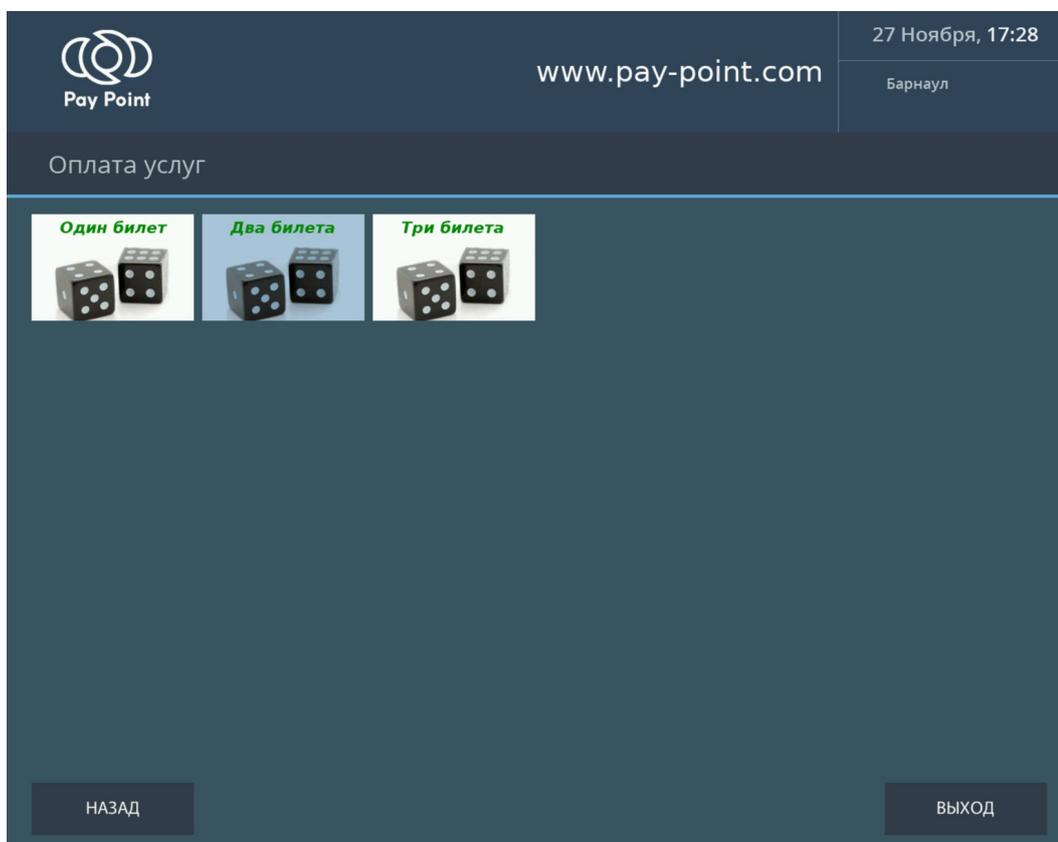


Рисунок 6.4.3.3 — Пример графического селектора `<image-selector>` с `decor=«image»` для интерфейса «Smoke»

Пример:

```
<image-selector name="id2" decor="cards"
  message="Выберите электронный кошелек">
  <item img="1ticket.png" text="1 билет" value="1" sum="13.00"/>
  <item img="2ticket.png" text="2 билета" value="2" sum="26.00"/>
  <item img="3ticket.png" text="3 билета" value="3" sum="39.00"/>
</image-selector>
```

В данном примере на экране селекторе заданы три элемента списка, описанные в тегах `<item>`, отображаются как кнопки с заданным логотипом — `1ticket.png`, `2ticket.png`, `3ticket.png` (рисунок 6.4.3.4).



Рисунок 6.4.3.4 — Графический селектор `<image-selector>` с `decor=«cards»`

При выборе одного из элементов (по нажатию кнопки) атрибут платежа `id2` получает значение, указанное в атрибуте `value`, и добавляется к набору атрибутов платежа. Если выбирается элемент с наименованием «2 билета», то атрибут платежа `id2` получит значение «2», для данного элемента списка задан атрибут `sum` (аналогичен заданию атрибута `change-sumpurchase=true` поля ввода числового значения `<numeric-field>`), значение атрибута будет использовано в качестве суммы платежа по умолчанию.

6.5 ВЫБОР ДАТЫ (<DATE-FIELD>)

Экран выбора даты реализуется с использованием элемента `<date-field>`.
Используется только на групповом экране. Внешний вид приведен на рисунке 6.5.1.



Рисунок 6.5.1 — Внешний вид элемента выбора даты

Возможности экрана ввода:

1. Позволяет выбирать дату в окне календарь.
2. Позволяет задать ограниченный диапазон дат.
3. Используется только на групповом экране.

Структура элемента выбора даты:

```
<date-field name=<string>  
  title=<titleType>  
  title-id=<resourceIdType>  
  message=<messageType>
```

```
message-id=<resourceIdType>
on-top=<notEmptyString>
group-id=<nonNegativeInteger>
flags=<flags_specification>
format=<string>
format-title=<string>
from=<nonNegativeInteger>
to=<nonNegativeInteger>
default=<string>
required=[true|false]
width=<nonNegativeInteger>
rowId=<nonNegativeInteger>/>
```

Дочерние элементы: отсутствуют.

Описание атрибутов приведено в таблице 6.5.1.

Таблица 6.5.1 — Атрибуты элемента `<date-field>`

Атрибут	Описание	Обяз.
name	Код атрибута платежа (идентификатор поля)	Да
title	Заголовок поля ввода. Предназначен для вывода пользователю названия вводимого поля (например, «Номер телефона»)	Да
title-id	Идентификатор текстовки для заголовка (титла) поля. Если для текущего выбранного языка текстовка в ресурсном файле отсутствует — используется текстовка из атрибута title	Нет
message	Информационное сообщение (подсказка) для вводимого поля	Нет
message-id	Идентификатор текстовки для сообщения (подсказки). Если для текущего выбранного языка текстовка в ресурсном файле отсутствует — используется текстовка из атрибута message	Нет
on-top	Позволяет редактировать значение поля ввода во всплывающем окне	Нет

Атрибут	Описание	Обяз.
group-id	Значение, по которому происходит группировка полей на экране, то есть поля, для которых указано одно и то же значение group-id , визуально будут объединены на экране в один блок. Поддерживается некоторыми интерфейсами	Нет
flags	Флаги, дополнительные свойства данного поля. Поддерживаемые флаги приведены в разделе 5.4.2	Нет
format	Позволяет указать формат даты. Например, "dd.ММ.уууу", dd — день, ММ — месяц, уууу — год. Указывается дата в указанном формате. DD — день, ММ — месяц, уууу — год. Возможный формат соответствует формату SimpleDateFormat (для 5 ТПО — https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html , для 7 ТПО — https://docs.oracle.com/javase/8/docs/api/java/text/SimpleDateFormat.html)	Да
format-title	Позволяет указать формат отображения пользователю	Да
from	Указывается дата в том же формате, что указан в format или число: количество предшествующих дней до текущей даты, в данном случае будет являться нижней границей заданного периода	Нет
to	Указывается дата в том же формате, что указан в format или число: количество последующих дней от текущей даты, в данном случае будет являться верхней границей заданного периода	Нет
default	Указывается дата по умолчанию для отображения в поле ввода или число: количество последующих дней от текущей даты. Если не указан, то считается, что необходимо использовать дату текущего дня. При	Нет

Атрибут	Описание	Обяз.
	этом, если дата текущего дня не попадает в заданный параметрами from, to период, то возникнет ошибка при обработке сценария. Для таких случаев необходимо указывать значение несмотря на то, что атрибут не является обязательным	
required	Используется для указания того, обязательным ли является поле <code><date-field></code> . Возможные значения: <ul style="list-style-type: none">• true — является обязательным;• false — не является обязательным	Нет
width	Ширина данного поля в текущей строке в процентах, если используется конструкция <code><row></code> . Если сумма значений width всех полей внутри <code><row></code> превышает 100, то возникнет ошибка при обработке сценария. Если сумма значений width всех полей внутри <code><row></code> превышает 100, но width указан не для всех полей, то возникнет ошибка при обработке сценария. Если значение не указано, то считается, что такое поле занимает все свободное место. Если таких полей несколько, то они делят все свободное место поровну. Поддержка атрибута зависит от интерфейса. В текущей версии реализовано только для «Платформы электронных кошельков Smartkeeper».	Нет
rowId	Идентификатор строки, в которой будет расположено поле. Поддержка атрибута зависит от интерфейса. В текущей версии реализовано только для «Платформы электронных кошельков Smartkeeper».	Нет

Пример:

```
<group-field>
  <date-field name="id4"
    title="Дата"
    format="dd.ММ.yyyy"
    format-title="dd-ММ-yyyy"
    from="03.03.2013"
    to="03.03.2014"
    default="01.03.2014"/>
</group-field>
```

Экран, соответствующий данному примеру, приведен на рисунке 6.5.2.

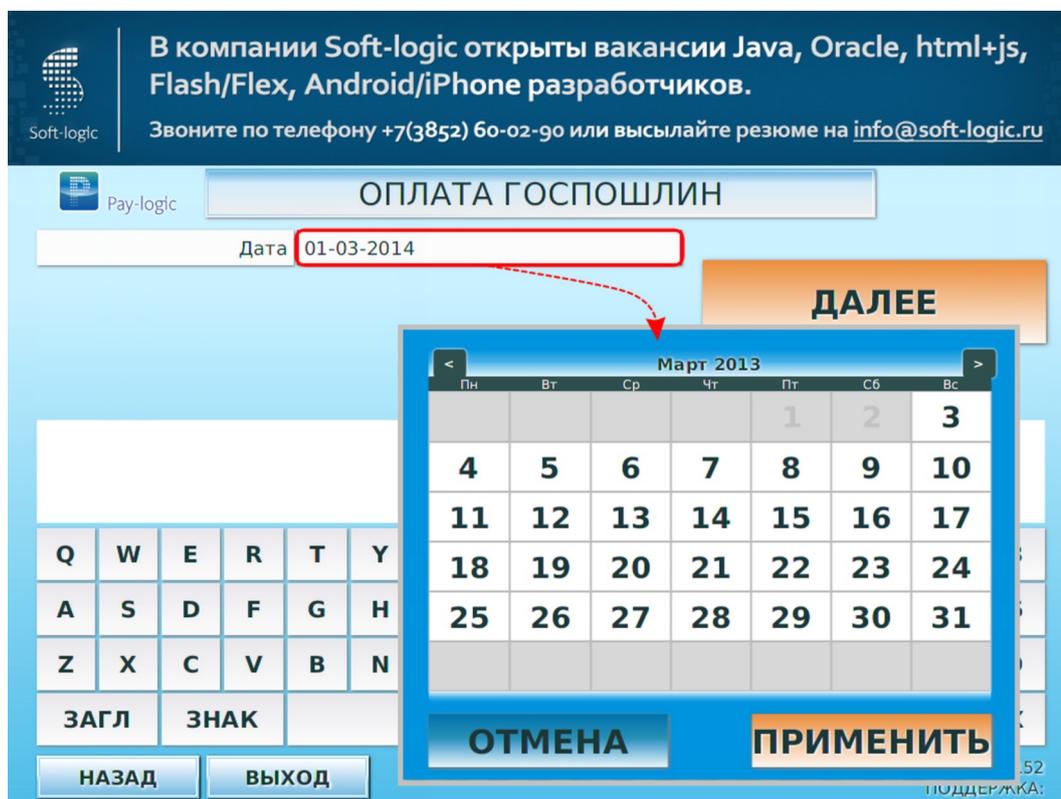


Рисунок 6.5.2 — Экран с полем выбора даты

6.6 ПРИСВОЕНИЕ ЗНАЧЕНИЯ (<SET-FIELD>)

Для тех случаев, когда атрибутам платежа необходимо автоматически (без ввода пользователем) присвоить какое-либо значение, например, при оплате по сервису, требующему некоторых реквизитов, являющихся константами (например, оплата госпошлин, штрафов) используется элемент `<set-field>`. Он позволяет добавлять и изменять произвольные данные без участия пользователя. Заданные значения могут выводиться на экран в зависимости от значения атрибута **show-on-confirm**.

Структура элемента:

```
<set-field name=<notEmptyString>
  title=<notEmptyString>
  value=<notEmptyString>
  show-on-confirm=[true|false]
  flags=<flags_specification>
  width=<nonNegativeInteger>
  rowId=<nonNegativeInteger>
  key-title-id=<resourceIdType>
  value-title-id=<resourceIdType/>
```

Дочерние элементы: отсутствуют.

Описание атрибутов приведено в таблице 6.6.1.

Таблица 6.6.1 — Атрибуты элемента `<set-field>`

Атрибут	Описание	Обяз.
name	Указывается ключ атрибута платежа (или идентификатор)	Да
title	Позволяет задать информационную подсказку для экрана подтверждения, для удобства можно	Да

Атрибут	Описание	Обяз.
	указывать наименование атрибута платежа. В кабинете отображается в окне информации о платеже в блоке «Общая информация», также может добавляться в чек	
value	Указывается значение атрибута платежа, добавляется в набор атрибутов при переходе на экран оплаты	Нет
show-on-confirm	Определяет, отображать ли на экране подтверждения: true — отображать, false — не отображать, значение по умолчанию	Нет
flags	Флаги атрибута, дополнительные свойства данного поля. Поддерживаемые флаги приведены в разделе 5.4.2	Нет
width	Ширина данного поля в текущей строке в процентах, если используется конструкция <code><row></code> . Если сумма значений width всех полей внутри <code><row></code> превышает 100, то возникнет ошибка при обработке сценария. Если сумма значений width всех полей внутри <code><row></code> превышает 100, но width указан не для всех полей, то возникнет ошибка при обработке сценария. Если значение не указано, то считается, что такое поле занимает все свободное место. Если таких полей несколько, то они делят все свободное место поровну. Поддержка атрибута зависит от интерфейса. В текущей версии реализовано только для «Платформы электронных кошельков Smartkeeper».	Нет
rowId	Идентификатор строки, в которой будет расположено поле. Поддержка атрибута зависит от интерфейса. В текущей версии реализовано только для «Платформы электронных кошельков Smartkeeper».	Нет

Атрибут	Описание	Обяз.
key-title-id	Значение ключа, по которому в ресурсах хранится текстовка, которая будет подставлена в title при отправке атрибута платежа на сервер. Текстовки расположены в каталоге <корень ТПО>/res/module/i18n/ в файле <i>input.properties</i> . Поддерживается локализация, например, файл с текстовками на английском языке будет называться <i>input_en.properties</i> . В 5 версии ТПО не обрабатывается	Нет
value-title-id	Значение ключа, по которому в ресурсах хранится текстовка, которая будет подставлена в value при отправке атрибута платежа на сервер. Текстовки расположены в каталоге <корень ТПО>/res/module/i18n/ в файле <i>input.properties</i> . Поддерживается локализация, например, файл с текстовками на английском языке будет называться <i>input_en.properties</i> . В 5 версии ТПО не обрабатывается	Нет

Пример:

```
<set-field name="rcpt" title="РСПТ"  
value="УФК по СК (ОВД по Труновскому району)" />
```

В данном примере переменной **rcpt** присваивается значение «**УФК по СК (ОВД по Труновскому району)**», переменная является атрибутом платежа и добавляется в набор атрибутов платежа. На экране терминала данная информация не выводится. В качестве подсказки будет отображаться наименование переменной, а именно «**РСПТ**».

6.7 ИНФОРМАЦИОННЫЙ ЭКРАН (<INFO-FIELD>)

Экран, позволяющий отображать информацию пользователю, описывается элементом <info-field>. Кроме того, возможно подключать внешний html-файл для вывода информации, а так же осуществлять вывод изображений. Внешний вид элемента приведен на рисунке 6.7.1. Возможности экрана ввода:

1. Позволяет форматировать текст с помощью html-тегов (выделение цветом, начертание шрифта, наклон и т. д.).
2. Позволяет размещать вспомогательное изображение.
3. Важная информация отображается на отдельном экране.

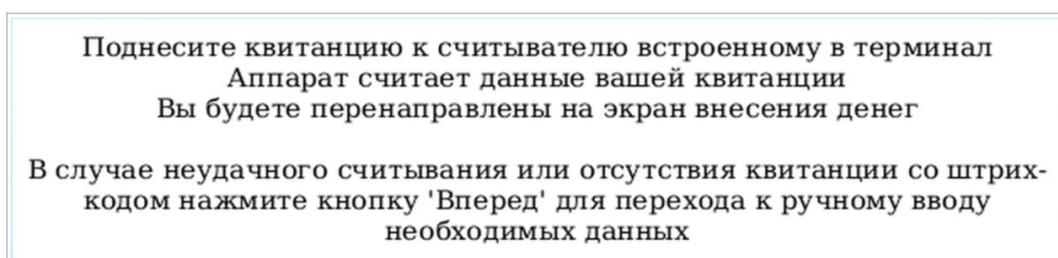


Рисунок 6.7.1 — Информационное сообщение

Структура элемента:

```
<info-field image=<fileNameType>  
  extra-button=<notEmptyString>  
  extra-file=<fileNameType>>  
  <![CDATA[форматированный текст]]>  
  <barcode-scanner> ... </barcode-scanner>  
</info-field>
```

Дочерние элементы: <barcode-scanner> (раздел [7.2](#)).

Описание других атрибутов приведено в таблице 6.7.1.

Таблица 6.7.1 — Атрибуты элемента <info-field>

Атрибут	Описание	Обяз.
image	На текущий момент не используется. Указывается наименование файла логотипа с расширением .png с заданным размером 124*80, размещённый в системе обновления в каталоге <i>img</i> .	Нет

В секции <![CDATA[]]> или в секции самого элемента <info-field> указывается текст, отображаемый на экране перехода. Значением может быть неформатированный текст, текст с использованием html-тегов, для некоторых типов интерфейсов поддерживается возможность загрузки текста из html-файла. При указании ссылки на html-файл, данный файл должен лежать в ТПО или в системе обновлений в каталоге *resources/info* относительно корневого каталога.

```
<info-field>
  file:// html-файл
</info-field>
```

или

```
<info-field>
  <![CDATA[ file:// html-файл]]>
</info-field>
```

Пример:

```
<info-field>
  <![CDATA[<html>Уважаемый платательщик! Для оплаты за технический
    осмотр, законодательство требует указания фамилии и
    адреса прописки платательщика, которые следует ввести на
    следующих экранах. Например:<br> ФИО: Иванов И
    И<br>Название населенного пункта:
    г. Барнаул <br>Адрес:ул. Ленина 15 кв 25]]>
</info-field>
```

Отображение информационного сообщения, указанного в примере, на экране терминала изображено на рисунках 6.7.2, 6.7.3.

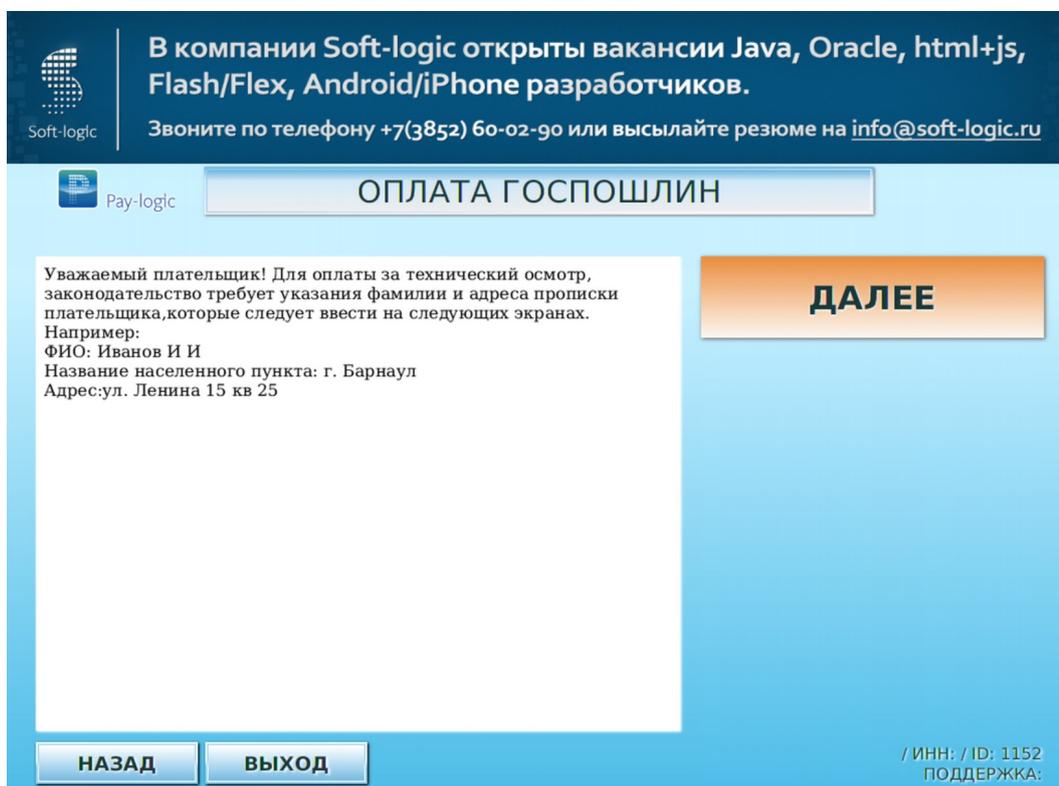


Рисунок 6.7.2 — Пример вывода информационного сообщения для интерфейса «Blues»

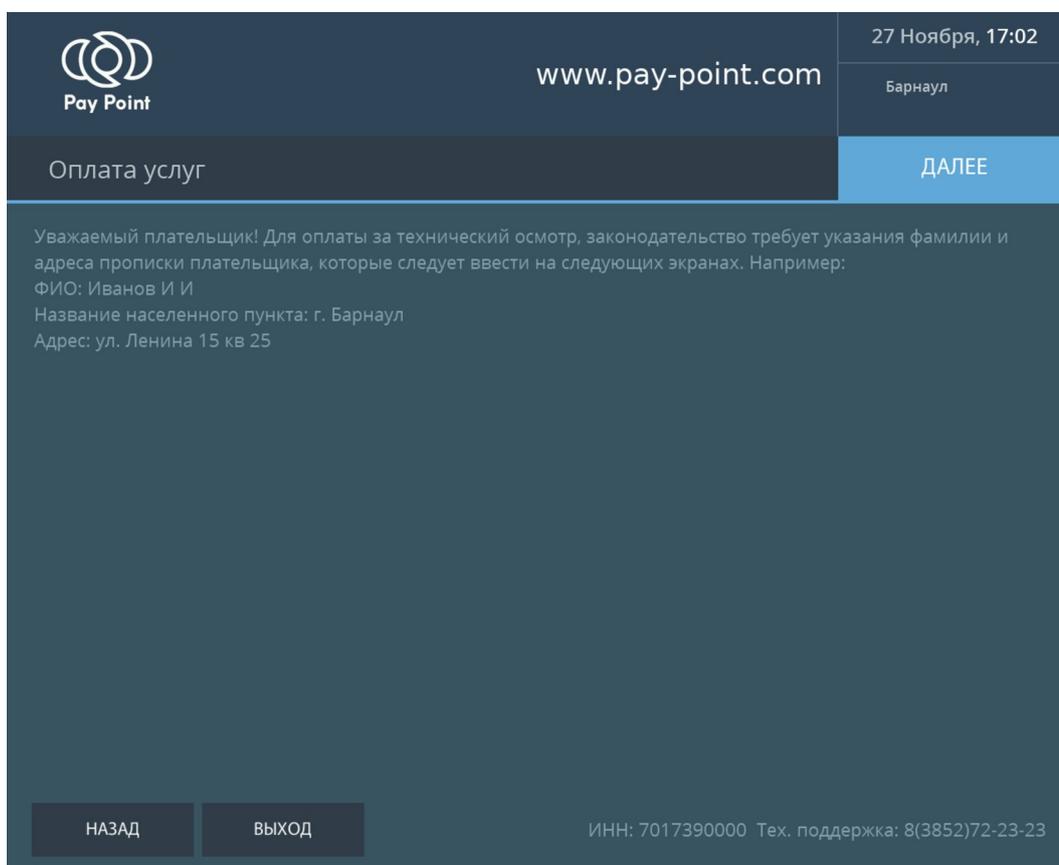


Рисунок 6.7.3 — Пример вывода информационного сообщения для интерфейса «Smoke»

Пример использования html-тегов при выводе сообщения на экран ввода `<info-field>`:

```
<info-field>
  <![CDATA[<html><br>
  <font size="4" face="Times New Roman"><b>-&nbsp;Сдаем
  рекламные места</b> <em>на экране терминалов</em> <br>
  <em>-&nbsp;Приглашаем владельцев терминалов</em><br>
  <b>-&nbsp;Подключаем поставщиков</b> <em>( ТСЖ, Управляющие
  компани, ВУЗы, ГСК, ЖСК и др. на выгодных условиях )
  </em><br><b>-&nbsp;Арендуем места</b> <em>под установку
  терминалов</em><br><br><br><br><br><br></font>
  <p align="right"> <font size="4" face="Times New
  Roman">Контактная информация:</font></p>
  ]]>
</info-field>
```

Экран, соответствующий данному примеру, приведен на рисунке 6.7.4.

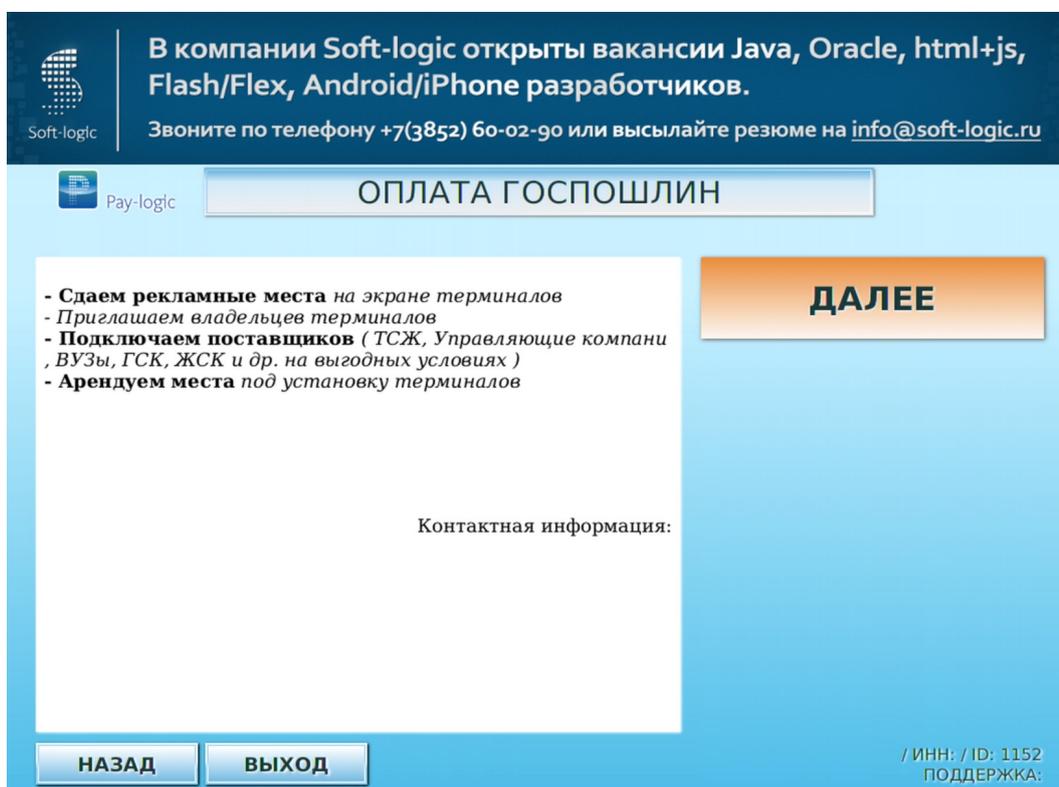


Рисунок 6.7.4 — Пример использования html-тегов при выводе сообщения на экран ввода `<info-field>`

Пример:

```
<forms>
<form service-id="1152">
  <info-field decor="agree">
    <![CDATA[<html><font size="5" color="red" weight="bold">
      ВНИМАНИЕ!!!</font><br><font size="5" color="blue"
      weight="bold">Остаток денежных
      средств будет зачислен на вашу карту сдачи!!!
    </font> <br>
      Карта сдачи будет отображена на экране с указанием
      персонального пин-кода]]>
  </info-field>
</form>
</forms>
```

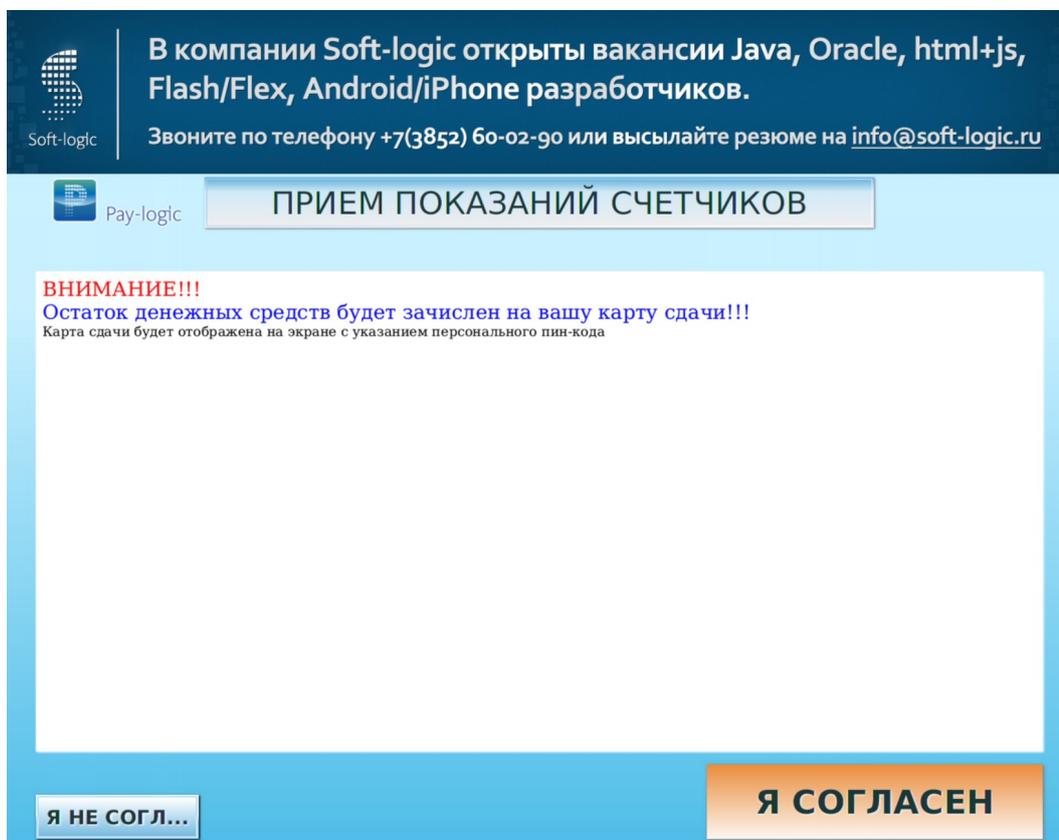


Рисунок 6.7.5 — Пример экрана `info-field` с `decor=«agree»`

6.8 АВТОЗАПОЛНЕНИЕ (<AUTOCOMPLETE-FIELD>)

Элемент `<autocomplete-field>` отображает поле ввода текста, позволяет по введенной части слова выбрать значение из выпадающего списка. Список возможных значений определяется источником (элемент `<data-source>`). Работает только на групповом экране. Доступен для обеих версий ТПО.

Структура элемента:

```
<autocomplete-field name=<string>
    title=<titleType>
    title-id=<resourceIdType>
    message=<messageType>
    message-id=<resourceIdType>
    on-top=<notEmptyString>
    group-id=<nonNegativeInteger>
    flags=<flags_specification>
    keyboard=<keyboard_specification>
    letter-case=[UpperCase|LowerCase]
    xbutton=<notEmptyString>
    min-chars=<positiveInteger>
    max-items=<positiveInteger>
    optional=[true|false]
    width=<nonNegativeInteger>
    rowId=<nonNegativeInteger>
    allow-custom=[true|false]>
    <data-source .../>
</autocomplete-field>
```

Дочерние элементы: `<data-source>`.

Описание атрибутов приведено в таблице 6.8.1.

Таблица 6.8.1 — Атрибуты элемента <autocomplete-field>

Атрибут	Описание	Обяз.
name	Код атрибута платежа (идентификатор поля)	Да
title	Заголовок поля ввода. Предназначен для вывода пользователю названия вводимого поля (например, «Номер телефона»)	Да
title-id	Идентификатор текстовки для заголовка (титла) поля. Если для текущего выбранного языка текстовка в ресурсном файле отсутствует — используется текстовка из атрибута title	Нет
message	Информационное сообщение (подсказка) для вводимого поля	Нет
message-id	Идентификатор текстовки для сообщения (подсказки). Если для текущего выбранного языка текстовка в ресурсном файле отсутствует — используется текстовка из атрибута message	Нет
on-top	Позволяет редактировать значение поля ввода во всплывающем окне	Нет
group-id	Значение, по которому происходит группировка полей на экране, то есть поля, для которых указано одно и то же значение group-id , визуально будут объединены на экране в один блок. Поддерживается некоторыми интерфейсами	Нет
flags	Флаги, дополнительные свойства данного поля. Поддерживаемые флаги приведены в разделе 5.4.2	Нет
keyboard	Указывается тип клавиатуры, отображаемой на экране терминала. Типы клавиатур описаны в разделе 5.3	Да
letter-case	Предоставляет возможность установить первую букву сообщения заглавной. Возможные значения: <ul style="list-style-type: none">• UpperCase — заглавная буква;• LowerCase — строчная буква	Да

Атрибут	Описание	Обяз.
xbutton	Указывается один символ из символьной или буквенной раскладки клавиатуры, который будет отображаться вместо функциональной кнопки «Стереть» (или сброс введенного значения) для цифровой раскладки. Данный атрибут используется только с цифровой раскладкой. При указании клавиатуры в 5 версии ТПО важен регистр (digital — неверно)	Нет
min-chars	Минимальное количество символов, начиная с которого ищется список значений из файла источника (начинает работать поиск значений)	Да
max-items	Максимальное количество найденных и отображенных значений из файла источника, удовлетворяющее введенной части слова в поле ввода больше которого считать фразу недостаточной для осуществления поиска и не отображать найденные значения	Да
optional	Определяет обязательность заполнения поля для использования выпадающего списка, доступные значения: <ul data-bbox="619 1323 1385 1453" style="list-style-type: none">• true — поле должно быть заполнено обязательно;• false — поле можно оставить пустым	Нет
allow-custom	Параметр определяет возможность ввода отсутствующих в справочнике значений. В ТПО 5 недоступен. Возможные значения: <ul data-bbox="619 1597 1385 1807" style="list-style-type: none">• true — разрешает принимать значения, отсутствующие в справочнике. Если пользователь вводит значение, ввел его не полностью и ничего не выбрал, либо ввел его таким, которое приводит к пустому списку	Нет

Атрибут	Описание	Обяз.																												
	<p>выбора, поле принимает введенное значение в качестве значения;</p> <ul style="list-style-type: none"> • false — запрещает принимать значения, отсутствующие в справочнике. Если значение из предлагаемых не выбрано, поле остается пустым. Значение по умолчанию <p>В совокупности с параметром optional реализуется следующая логика работы:</p> <table border="1" data-bbox="576 781 1374 1364"> <thead> <tr> <th rowspan="2">optional</th> <th rowspan="2">allow-custom</th> <th colspan="3">Введено значение</th> </tr> <tr> <th>пустое</th> <th>не из списка</th> <th>из списка</th> </tr> </thead> <tbody> <tr> <td>false</td> <td>false</td> <td>неактивная</td> <td>неактивная</td> <td>активная</td> </tr> <tr> <td>false</td> <td>true</td> <td>неактивная</td> <td>активная</td> <td>активная</td> </tr> <tr> <td>true</td> <td>false</td> <td>активная</td> <td>неактивная</td> <td>активная</td> </tr> <tr> <td>true</td> <td>true</td> <td>активная</td> <td>активная</td> <td>активная</td> </tr> </tbody> </table> <p>Пример:</p> <pre data-bbox="612 1460 1374 1800"> <autocomplete-field name="departure-point" title="Пункт отправления" min-chars="3" max-items="7" keyboard="any:[ru,en,symb]:lower:true" optional="false" allow-custom="true"> <data-source type="plain" url="stations.dat" partial="none" /> <help> <![CDATA[<html>Введите пункт </pre>	optional	allow-custom	Введено значение			пустое	не из списка	из списка	false	false	неактивная	неактивная	активная	false	true	неактивная	активная	активная	true	false	активная	неактивная	активная	true	true	активная	активная	активная	
optional	allow-custom			Введено значение																										
		пустое	не из списка	из списка																										
false	false	неактивная	неактивная	активная																										
false	true	неактивная	активная	активная																										
true	false	активная	неактивная	активная																										
true	true	активная	активная	активная																										

Атрибут	Описание	Обяз.
	<pre>отправления]]> </help> </autocomplete-field></pre>	
width	<p>Ширина данного поля в текущей строке в процентах, если используется конструкция <code><row></code>.</p> <p>Если сумма значений width всех полей внутри <code><row></code> превышает 100, то возникнет ошибка при обработке сценария. Если сумма значений width всех полей внутри <code><row></code> превышает 100, но width указан не для всех полей, то возникнет ошибка при обработке сценария. Если значение не указано, то считается, что такое поле занимает все свободное место. Если таких полей несколько, то они делят все свободное место поровну.</p> <p>Поддержка атрибута зависит от интерфейса. В текущей версии реализовано только для «Платформы электронных кошельков Smartkeeper».</p>	Нет
rowId	<p>Идентификатор строки, в которой будет расположено поле.</p> <p>Поддержка атрибута зависит от интерфейса. В текущей версии реализовано только для «Платформы электронных кошельков Smartkeeper».</p>	Нет

Элемент `<data-source>` содержит описание источника данных для поля автозаполнения `<autocomplete-field>`.

Структура элемента:

```
<data-source type=[plain]
  url=<notEmptyString>
  default=<notEmptyString>/>
```

Дочерние элементы: отсутствуют.

Описание атрибута **default** аналогично описанию для текстового поля (раздел 6.1), описание других атрибутов приведено в таблице 6.8.2.

Таблица 6.8.2 — Атрибуты элемента `<data-source>`

Атрибут	Описание	Обяз.
type	Указывается тип источника данных. Определены значения: plain — источником данных служит файл	Да
url	Имя файла источника данных. Источник данных должен быть расположен по адресу <code><корень ТПО>/res/module/input/ds/<имя файла>.dat</code> для 7 версии ТПО или <code><корень ТПО>/resources/autocomplete/<имя файла>.dat</code> для 5 версии ТПО. Пример: <pre>url="stations.dat"</pre>	Да
order	Определяет вариант сортировки. Возможны следующие значения: <ul style="list-style-type: none"> name — сортировка по отображаемому названию. Значение по умолчанию; rate — сортировка по рейтингу 	Нет

Пример:

```
<group-field>
  <autocomplete-field name="id3" title="Населенный пункт:"
    min-chars="3" max-items="5"
    letter-case="UpperCase" optional="false"
    keyboard="any: [ru,uk,en,symb]:lower:true">
    <data-source type="plain" url="1152/stations.dat"
      order="name"/>
  </autocomplete-field>
</group-field>
```

Пример содержимого файла источника данных `stations.dat`:

```
2004707 [5] : РОССИЯ АМБАРНЫЙ : АМБАРНЫЙ
2020774 [4] : РОССИЯ ЧЕМИЗОВКА : ЧЕМИЗОВКА
2020775 [1] : РОССИЯ ЖЕРНОВКА : ЖЕРНОВКА
2020776 [7] : РОССИЯ СОБОЛЕВКА : СОБОЛЕВКА
2020782 [3] : РОССИЯ БАРНУКОВ : КАБАРНУКОВКА
2040513 [2] : РОССИЯ БАРНЕВКА : БАРНЕВКА
2044800 [6] : РОССИЯ БАРНАУЛ : БАРНАУЛ
```

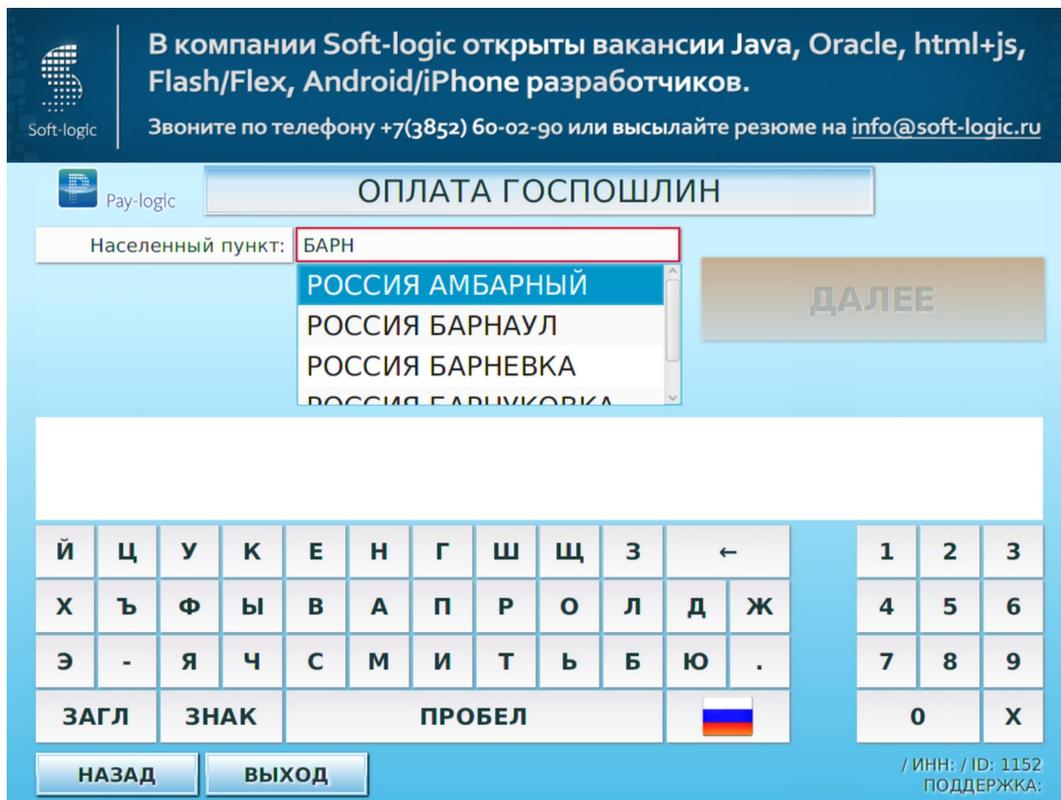
Структура файла: значение, указанное в квадратных скобках «[]», представляет собой вес значения. Большие значения в списке вариантов отображаются первыми среди вариантов при установленном **order="rate"** в `<data-source/>`. Значение до «[]» является значением создаваемого атрибута платежа, вторая строка (после символа «[]» — заголовок для значения атрибута платежа. Если нет символа «:», то значение строки станет и значением, и заголовком. Все значения после второго «:» распознаются, как альтернативное название. Количество альтернативных названий неограниченно, указываются через «:». То есть, формат строки содержимого файла источника имеет следующий вид:

```
value[rate]:valueTitle:altName1,altName2, ... , altNameN
```

Значение элементов следующее:

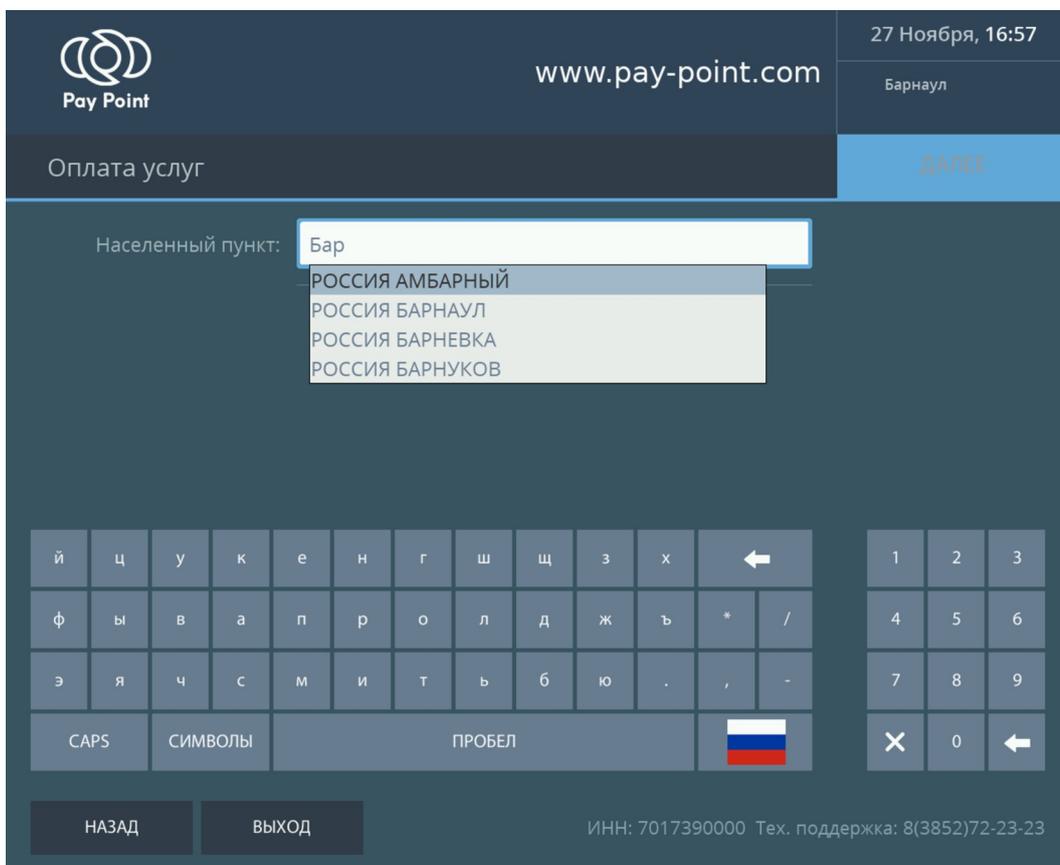
1. **value** — значение элемента. Обязательное поле.
2. **[rate]** — рейтинг. По умолчанию **rate = 0**.
3. **valueTitle** — заголовок элемента. По умолчанию **valueTitle = value**.
4. **altName1, altName2, ... , altNameN** — альтернативные названия. По умолчанию отсутствуют.
5. Если используется сортировка **order="name"**, то **[rate]** из названия исключается и нигде не используется.

Экран, соответствующий примеру, приведен на рисунке 6.8.1.



The screenshot shows a web interface for 'ОПЛАТА ГОСПОШЛИН'. At the top, there is a banner for Soft-logic with job openings for Java, Oracle, HTML+JS, Flash/Flex, and Android/iPhone developers. Below the banner, the 'Pay-logic' logo is visible. The main form has a title 'ОПЛАТА ГОСПОШЛИН' and a field for 'Населенный пункт:' (Settlement) containing 'БАРН'. A dropdown menu is open, showing suggestions: 'РОССИЯ АМБАРНЫЙ', 'РОССИЯ БАРНАУЛ', 'РОССИЯ БАРНЕВКА', and 'РОССИЯ БАЛУКОВКА'. To the right of the dropdown is a 'ДАЛЕЕ' button. Below the form is a Russian keyboard with Cyrillic characters and a numeric keypad. At the bottom, there are 'НАЗАД' and 'ВЫХОД' buttons, and a footer with the text '/ ИНН: / ID: 1152' and 'ПОДДЕРЖКА:'.

Рисунок 6.8.1 — Пример ввода данных на экране с автозаполнением



The screenshot shows the Pay Point website interface. At the top left is the Pay Point logo. The top right shows the date and time: 27 ноября, 16:57. Below this, the website URL www.pay-point.com is displayed. The main heading is "Оплата услуг" (Payment of services). A dropdown menu is open for the "Населенный пункт:" (Populated point) field, which currently contains "Бар" (Bar). The dropdown options are: "РОССИЯ АМБАРНЫЙ", "РОССИЯ БАРНАУЛ", "РОССИЯ БАРНЕВКА", and "РОССИЯ БАРНУКОВ". Below the dropdown is a virtual keyboard with Cyrillic characters, function keys like "CAPS", "СИМВОЛЫ", "ПРОБЕЛ", and a numeric keypad. At the bottom, there are "НАЗАД" (Back) and "ВЫХОД" (Exit) buttons, and contact information: ИНН: 7017390000, Тех. поддержка: 8(3852)72-23-23.

Рисунок 6.8.2 — Пример ввода данных на экране с автозаполнением

6.9 ГРУППОВОЙ ЭКРАН (<GROUP-FIELD>)

Групповой экран позволяет отображать несколько полей ввода на одном экране. Используется при вводе логически сгруппированной информации. Например, фамилию, имя, отчество плательщика удобно вводить на одном экране ввода или при оплате банковского платежа реквизиты банка получателя логичнее расположить на одном экране перехода. задается элементом <group-field>. Внешний вид приведен на рисунке 6.9.1.

Фамилия	Петров
Имя	Сергей
Отчество	Федорович

Рисунок 6.9.1 — Элементы группового экрана

Элемент <group-field> может использоваться со списком полей и полноразмерной клавиатурой — вариант 1.

Поддерживается возможность использовать несколько полей ввода: <text-field>, <numeric-field>, <date-field>, <text-selector>, <autocomplete-field>. Ветвления на групповых экранах не поддерживаются, то есть в <group-field> нельзя использовать еще один элемент <group-field>. Структура элемента:

```
<group-field>
  <text-field> ... </text-field>
  <numeric-field> ... <numeric-field/>
  <date-field ... />
  ...
</group-field>
```

Вариант 2 — с возможностью использования всплывающих окон на одном групповом экране ввода.

Структура элемента:

```
<group-field message=<string>
    type=<notEmptyString>>
    <text-field ...
        on-top=<notEmptyString>>
    ...
</text-field>
<text-field >
    ...
</text-field>
...
</group-field>
```

Функционал всплывающих окон реализован для текстового и цифрового полей ввода с использованием атрибута **on-top="numeric/popup"**.

Вариант 3 — с возможностью размещения указанных внутри него полей ввода в одну строчку. Возможность использования данного варианта зависит от интерфейса. В текущей версии реализовано только для «Платформы электронных кошельков Smartkeeper».

Структура элемента:

```
<group-field>
    <row orientation=[PORTRAIT|LANDSCAPE]>
    <text-field width=<nonNegativeInteger>
        ...
    </text-field>
    ...
    <placeholder width=<nonNegativeInteger>/>
    ...
    </row>
</group-field>
```

Описание применения элемента приведено в разделе [5.2.4](#).

Описание атрибута **title** приведено в разделе 5.4.1 (используется только в 5 версии)

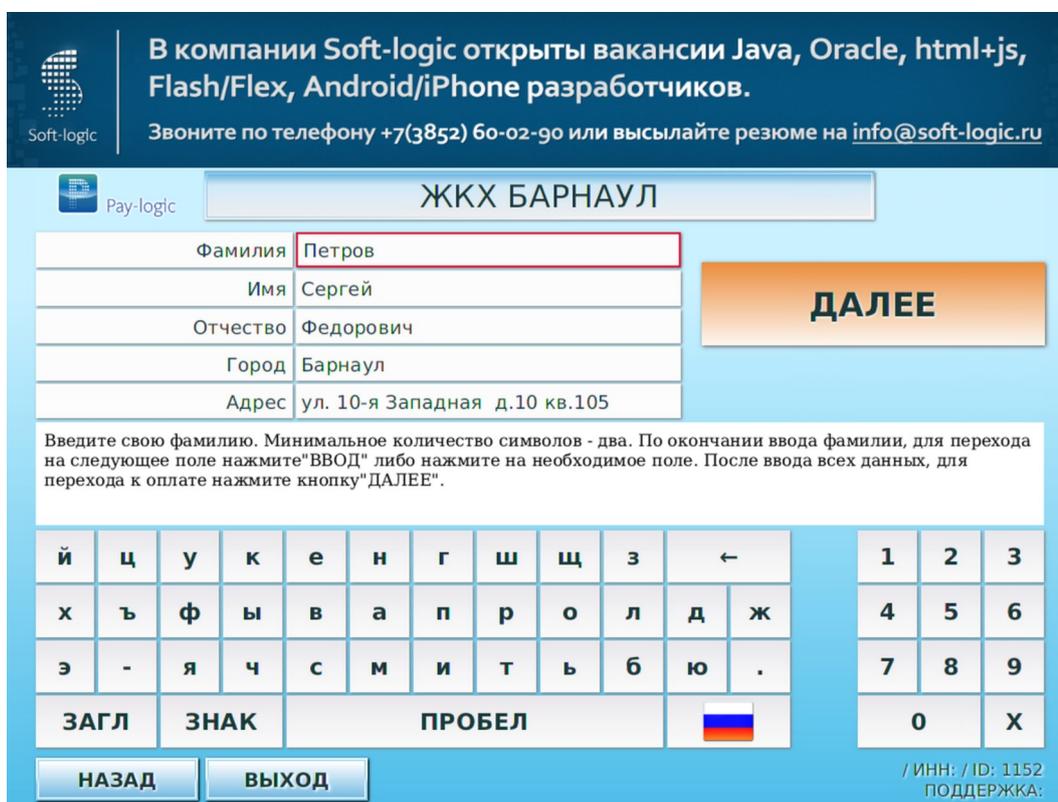
ТПО).

Пример (1 вариант):

```
<group-field>
<!--Создание поля ввода, со следующими параметрами: активна цифровая,
символьная и буквенная клавиатура, язык ввода - русский, без возможности
смены, максимальное количество вводимых символов 40. Название поля ввода:
"Фамилия"-->
<text-field name="fam" keyboard="any:[ru,symb]:lower:true" max-len="40"
example="Иванов" message="Введите Фамилию" title="Фамилия">
<!--Подсказка, отображается на экране-->
<help><![CDATA[<html>Введите <b>ФАМИЛИЮ</b> (без сокращений), на кого
оформлен кредитный договор. Пример ввода: ИВАНОВ.]]></help>
<!--Регулярное выражение для валидации вводимого номера-->
<verify>
<rule regex="^[a-яА-Я0-9-]{2,40}$"/>
</verify>
</text-field>
<!--Создание поля ввода, со следующими параметрами: активна цифровая,
символьная и буквенная клавиатура, язык ввода - русский, без возможности
смены, максимальное количество вводимых клиентом символов 40. Название
поля ввода: "Имя"-->
<text-field name="name" keyboard="any:[ru,symb]:lower:true" max-len="40"
example="Иван" message="Введите Имя" title="Имя">
<!--Подсказка, отображается на экране-->
<help><![CDATA[<html>Введите <b>ИМЯ</b> (без сокращений), на кого
оформлен кредитный договор. Пример ввода: ИВАН.]]> </help>
<!--Регулярное выражение для валидации вводимого номера-->
<verify>
<rule regex="^[a-яА-Я-]{2,40}$"/>
</verify>
</text-field>
<!--Создание поля ввода, со следующими параметрами: активна цифровая,
символьная и буквенная клавиатура, язык ввода - русский, без возможности
смены, максимальное количество вводимых клиентом символов 40. Название
поля ввода: "Отчество"-->
<text-field name="name2" keyboard="any:[ru,symb]:lower:true"
max-len="40" example="Иванович" message="Введите Отчество"
title="Отчество">
<!--Подсказка, отображается на экране-->
<help><![CDATA[<html>Введите <b>ОТЧЕСТВО</b> (без сокращений), на кого
```

```
оформлен кредитный договор. Пример ввода: ИВАНОВИЧ.]]></help>  
<!--Регулярное выражение для валидации вводимого номера-->  
<verify>  
  <rule regex="^[a-яА-Я-]{2,40}$"/>  
</verify>  
</text-field>  
</group-field>
```

В данном примере `<group-field>` позволяет вывести на одном экране три текстовых поля ввода (рисунок 6.9.2).



The screenshot shows a payment form for 'ЖКХ БАРНАУЛ'. The form contains several text input fields: 'Фамилия' (Family name) with 'Петров' entered, 'Имя' (Name) with 'Сергей', 'Отчество' (Patronymic) with 'Федорович', 'Город' (City) with 'Барнаул', and 'Адрес' (Address) with 'ул. 10-я Западная д.10 кв.105'. A large orange button labeled 'ДАЛЕЕ' (Next) is positioned to the right of the fields. Below the form is a keyboard overlay with Cyrillic characters and a numeric keypad. At the bottom right, there is a small text area: '/ ИНН: / ID: 1152 ПОДДЕРЖКА:'.

Рисунок 6.9.2 — Пример экрана `<group-field>` для интерфейса «Blues»

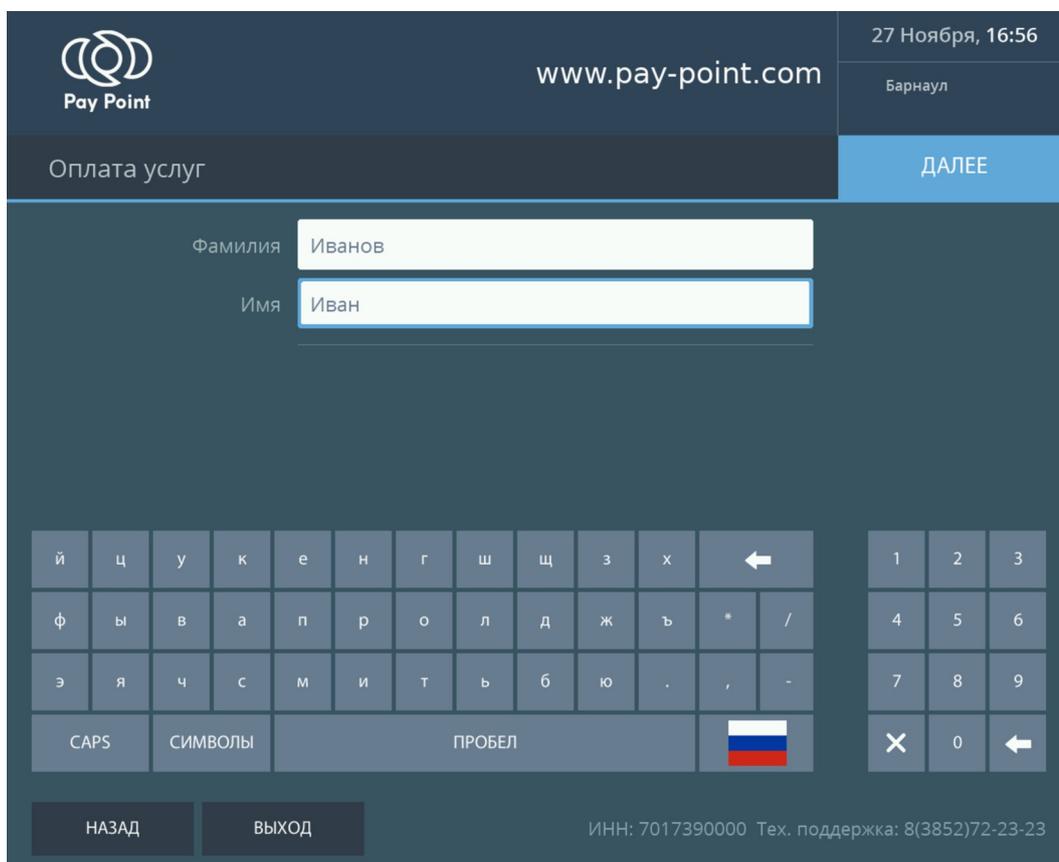


Рисунок 6.9.3 — Пример экрана `<group-field>` для интерфейса «Smoke»

В каждом текстовом поле атрибуту платежа, указанному в атрибуте **name** текстового поля ввода `<text-field>`, присваивается значение и добавляется к набору атрибутов платежа.

Пример (2 вариант):

```
<group-field>
  <numeric-field on-top="numeric/popup" name="account"
    min="0" max="999999999999"
    message="Введите номер лицевого счета"
    title="Лицевой счет">
  <!--Подсказка, отображается на терминале-->
  <help><![CDATA[<html>Введите номер лицевого счета]]></help>
</numeric-field>
```

```
<numeric-field on-top="numeric/popup" name="contract"
  min="1" max="999"
  message="Введите номер договора"
  title="Номер договора">
<!--Подсказка, отображается на терминале-->
<help><![CDATA[<html>Введите номер договора]]> </help>
</numeric-field>
</group-field>
```

В данном примере при нажатии на поле ввода открывается отдельное окно с цифровой клавиатурой (рисунок 6.9.4).

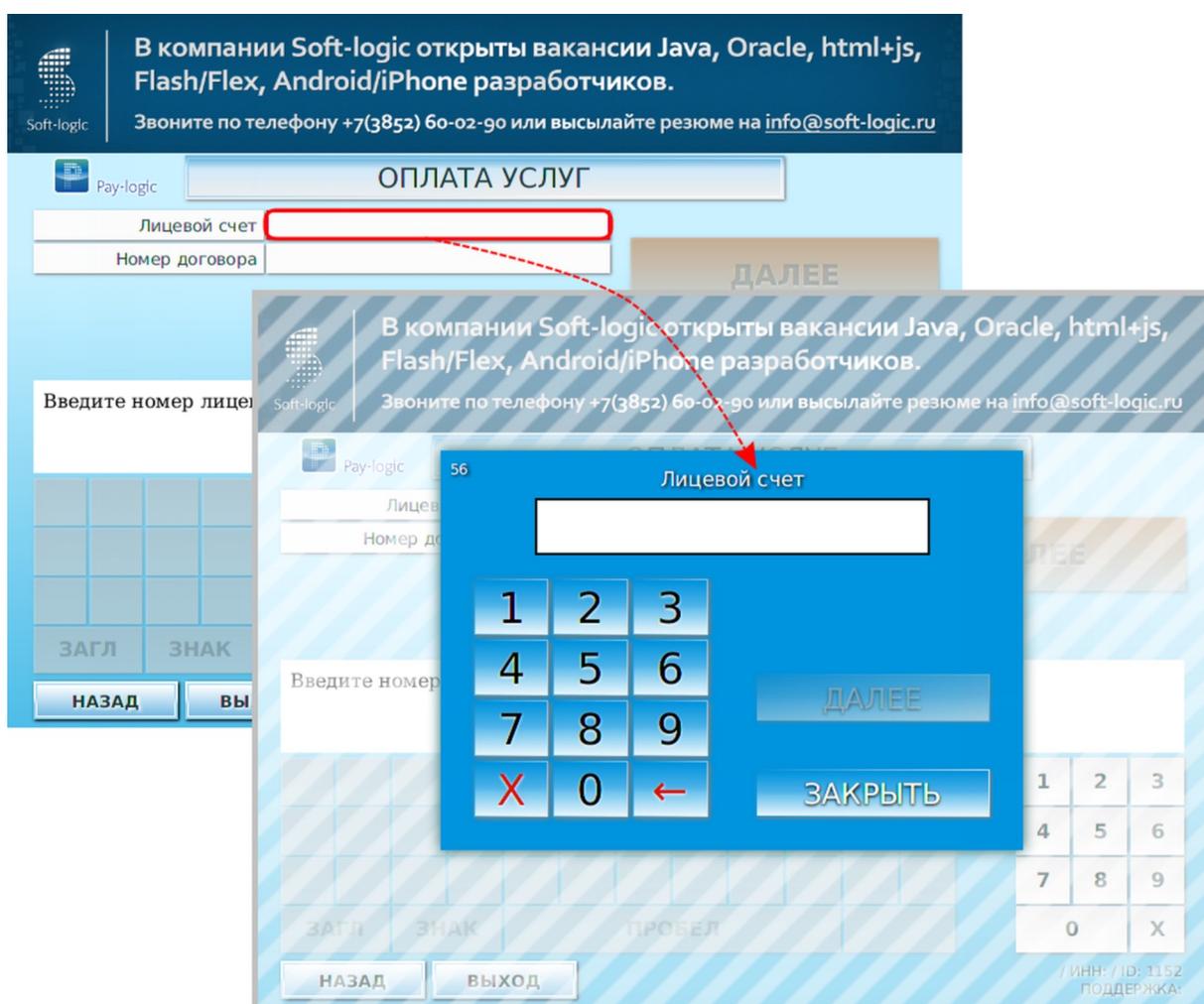


Рисунок 6.9.4 — Всплывающее окно с цифровой клавиатурой

6.10 ЭКРАН ПОДТВЕРЖДЕНИЯ С КОМИССИЕЙ (<COMMISSION-CONFIRM>)

На некоторых интерфейсах поддерживается экран подтверждения с отображением комиссии (рисунок 6.10.1).

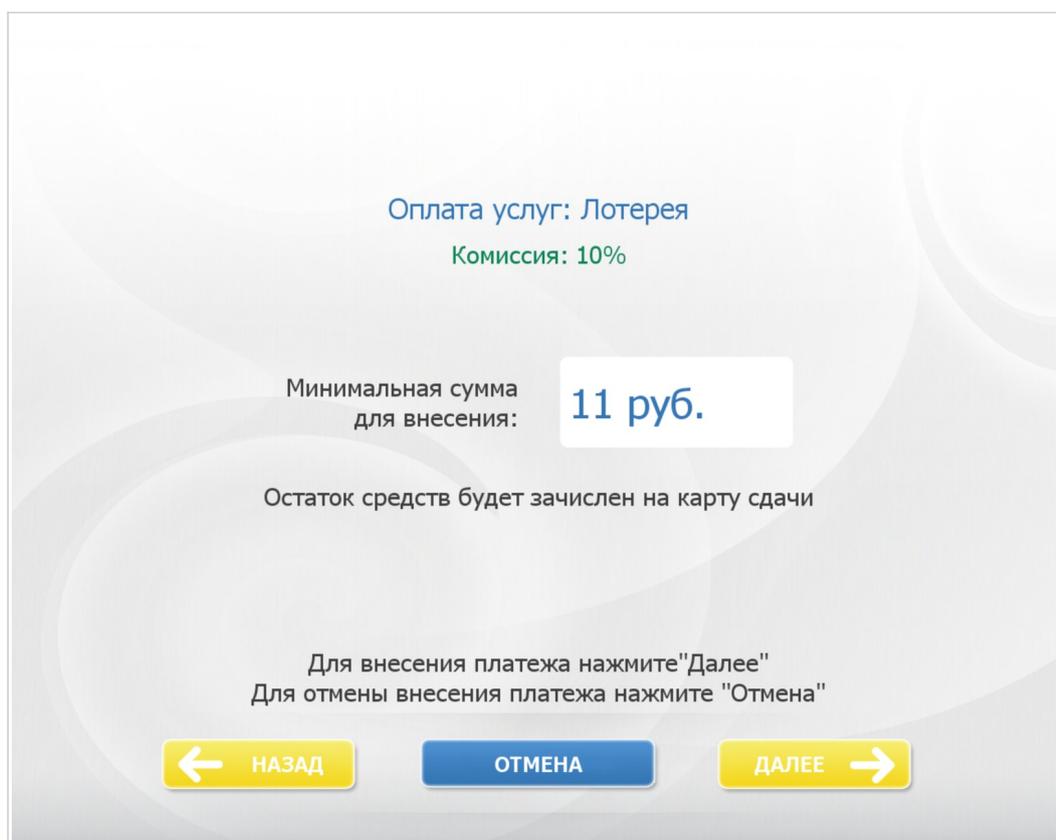


Рисунок 6.10.1 — Экран подтверждения с комиссией

Для отображения такого экрана в форме необходимо прописать строку <commission-confirm>.

Пример:

```
<form service-id="1152">
  <commission-confirm/>
  <!--Создание поля ввода, со следующими параметрами: активна
цифровая клавиатура, максимальное количество вводимых клиентом символов
11. Название поля ввода: "Номер счета"-->
  <text-field name="account" keyboard="Digital" max-len="11"
    message="Введите номер лицевого счета"
    title="Номер счета">
  <!--Регулярное выражение для валидации вводимого номера-->
  <verify>
    <rule regex="^\d{6,11}$"/>
  </verify>
  <!--Подсказка, отображается на терминале-->
  <help>Введите номер лицевого счета</help>
</text-field>
</form>
```

Поддерживается ли экран вашим интерфейсом уточняйте у сотрудников компании Soft-logic.

7 ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ

7.1 ПОДСКАЗКА (<HELP>)

Для вывода текстового сообщения (подсказки) на экране ввода используются элемент `<help>`. Для представления текста в форматированном виде допустимо использование html-тегов. Для этого используйте тег `<html>` внутри элемента `<![CDATA[]]>`. Отображаемый текст может размещаться в несколько строк, если позволяет реализация интерфейса терминала. Например, поле подсказки для стандартного интерфейса Bluesphere2 при использовании полноценной клавиатуры, может содержать несколько строк.

При использовании экрана `<numeric-field>` `<help>` на экране может не отображаться, если интерфейс сконфигурирован для показа рекламных баннеров или же в описании экрана явно задана настройка декор, которая отображает баннер вместо `<help>`. Доступно только на 5 версии ТПО.

```
<help>
  <![CDATA[<html>    форматированный текст ]]>
</help>
```

Атрибутов не имеет. В секции `<![CDATA[]]>` создается сообщение, выводимое как подсказка для поля ввода.

В 7 версии ТПО в элемента `<help>` возможно указание типа точки. Например, `<help types="4">` — для электронного кошелька. Если не будет найдена подсказка для типа точки, то будет использована подсказка по умолчанию.

Возможные флаги:

1. **TERMINAL=0x01** — подсказка для терминала.

2. **RMA=0x02** — подсказка для РМА.
3. **KEEPER=0x04** — подсказка для кошелька.
4. **KEEPER-ANDROID=0x08** — подсказка для кошелька Android.
5. **KEEPER-IOS=0x10** — подсказка для iOS-клиентов.
6. **KEEPER-WEB=0x20** — подсказка для веб-клиентов.
7. **MOBILE-ANDROID=0x40** — подсказка для терминалов Android.
8. **MOBILE-IOS=0x80** — подсказка для терминалов iOS.

Пример:

```
<text-field>
  <help>
  Введите номер телефона
  </help>
  <help types="4">
  Введите номер телефона или выберите его из вашей телефонной
  книги
  </help>
  <help types="RMA | TERMINAL">
  Введите номер телефона, после чего нажмите кнопку ДАЛЕЕ
  </help>
</text-field>
```

Кроме того, при выводе подсказок возможно использовать `<help>` с атрибутом `id`, содержащим код текстовки.

Пример:

```
<help id="msg.help">
```

В данном примере "msg.help" — код текстовки. Текстовки используются для локализации, подробно локализация описана в разделе [5.4.3](#).

В ТПО 7 версии в теге <help> возможно вывести минимальную и максимальную сумму платежа. Для этого используются атрибуты:

- 1) **\$min_sum_purchase** — минимальная сумма покупки (совпадает с минимальной суммой по сервису).
- 2) **\$max_sum_purchase** — максимальная сумма покупки (со своей комиссией не превышает максимальную сумму по сервису).

Пример:

```
<help>
  <![CDATA[Если хотите внести фиксированную сумму – укажите её.
           Можете оставить поле пустым.
           Минимальная сумма: $min_sum_purchase
           Максимальная сумма: $max_sum_purchase]]>
</help>
```

7.2 СЧИТЫВАНИЕ ДАННЫХ СО ШТРИХ-КОДОВ (<BARCODE-SCANNER>)

7.2.1 НАЗНАЧЕНИЕ, СТРУКТУРА <BARCODE-SCANNER>

Элемент `<barcode-scanner>` обрабатывает сигнал со сканера штрих-кода для использования считанных данных на экране. Для работы со сканером не требуется специального экрана и поля.

Структура элемента:

```
<barcode-scanner show-error=[true|false]
  on-success={refresh|next}>
  <barcode-parser type=[regex|block]>
  <rules validate=<string>>
    <rule key=<string>
      title=<string>
      regex=<notEmptyString>
      is-bind=[true|false]/>
    </rule>
  </rules>
  </barcode-parser>
</barcode-scanner>
```

Дочерние теги: `<barcode-parser>`.

Описание атрибутов элемента `<barcode-scanner>` приведено в таблице 7.2.1.1.

Таблица 7.2.1.1 — Описание атрибутов элемента `<barcode-scanner>`

Атрибут	Описание	Обяз.
show-error	Показывает диалог ошибки считывания штрих-кода, возможные значения:	Нет

Атрибут	Описание	Обяз.
	<ul style="list-style-type: none">• true — диалог отображается;• false — диалог не отображается, значение по умолчанию	
on-success	Тип события при успешном считывании данных со сканера штрих-кода	Нет

Доступны следующие события при успешном считывании данных со сканера штрих-кодов:

1. **next** — событие перехода на следующий экран.
2. **refresh** — событие обновления текущего экрана (есть возможность скорректировать отсканированные данные).

7.2.2 ПРАВИЛА РАСПОЗНАВАНИЯ ШТРИХ-КОДА (<BARCODE-PARSER>)

Элемент `<barcode-parser>` — определяет правила преобразования считанного штрих-кода для использования в качестве набора атрибутов платежа. Существуют два типа парсера, определяемые значением атрибута **type**:

1. **regex** — правила создания атрибутов определяются регулярным выражением.
2. **block** — позволяет формировать атрибуты платежа из части считанного значения.

Правила преобразования для элемента `<barcode-parser>` с типом **regex** определяются в дочерних элементах `<rules>`. В элементе `<rules>` может быть задан атрибут **validate**, который содержит регулярное выражение, определяющие допустимое значение штрих-кода. Например, если нужно работать только со штрих-кодами товаров, произведенных в России, требуется указать следующее значение **validate="^46\d.*"**, то есть все штрих-коды начинающиеся не с чисел 460-469 будут проигнорированы. Правила преобразования определяются в элементах `<rule>`. Структура элемента:

```
<barcode-parser type="regex" >
  <rules validate=<string>>
    <rule key=<string>
      title=<string>
      regex=<notEmptyString>
      is-bind=[true|false]
      convert-from=<integer>
      convert-to=<integer>
      flags=<flags_specification/>
    ...
  </rules>
</barcode-parser>
```

Для элемента `<barcode-parser>` добавлен атрибут **`disable-next=[true|false]`**, который обрабатывается только при использовании на экране «INFO». Если указан соответствующий флаг в сценарии/форме, то кнопка «Далее» на «INFO»-экране будет активна только после считывания баркода, при условии что **`on-success=«refresh»`**. На остальных типах экранов видимость кнопки регулируется валидацией полей ввода.

Описание атрибутов элемента `<rule>` приведено в таблице 7.2.2.1.

Таблица 7.2.2.1 — Описание атрибутов элемента `<rule>`

Атрибут	Описание	Обяз.
key	Ключ атрибута платежа, под которым будет сохранено значение полученного элемента	Да
title	Заголовок создаваемого атрибута платежа	Да
regex	Значение первой группы будет помещено в создаваемый атрибут платежа в качестве значения	Да
flags	Флаги, дополнительные свойства для получаемого атрибута платежа из данного поля, могут задаваться как числом (битовой маской), так и символьными константами. Описаны в разделе 5.4.2	Нет
is-bind	Определяет обязательность создания данного атрибута платежа. Возможные значения: <ul style="list-style-type: none">• true — создание элемента обязательно. Если элемент не был создан, выполнение сценария будет остановлено;• false — создание элемента необязательно. Выполнение сценария не будет остановлено, даже если элемент не был создан	Нет
convert-from	Целочисленное значение, определяющее основание системы счисления полученного значения	Нет
convert-to	Целочисленное значение, определяющее основание системы счисления для результирующего значения	Нет

Пример:

```
<forms>
  <form service-id="321">
    <!--Создание поля ввода, со следующими параметрами: активна цифровая
    клавиатура, максимальное количество вводимых клиентом символов 11.
    Название поля ввода: "Номер лицевого счета"-->
    <text-field name="id1" keyboard="Digital"
      max-len="11" message="Введите номер лицевого счета"
      title="Номер лицевого счета">
      <!--Регулярное выражение для валидации вводимого номера-->
      <verify> <rule regex="\d{6,11}$"/> </verify>
      <barcode-scanner>
        <barcode-parser type="regex" >
          <rules validate="^.*$" >
            <rule key="id1" title="Л/с" regex="^(.*)$" is-bind="true"/>
            <rule key="id2" title="Период" regex="^(.*)$" is-bind="true"/>
          </rules>
        </barcode-parser>
      </barcode-scanner>
    </text-field>
  </form>
</forms>
```

Элемент **<barcode-parser>** с типом **block** позволяет передавать в атрибуты платежа часть считанного значения. Структура:

```
<barcode-parser type="block" >
  <rules validate=<string>>
    <rule key=<string>
      title=<string>
      begin=<integer>
      end=<integer>
      flags=<flags_specification>/>
    </rule>
  </rules>
</barcode-parser>
```

Описание атрибутов элемента **<rule>** приведено в таблице 7.2.2.2.

Таблица 7.2.2.2 — Описание атрибутов элемента <rule>

Атрибут	Описание	Обяз.
key	Ключ атрибута платежа, под которым будет сохранено значение полученного элемента	Да
title	Заголовок создаваемого атрибута платежа	Да
begin	Целое число, определяет позицию первого символа выделяемой из штрих-кода подстроки	Да
end	Целое число, определяет позицию последнего символа выделяемой из штрих-кода подстроки, то есть символ, находящийся на позиции, указанной в параметре end , в подстроку не включается. Например, если указать end=4 , то в подстроку будет включен символ, находящийся на 3 позиции, на 4 — не будет включен	Да
flags	Флаги, дополнительные свойства для получаемого атрибута платежа из данного поля, могут задаваться как числом (битовой маской), так и символьными константами. Описаны в разделе 5.4.2	Нет

Пример:

```
<forms>
  <form service-id="321">
    <!--Создание поля ввода, со следующими параметрами: активна цифровая
    клавиатура, максимальное количество вводимых клиентом символов 11.
    Название поля ввода: "Номер лицевого счета"-->
    <text-field name="id1" keyboard="Digital"
      max-len="11" message="Введите номер лицевого счета"
      title="Номер лицевого счета">
    <!--Регулярное выражение для валидации вводимого номера-->
    <verify>
      <rule regex="^\d{6,11}$"/>
    </verify>
    <barcode-scanner>
      <barcode-parser type="block" >
        <rules>
```

```
<rule key="id1" title="Л/с" begin="1" end="11"/>  
<rule key="id2" title="Период" begin="12" end="13"/>  
</rules>  
</barcode-parser>  
</barcode-scanner>  
</text-field>  
</form>  
</forms>
```

7.3 ВКЛЮЧЕНИЕ ДАННЫХ ИЗ ВНЕШНЕГО ФАЙЛА (<XI:INCLUDE>)

Включение данных из внешнего файла в обеих версиях ТПО осуществляется с использованием элемента `<xi:include>`. В корневом теге файла форм (`<forms>` или `<form>`) необходимо подключить спецификацию `Xinclude`, указав конструкцию `xmlns:xi="http://www.w3.org/2001/Xinclude"`. Поиск файла-источника ведется в корневом каталоге ТПО. Файл-источник поддерживает те же правила локализации, что формы.

Пример:

```
<form xmlns:xi="http://www.w3.org/2001/XInclude" service-id="1152">
  <xi:include href="include.xml"/>
</form>
```

Содержимое файла `include.xml`:

```
<!--Создание поля ввода, со следующими параметрами: активна цифровая
клавиатура, максимальное количество вводимых клиентом символов 6.
Название поля ввода: "Текущее показание по тарифу 3"-->
<text-field name="tarif3" keyboard="any:[ru,symb]:upper:true"
  max-len="6" message="Введите текущее показание по тарифу 3"
  title="Текущее показание по тарифу 3">
  <!--Регулярное выражение для валидации вводимых данных-->
  <verify>
    <rule regex="^R[0-9]{12,12}$"/>
  </verify>
  <!--Подсказка, отображается на экране-->
  <help>
    Введите текущее показание по тарифу 3 (полупик)\n Внимание!
    В соответствии с Постановлением Правительства РФ № 354 от
    06.05.2011г.\n передача показаний осуществляется потребителем с
    15 по 26 число и учитывается в очередном счете за электроэнергию
  </help>
</text-field>
```

В примере файл-источник расположен в корневом каталоге ТПО.

Возможно указать полный путь до файла-источника и расположить его в другом каталоге, например:

```
<xi:include  
href="/home/user/Work/atm7/res/module/input/universal/include.xml"/>
```

7.4 НАСТРОЙКА ГОЛОСОВОГО СОПРОВОЖДЕНИЯ

В ТПО обеих версий поддерживается возможность голосового сопровождения шагов оплаты.

Воспроизводимые файлы должны располагаться в каталоге *<корень ТПО>/res/sound/* или в каталоге *<корень ТПО>/res/sound/<код локали>/* в 7 версии ТПО и *<корень ТПО>/sound/*, *<корень ТПО>/sound/<код локали>/* в 5 версии ТПО. Первоначально поиск файла осуществляется в каталоге *<корень ТПО>/res/sound/<код локали>/* в 7 версии ТПО и *<корень ТПО>/sound/<код локали>/* в 5 версии ТПО. В случае, если он оказывается не найден, то поиск осуществляется в папке *sound* по умолчанию.

Поддерживаются два типа воспроизведения файлов:

1. С начала до конца, не прерываются другими (название таких файлов должно начинаться с символа нижнего подчеркивания «_»).
2. С прерыванием (воспроизведение файла прерывает воспроизведение текущего).

Поддерживаются следующие стандартные позиции звуковых файлов:

1. *_click.mp3* — на нажатие любой кнопки.
2. *hello.mp3* — при выходе в меню или перерисовке меню (по умолчанию проигрывается 2 минуты).
3. *operator.mp3* — при выборе любой группы в меню.
4. *pay_data.mp3* — при открытии экрана оплаты.
5. *complete.mp3* — на финальном диалоге.

В формах в атрибуте **confirm-sound** элемента `<form>` можно указать звуковой файл, который будет воспроизведен при показе экрана подтверждения.

Пример:

```
<form service-id="192" confirm-sound="confirm_mobile.mp3">
  ...
</form>
```

Для озвучки экрана на формах можно указать файл в атрибуте **sound** этого экрана.

Пример:

```
<text-field name="id1" keyboard="Digital" max-len="10"
  example="123456789" message="Введите свой номер телефона"
  title="Телефон" sound="input_mobile.mp3">
  ...
</text-field>
```

Кроме того, возможна доработка ТПО для увеличения вариантов воспроизведения звука.

7.5 ИСПОЛЬЗОВАНИЕ ПЕРЕМЕННОЙ, ОБОЗНАЧАЮЩЕЙ КОРНЕВОЙ КАТАЛОГ, ПРИ УКАЗАНИИ ПУТЕЙ

При необходимости использования ссылок на внешние файлы в формах возможно использовать переменную `$terminal_home`, принимающую значение корневого каталога ТПО. Примером использования может служить вывод изображения в разделе с подсказкой.

Пример (5 версия ТПО):

```
<?xml version="1.0" encoding="UTF-8"?>
<forms xmlns="http://pay-logic.ru">
  <form service-id="1152">
    <info-field>
      <!--Подсказка, отображается на экране-->
      <help>
        <![CDATA[<html>Введите номер лицевого счета
          ]]>
      </help>
    </info-field>
  </form>
</forms>
```

Экран, соответствующий примеру, приведен на рисунке 7.5.1.

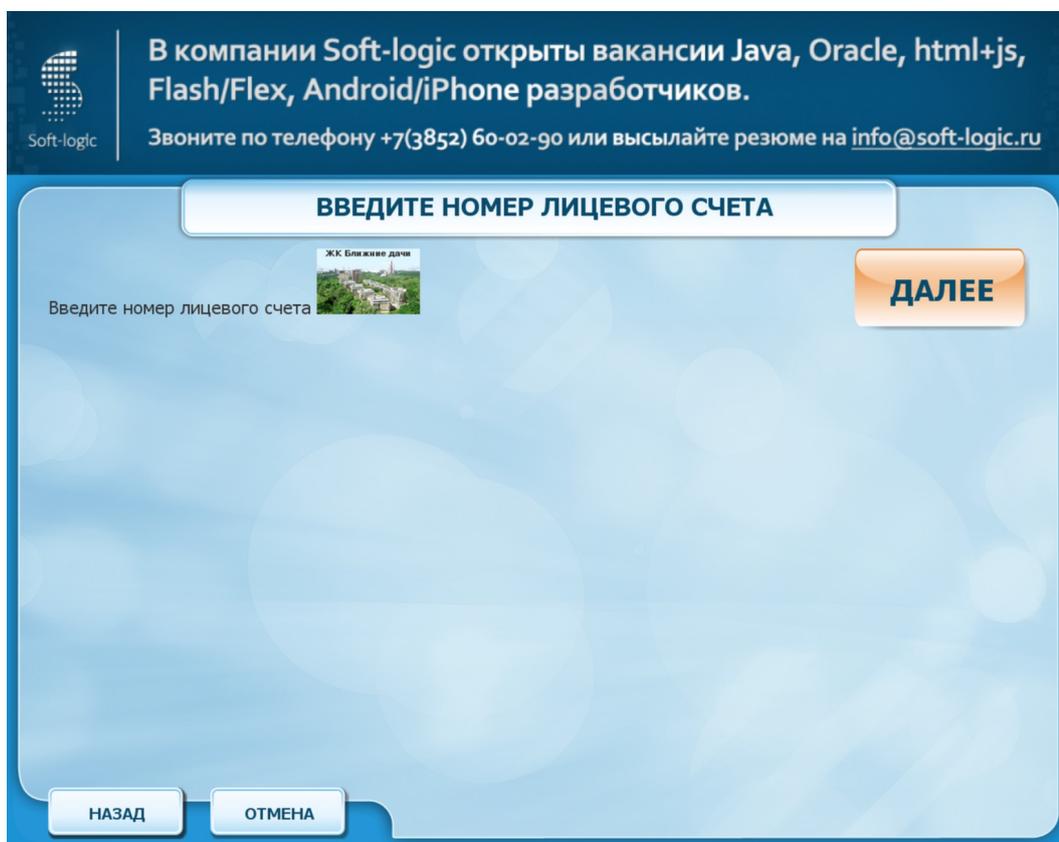


Рисунок 7.5.1 — Пример вывода изображения с использованием переменной, обозначающей корневой каталог ТПО

8 ЦЕПОЧКИ ЭКРАНОВ И ВЕТВЛЕНИЯ

Экраны, которые описаны выше, являются лишь блоками для построения форм оплаты. В файле формы можно использовать неограниченное число селекторов и группировать поля ввода. Древоподобная структура селекторов позволяет использовать выбор из нескольких списков значений, и однозначно определять значения атрибутов платежа по умолчанию. В таких случаях клиенту необязательно помнить коды, названия или номера для осуществления оплаты услуги, достаточно выбрать из списка необходимую опцию, что позволяет избежать многочисленных ошибок ввода данных при платеже.

Пример сложного ветвления с заданными условиями оплаты по сервису сбор платежей за государственный технический осмотр приводится в разделе [9](#), Задача 9.

9 ТИПОВЫЕ ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ ЯЗЫКА ОПИСАНИЯ ФОРМ

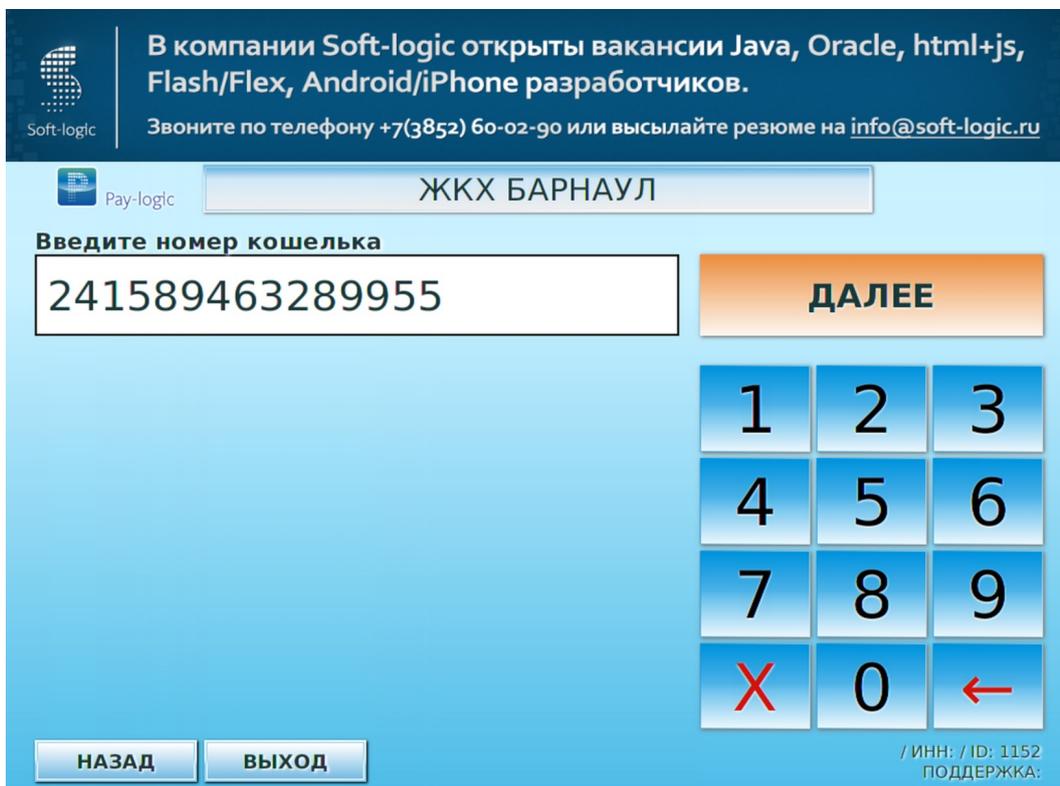
Задача 1

Создание сервиса, где пользователь должен ввести номер, состоящий из цифр. Допустимая длина введенных данных от 1 до 15 символов. Форматирование не требуется.

Решение:

```
<form service-id="28">
  <!--Создание поля ввода, со следующими параметрами: активна цифровая
  клавиатура, максимальное количество вводимых клиентом символов 15.
  Название поля ввода: "Номер кошелька"-->
  <text-field name="id1" keyboard="Digital" max-len="15"
    example="1234567890" message="Введите номер кошелька"
    title="Номер кошелька">
    <!--Регулярное выражение для валидации вводимого номера-->
    <verify> <rule regex="^[0-9]{1,15}$"/> </verify>
  </text-field>
</form>
```

Экран, соответствующий данной форме, приведен на рисунке 9.1.



В компании Soft-logic открыты вакансии Java, Oracle, html+js, Flash/Flex, Android/iPhone разработчиков.
Звоните по телефону +7(3852) 60-02-90 или высылайте резюме на info@soft-logic.ru

Pay-logic ЖКХ БАРНАУЛ

Введите номер кошелька

241589463289955

ДАЛЕЕ

1	2	3
4	5	6
7	8	9
X	0	←

НАЗАД Выход

/ ИНН: / ID: 1152
ПОДДЕРЖКА:

Рисунок 9.1 — Экран для формы, предусматривающей ввод номера

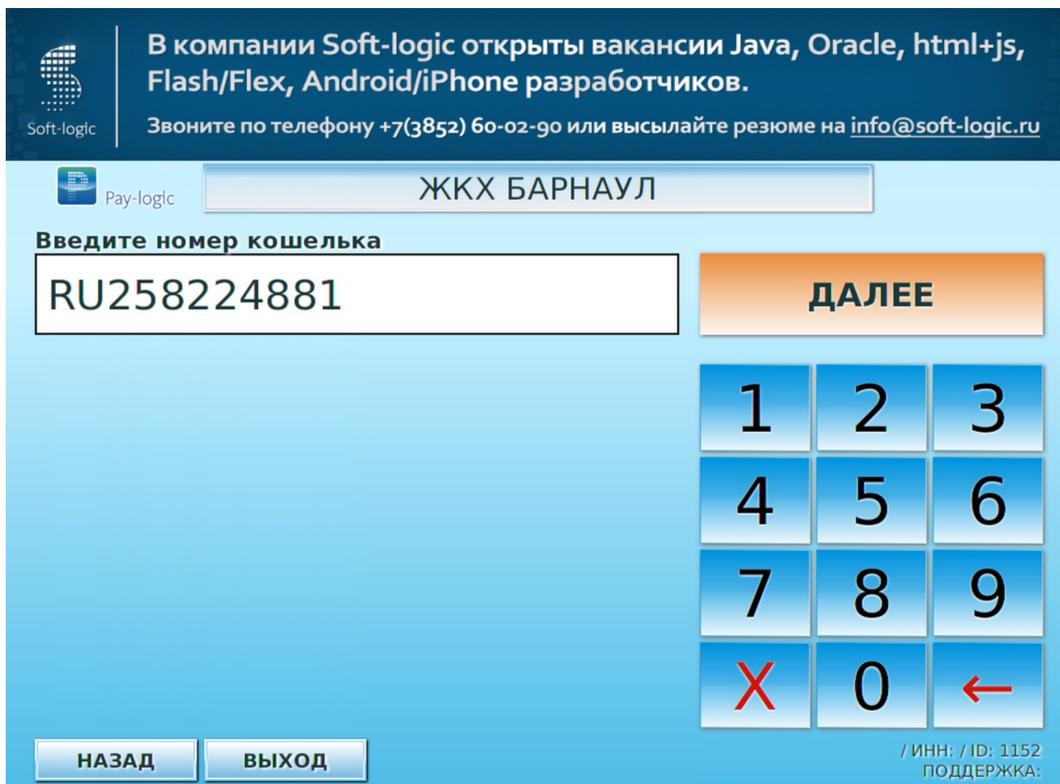
Задача 2

Создание сервиса, где пользователь должен ввести номер, состоящий из цифр. Номер имеет префикс RU. Допустимая длина введенных данных от 7 до 11 символов. Длина задается с учетом префикса и постфикса. Форматирование не требуется.

Решение:

```
<form service-id="29">
  <!--Создание поля ввода, со следующими параметрами: активна
цифровая клавиатура, максимальное количество вводимых клиентом символов
11. Название поля ввода: "Номер кошелька"-->
  <text-field name="id1" keyboard="Digital" max-len="11"
    prefix="RU" example="RU123456789"
    message="Введите номер кошелька"
    title="Номер кошелька">
  <!--Регулярное выражение для валидации вводимого номера-->
  <verify> <rule regex="^RU[0-9]{5,9}$"/> </verify>
</text-field>
</form>
```

Экран, соответствующий данной форме, приведен на рисунке 9.2.



В компании Soft-logic открыты вакансии Java, Oracle, html+js, Flash/Flex, Android/iPhone разработчиков.
Звоните по телефону +7(3852) 60-02-90 или высылайте резюме на info@soft-logic.ru

Pay-logic ЖКХ БАРНАУЛ

Введите номер кошелька

RU258224881

ДАЛЕЕ

1	2	3
4	5	6
7	8	9
X	0	←

НАЗАД Выход

/ ИНН: / ID: 1152
ПОДДЕРЖКА:

Рисунок 9.2 — Экран для формы, предусматривающей ввод номера с префиксом RU

Задача 3

Создание сервиса, где пользователь должен ввести номер кошелька, состоящий из цифр 12 и имеющего префикс R. После этого пользователь должен ввести номер мобильного телефона, состоящий из 10 цифр. Для лучшего восприятия номера должны быть отформатированы.

Решение:

```
<form service-id="25">
  <!--Создание поля ввода, со следующими параметрами: активна цифровая
  клавиатура, максимальное количество вводимых символов 13. Название поля
  ввода: "Номер кошелька"-->
  <text-field name="id1"
    keyboard="Digital"
    max-len="13"
    prefix="R"
    example="R123456789012"
    message="Введите номер кошелька"
    title="Номер кошелька">
    <verify> <rule regex="^R[0-9]{12,12}$"/> </verify>
    <!--Маска для форматирования вводимого номера, применяется по
    умолчанию-->
    <split default="*-****-****-****"/>
  </text-field>
  <!--Создание поля ввода, со следующими параметрами: активна цифровая
  клавиатура, максимальное количество вводимых символов 10. Название поля
  ввода: "Номер телефона"-->
  <text-field name="id2"
    keyboard="Digital"
    max-len="10"
    example="903333333"
    message="Введите номер своего сотового телефона"
    title="Номер телефона">
    <verify> <rule regex="^\d{10}$"/> </verify>
    <!--Маска для форматирования вводимого номера, применяется по
    умолчанию-->
    <split default="8-***-***-****"/>
  </text-field>
</form>
```

Экраны, соответствующие данной форме, приведены на рисунке 9.3.

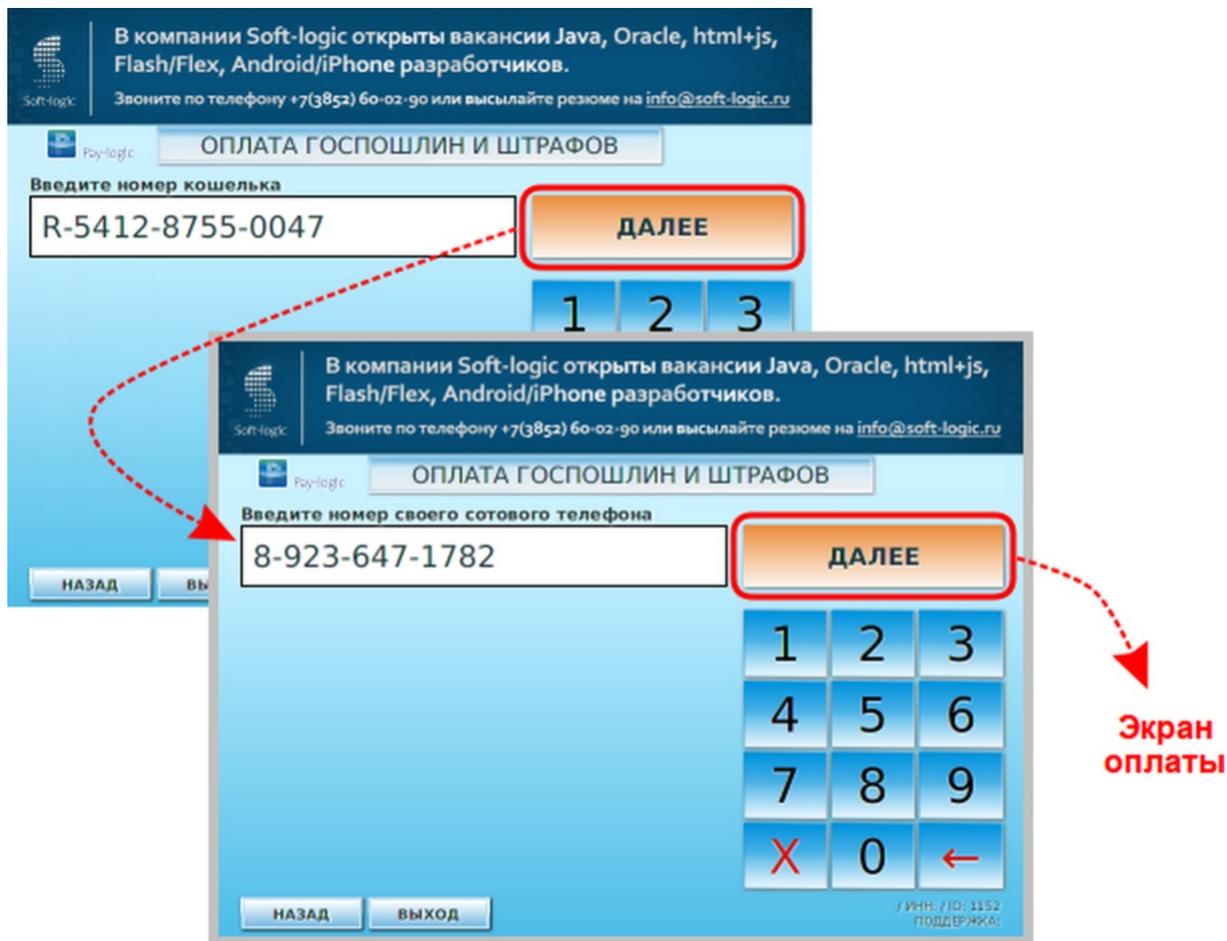


Рисунок 9.3 — Ввод данных для оплаты по сервису

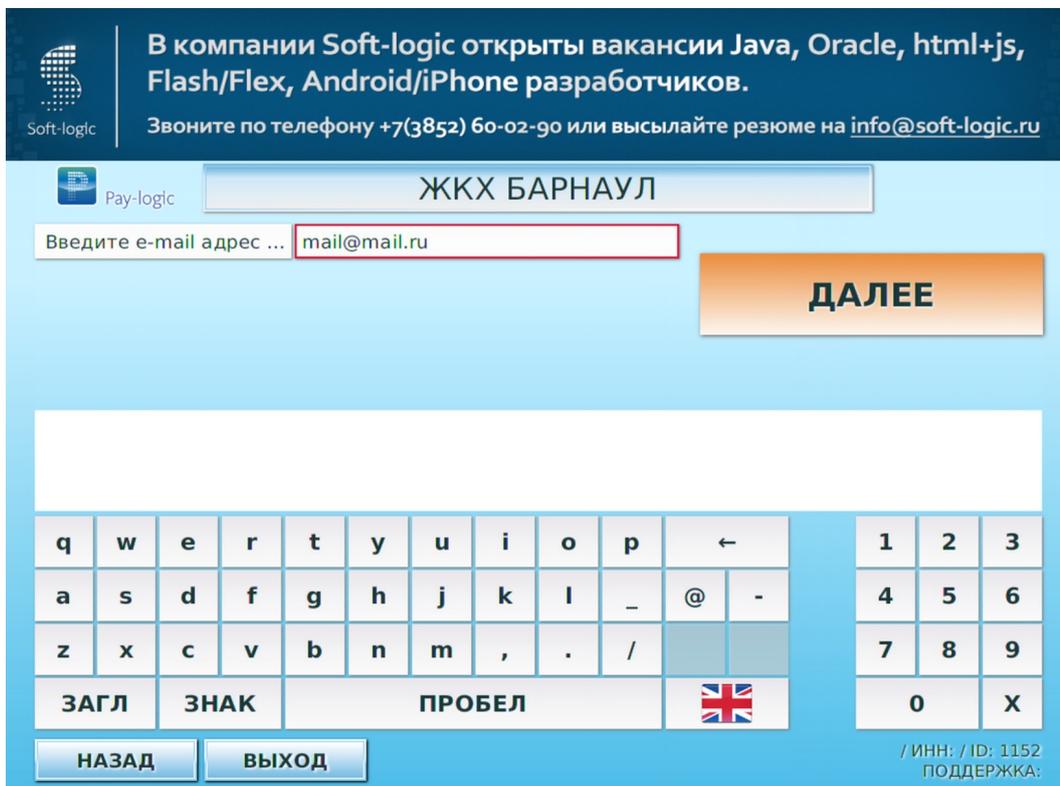
Задача 4

Создание сервиса, где пользователь должен ввести адрес электронной почты.

Решение:

```
<form service-id="31">
  <!--Создание поля ввода, со следующими параметрами: активна цифровая,
  символьная и буквенная клавиатура с возможностью переключения языка,
  максимальное количество вводимых клиентом символов 40. Название поля
  ввода: "E-mail адрес кошелька"-->
  <text-field name="id1"
    keyboard="any:[ru,en,symb]:upper:true"
    max-len="40" example="example@mail.ru"
    message="Введите e-mail адрес кошелька"
    title="E-mail адрес кошелька">
    <verify>
      <rule regex="^[a-z][\w.-]+@\w[\w.-]+\.[\w.-]*[a-z][a-z]$"/>
    </verify>
  </text-field>
</form>
```

Экран, соответствующий данной форме, приведен на рисунке 9.4.



Soft-logic

В компании Soft-logic открыты вакансии Java, Oracle, html+js, Flash/Flex, Android/iPhone разработчиков.
Звоните по телефону +7(3852) 60-02-90 или высылайте резюме на info@soft-logic.ru

Pay-logic

ЖКХ БАРНАУЛ

Введите e-mail адрес ...

ДАЛЕЕ

q	w	e	r	t	y	u	i	o	p	←		
a	s	d	f	g	h	j	k	l	_	@	-	
z	x	c	v	b	n	m	,	.	/			
ЗАГЛ	ЗНАК	ПРОБЕЛ										

НАЗАД **ВЫХОД**

/ ИНН: / ID: 1152
ПОДДЕРЖКА:

Рисунок 9.4 — Экран для формы, предусматривающей ввод адреса электронной почты

Задача 5

Создание сервиса, где пользователь на первом шаге должен указать тип услуги, а затем ввести номер. Экран выбора типа услуги должен представлять собой кнопки с изображениями.

Решение:

```
<form service-id="157">
  <!--Создание экрана выбора пакета каналов-->
  <image-selector name="id2" decor="cards"
    message="Выберите пакет каналов">
    <item img="tricolor-co.png"
      text="Центр Оптимум"
      value="1" sum="600.00"/>
    <item img="tricolor-cn.png"
      text="Центр Ночной"
      value="2" sum="500.00"/>
  </image-selector>
  <!--Создание поля ввода, со следующими параметрами: активна
цифровая клавиатура, максимальное количество вводимых клиентом символов
12. Название поля ввода: "ID приемника"-->
  <text-field name="id1"
    keyboard="Digital"
    max-len="12"
    example="111122223333"
    message="Введите ID приемника"
    title="ID приемника">
  <verify>
    <rule regex="^\d{12,12}$"/>
  </verify>
  <!--Маска для форматирования вводимых данных, применяется по
умолчанию-->
  <split default="****-****-****"/>
  </text-field>
</form>
```

Экран, соответствующий селектору, приведен на рисунке 9.5.

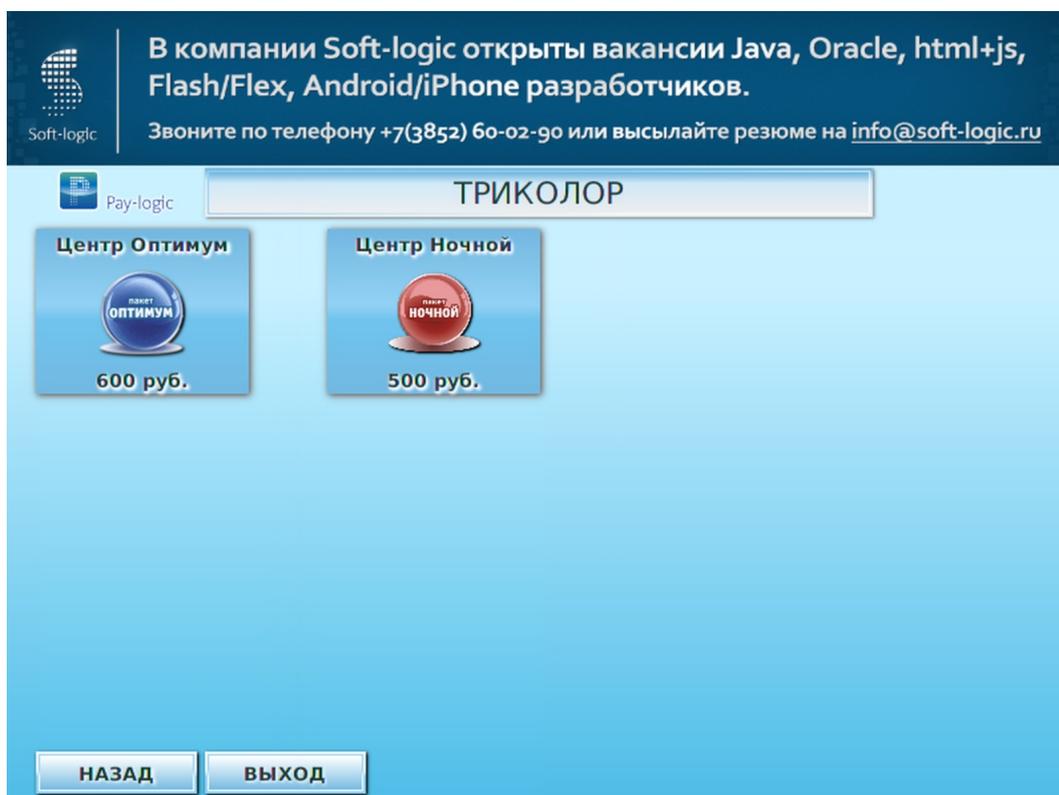


Рисунок 9.5 — Экран выбора типа услуги

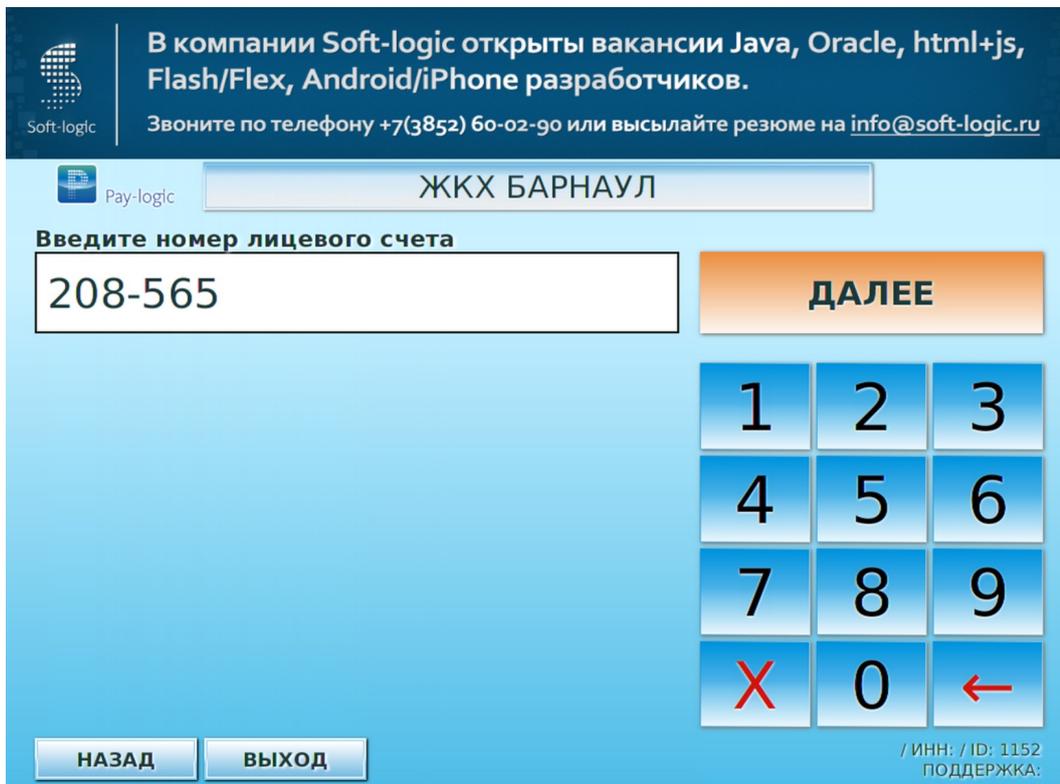
Задача 6

Допускается ввод 6-тизначного либо 12-тизначного номера. Оба варианта номеров должны быть отформатированы для лучшего восприятия.

Решение:

```
<form service-id="35">
  <!--Создание поля ввода, со следующими параметрами: активна цифровая
  клавиатура, максимальное количество вводимых клиентом символов 12.
  Название поля ввода: "Номер лицевого счета"-->
  <text-field name="id1" keyboard="Digital" max-len="12"
    message="Введите номер лицевого счета"
    title="Номер лицевого счета">
    <verify> <rule regex="^(\\d{6}|\\d{12})$"/> </verify>
    <!--Маска для форматирования вводимых данных, применяется по
    умолчанию-->
    <split default="***-***">
      <rule regex="^[0-9]{7,12}$" value="***-***-***-***"/>
    </split>
  </text-field>
</form>
```

Экран, соответствующий данной форме, приведен на рисунке 9.6.



В компании Soft-logic открыты вакансии Java, Oracle, html+js, Flash/Flex, Android/iPhone разработчиков.
Звоните по телефону +7(3852) 60-02-90 или высылайте резюме на info@soft-logic.ru

Pay-logic ЖКХ БАРНАУЛ

Введите номер лицевого счета

208-565

ДАЛЕЕ

1	2	3
4	5	6
7	8	9
X	0	←

НАЗАД Выход

/ ИНН: / ID: 1152
ПОДДЕРЖКА:

Рисунок 9.6 — Экран для формы, предусматривающей ввод лицевого счета длиной 6 или 12 цифр

Задача 7

На первом экране пользователь должен выбрать тип оплачиваемой услуги. На втором экране должен быть введен номер счета. При этом, если пользователь выбрал первый тип услуги, то вводимые данные должны удовлетворять одним условиям, если выбрал второй тип — другим.

Решение:

Необходимо воспользоваться функционалом ветвлений.

```
<form service-id="99">
  <!--Создание экрана выбора назначения платежа-->
  <text-selector name="type" message="Выберите назначение платежа">
    <item text="Интернет" value="1">
      <!--Создание поля ввода, со следующими параметрами: активна
цифровая клавиатура, максимальное количество вводимых клиентом символов
5. Название поля ввода: "Номер счета"-->
      <text-field name="id1"
        keyboard="Digital"
        max-len="5"
        example="55123"
        message="Введите номер счета"
        title="Номер счета">
        <verify>
          <rule regex="^\d{5,5}$"/>
        </verify>
        <!--Маска для форматирования вводимого номера, применяется по
умолчанию-->
        <split default="И-*****"/>
      </text-field>
    </item>
    <item text="ТВ" value="2">
      <!--Создание поля ввода, со следующими параметрами: активна
цифровая клавиатура, максимальное количество вводимых клиентом символов
12. Название поля ввода: "Номер лицевого счета"-->
      <text-field name="id1"
        keyboard="Digital"
        max-len="5"
        example="15523"
```

```
message="Введите номер счета"  
title="Номер счета">  
<verify>  
  <rule regex="^\d{5,5}$"/>  
</verify>  
<!--Маска для форматирования вводимого номера, применяется по  
умолчанию-->  
  <split default="К-*****"/>  
</text-field>  
</item>  
</text-selector>  
</form>
```

Схема смены экранов, соответствующая данному примеру приведена на рисунке ниже.

**Внимание!**

При выборе каждого из элементов селектора начнет выполняться отдельная последовательность экранов.

Задача 8

Прием штрафов ГИБДД. Необходимо показать предупреждение, после чего организовать ввод ФИО, адреса, номера штрафной квитанции. Пользователь должен сам ввести сумму штрафа. Пользователь должен выбрать статью, по которой выписан штраф, в базу данных процессингового центра должен попасть ОКАТО, который соответствует статье нарушений.

Решение:

```
<form service-id="206">
  <info-field>
    <![CDATA[<html>Уважаемый плательщик! Для оплаты штрафа ГИБДД,
      законодательство требует указания фамилии и адреса прописки
      плательщика, которые следует ввести на следующих экранах.
      Например:<br> ФИО: Иванов Иван Иванович<br>Название
      населенного пункта: г. Барнаул <br>Адрес: ул.
      Ленина 15 кв 25]]>
  </info-field>
  <!--Создание поля ввода, со следующими параметрами: активна цифровая,
  символьная и буквенная клавиатура, язык ввода - русский без возможности
  переключения языка, максимальное количество вводимых клиентом символов
  80. Название поля ввода: "Фамилия и инициалы"-->
  <text-field name="id3" keyboard="any:[ru,symb]:upper:true"
    max-len="80" example="Петров Петр Петрович"
    message="Введите фамилию и инициалы"
    title="Фамилия и инициалы">
    <verify>
      <rule regex="^[А-Яа-яёЁ]{3,}(\s+[А-Яа-яёЁ]{1,}\.?)\{0,2\}$"/>
    </verify>
  </text-field>
  <!--Создание поля ввода, со следующими параметрами: активна цифровая,
  символьная и буквенная клавиатура, язык ввода - русский без возможности
  переключения языка, максимальное количество вводимых клиентом символов
  80. Название поля ввода: "Название населенного пункта"-->
  <text-field name="id4" keyboard="any:[ru,symb]:upper:true"
    max-len="80" example="г. Барнаул" default="г. Барнаул"
    message="Введите название населенного пункта"
    title="Название населенного пункта">
    <verify>
```

```
<rule regex="^[А-Яа-яёЁ\d .-]{5,}$"/>
</verify>
</text-field>
<!--Создание поля ввода, со следующими параметрами: активна цифровая,
символьная и буквенная клавиатура, язык ввода - русский без возможности
переключения языка, максимальное количество вводимых клиентом символов
80. Название поля ввода: "Адрес"-->
<text-field name="id5" keyboard="any:[ru,symb]:upper:true"
max-len="80" example="ул. 10-я Западная 5 кв 1"
message="Введите адрес" title="Адрес" >
<verify>
<rule regex="^([А-Яа-яёЁ\d-]{1,}\.?) (\s*[А-Яа-яёЁ\d-]{1,}\.?)*$"/>
</verify>
</text-field>
<!--Создание экрана выбора статьи КОАП-->
<text-selector name="id2" message="Выберите статью КОАП, по которой
произведено нарушение">
<item text="Все статьи главы 12 КОАП" value="18811630000010000140"/>
<item text="Статьи 8.22, 8.23 КОАП" value="18811625050010000140"/>
<item text="Статьи 11.25-11.29 КОАП" value="18811629000010000140"/>
<item text="Статьи 19.20-11.24 и 20.15-20.25 КОАП"
value="18811690040040000140"/>
</text-selector>
<info-field>
<![CDATA[<html>Уважаемый плательщик! На следующем экране необходимо
ввести номер штрафной квитанции. Номер расположен в самом
верху штрафной квитанции.]]>
</info-field>
<!--Создание поля ввода, со следующими параметрами: активна цифровая,
символьная и буквенная клавиатура, язык ввода - русский без возможности
переключения языка, максимальное количество вводимых клиентом символов
80. Название поля ввода: "Номер штрафной квитанции"-->
<text-field name="id1" keyboard="any:[ru,symb]:upper:true"
max-len="10" prefix="22" example="49-КА-511122"
message="Введите номер штрафной квитанции"
title="Номер штрафной квитанции">
<verify> <rule regex="^22[А-Яа-яёЁ]{2}\d{6}$"/> </verify>
<!--Маска для форматирования вводимого номера, применяется по
умолчанию-->
<split default="**-**-*****"/>
</text-field>
</info-field>
```

```
<![CDATA[<html> На следующем экране вам необходимо ввести сумму  
штрафа. Это необходимо для того, чтобы программа правильно  
рассчитала сумму к внесению.<br> <b>Внимание! Если сумма  
внесенных средств будет больше необходимой суммы, то Вы  
получите "Карту сдачи". "Карта сдачи" - это чек с паролем,  
который вы можете использовать для оплаты услуг через  
торговый автомат нашей системы. Инструкция по ее  
использованию будет распечатана на чеке.]]>  
</info-field>  
<numeric-field name="sum" message="Введите сумму штрафа"  
allow-decimal="false" min="100" max="5000"  
change-sumpurchase="true"/>  
</form>
```

На рисунке 9.7 отображен экран выбора статьи КОАП, по которой произведено нарушение. Пользователь выбирает статьи 11.25-11.29 КОАП. В параметрах платежа должно быть установлено значение КОАП 18811629000010000140.

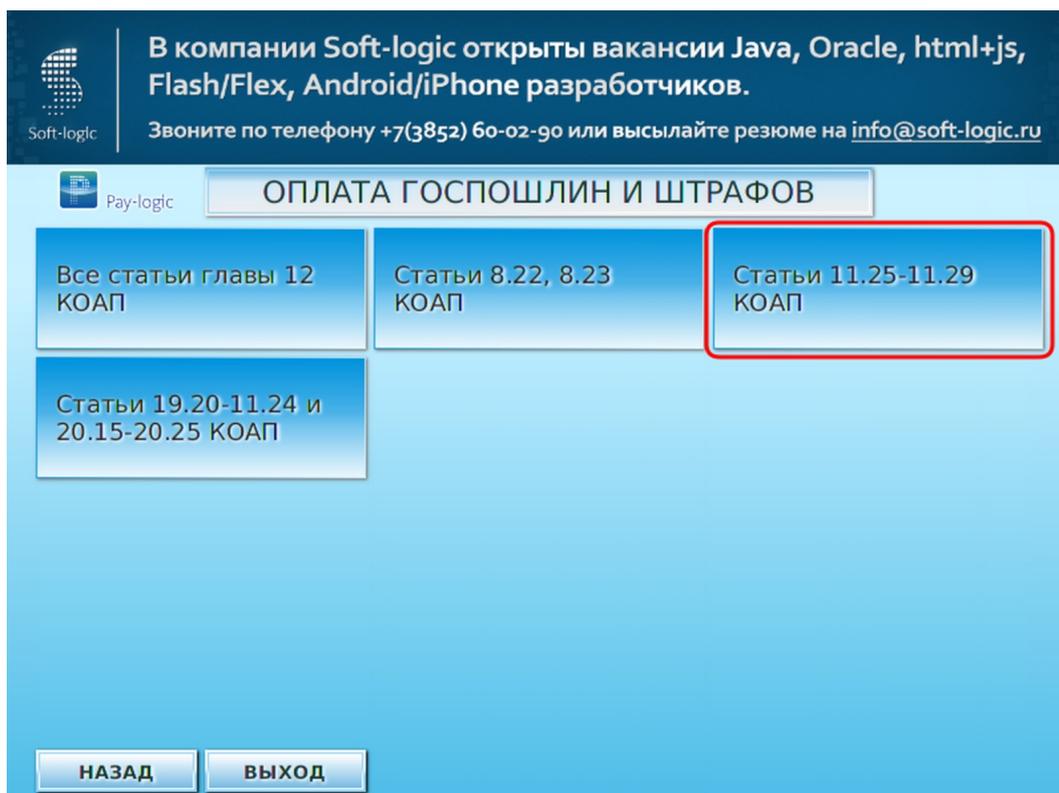


Рисунок 9.7 — Выбор статьи, по которой производится оплата

При просмотре подробной информации о платеже (рисунок 9.8) видно, что атрибуту **id2** установлено значение **18811629000010000140**.

"Общая информация"

ID операции: 10599045
Номер чека: 7504
Введите фамилию и инициалы: ИВАНОВ СЕМЕН ФЕДОРОВИЧ
Введите название населенного пункта: г. Барнаул
Введите адрес: ул. 10-я Западная 10-56
Пользователь: 22KA585222
id2: 18811629000010000140
Время терминала: 26.10.2015 15:47:29
Время сервера: 26.10.2015 15:48:27
Время обработки: 26.10.2015 15:48:28
Статус: Проведение ([Сброс провайдера](#))
Субстатус: Провайдер определен
Сгенерированные проводки: [Просмотр](#)
Fraud: [Добавить в черный список](#) | [Добавить в белый список](#)

Рисунок 9.8 — Просмотр подробной информации о платеже

Задача 9

Организовать сбор платежей за государственный технический осмотр. Стоимость технического осмотра зависит от типа транспортного средства. Провайдер предоставил таблицу платежей.

Решение:

```
<form service-id="209">
  <info-field>
    <![CDATA[<html>Уважаемый плательщик! Для оплаты за технический
      осмотр, законодательство требует указания фамилии и адреса
      прописки плательщика, которые следует ввести на следующих
      экранах. Например:<br> ФИО: Иванов И И<br>Название
      населенного пункта: г. Барнаул <br>Адрес: ул. Ленина 15 кв
      25]]>
  </info-field>
  <!--Создание поля ввода, со следующими параметрами: активна цифровая,
  символьная и буквенная клавиатура, язык ввода - русский без возможности
  переключения языка, максимальное количество вводимых клиентом символов
  80. Название поля ввода: "Фамилия"-->
  <text-field name="idl" keyboard="any:[ru,symb]:upper:true"
    max-len="40" example="Иванов Иван Иванов"
    message="Введите фамилию полностью"
    title="Фамилия">
    <verify>
      <rule regex="^[А-Яа-яёЁ\d]{1,}(\s+[А-Яа-яёЁ\d]{1,}\.?)*$"/>
    </verify>
  </text-field>
  <!--Создание поля ввода, со следующими параметрами: активна цифровая,
  символьная и буквенная клавиатура, язык ввода - русский без возможности
  переключения языка, максимальное количество вводимых клиентом символов
  80. Название поля ввода: "Название населенного пункта"-->
  <text-field name="city" keyboard="any:[ru,symb]:upper:true"
    max-len="80" example="г. Барнаул" default="г. Барнаул"
    message="Введите название населенного пункта"
    title="Название населенного пункта">
    <verify>
      <rule regex="^[А-Яа-яёЁ\d .-]{5,}$"/>
    </verify>
</form>
```

```
</text-field>
<!--Создание поля ввода, со следующими параметрами: активна цифровая,
символьная и буквенная клавиатура, язык ввода - русский без возможности
переключения языка, максимальное количество вводимых клиентом символов
80. Название поля ввода: "Адрес"-->
  <text-field name="address" keyboard="any:[ru,symb]:upper:true"
    max-len="80" example="ул. 10-я Западная 5 кв 1"
    message="Введите адрес" title="Адрес">

  <verify>
  <rule regex="^([А-Яа-яёЁ\d-]{1,}\.?) (\s*[А-Яа-яёЁ\d-]{1,}\.?) *$"/>
  </verify>
</text-field>
<!--Создание экрана выбора транспортного средства-->
<text-selector name="id4" message="Транспортное средство">
  <item text="Легковой автомобиль" value="1"/>
  <item text="Автобус" value="2">
<!--Создание экрана выбора массы автобуса-->
  <text-selector name="id5" message="Выберите массу автобуса">
    <item text="полной массой до 5 тонн включительно"
      value="2.1" sum="391.76"/>
    <item text="полной массой свыше 5 тонн"
      value="2.2" sum="464.92"/>
  </text-selector>
</item>
<item text="Грузовой автомобиль" value="3">
  <!--Создание экрана выбора массы грузового автомобиля для типа ТС
"Грузовой автомобиль"-->
  <text-selector name="id5"
    message="Выберите массу грузового автомобиля">
    <item text="полной массой до тонн включительно" value="3.1">
      <!--Создание экрана выбора типа перевозимого груза-->
      <text-selector name="id6"
        message="Выберите тип перевозимого груза">
        <item text="Обычный груз" value="3.1.1" sum="341.02"/>
        <item text="Опасный груз" value="3.1.2" sum="443.33"/>
        <item text="Негабаритный груз" value="3.1.3" sum="409.22"/>
      </text-selector>
    </item>
    <item text="полной массой от 3.5 до 12 тонн включительно"
      value="3.2">
      <!--Создание экрана выбора типа перевозимого груза для типа ТС
"полной массой от 3.5 до 12 тонн включительно"-->
      <text-selector name="id6"
```

```
        message="Выберите тип перевозимого груза">
        <item text="Обычный груз" value="3.2.1" sum="456.66"/>
        <item text="Опасный груз" value="3.3.2" sum="593.66"/>
        <item text="Негабаритный груз" value="3.2.3" sum="547.99"/>
    </text-selector>
</item>
<item text="полной массой свыше 12 тонн" value="3.3">
    <!--Создание экрана выбора типа перевозимого груза для типа ТС
"олной массой свыше 12 тонн"-->
    <text-selector name="id6"
        message="Выберите тип перевозимого груза">
        <item text="Обычный груз" value="3.3.1" sum="484.98"/>
        <item text="Опасный груз" value="3.3.2" sum="630.47"/>
        <item text="Негабаритный груз" value="3.3.3" sum="581.98"/>
    </text-selector>
    </item>
</text-selector>
</item>
<item text="Полуприцеп" value="4" sum="300.9"
    <!--Создание экрана выбора типа перевозимого груза для типа ТС
"Полуприцеп"-->
    <text-selector name="id6" message="Выберите тип перевозимого груза">
        <item text="Обычный груз" value="4.1" sum="300.90"/>
        <item text="Опасный груз" value="4.2" sum="391.17"/>
        <item text="Негабаритный груз" value="4.3" sum="361.08"/>
    </text-selector>
    </item>
<item text="Прицеп" value="5">
    <!--Создание экрана выбора типа перевозимого груза для типа ТС
"Прицеп"-->
    <text-selector name="id5" message="Выберите массу прицепа">
        <item text="полной массой до 0.75 тонн включительно" value="5.1">
            <text-selector name="id6"
                message="Выберите тип перевозимого груза">
                <item text="Обычный груз" value="5.1.1" sum="107.38"/>
                <item text="Опасный груз" value="5.1.2" sum="139.59"/>
                <item text="Негабаритный груз" value="5.1.3" sum="128.86"/>
            </text-selector>
            </item>
        <item text="полной массой от 0.75 до 3.5 тонн включительно"
            value="5.2">
            <!--Создание экрана выбора типа перевозимого груза-->
            <text-selector name="id6"
```

```
        message="Выберите тип перевозимого груза">
        <item text="Обычный груз" value="5.2.1" sum="167.56"/>
        <item text="Опасный груз" value="5.2.2" sum="217.83"/>
        <item text="Негабаритный груз" value="5.2.3" sum="201.07"/>
        </text-selector>
    </item>
    <item text="полной массой свыше 3.5 тонн" value="5.3">
        <!--Создание экрана выбора типа перевозимого груза для ТС "полной
массой свыше 3.5 тонн"-->
        <text-selector name="id6"
            message="Выберите тип перевозимого груза">
            <item text="Обычный груз" value="5.3.1" sum="205.32"/>
            <item text="Опасный груз" value="5.3.2" sum="266.92"/>
            <item text="Негабаритный груз" value="5.3.3" sum="246.38"/>
            </text-selector>
        </item>
    </text-selector>
</item>
    <item text="Мототранспортное средство" value="6" sum="108.56"/>
</text-selector>
</form>
```

Размер сбора зависит от типа транспортного средства и его параметров. Например, для грузового автомобиля полной массой свыше 12 тонн, перевозящего обычный груз, размер сбора составит 484,98 руб., комиссия 3 руб., общая сумма оплаты 487,98 руб. (рисунок 9.9).

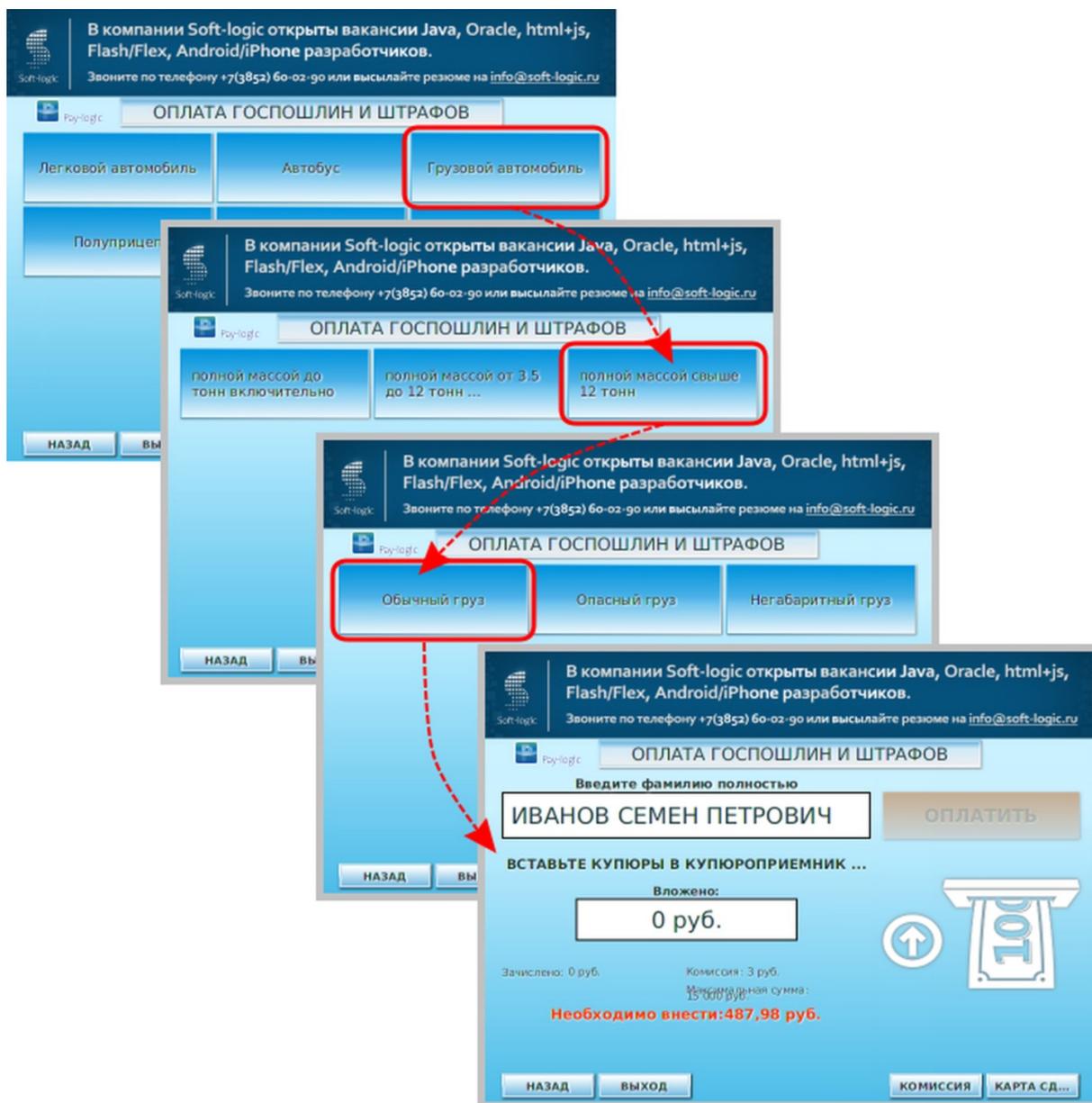


Рисунок 9.9 — Выбор транспортного средства и параметров для оплаты сбора

10 ОТЛИЧИЯ ТПО 5 И 7 ВЕРСИИ

Существуют следующие отличия между 5 и 7 версиями ТПО:

1. В системе обновлений формы располагаются в каталогах **<корень ТПО>/resources/forms/** для 5 версии ТПО и **<корень ТПО>/res/module/input/universal/** для 7 версии ТПО.
2. Ресурсные файлы с текстовками для ТПО 7 расположены в **<корень ТПО>/res/module/i18n/**, для ТПО 5 — **<корень ТПО>/resources/i18n/**.
3. В атрибутах **text** элемента **<checkbox-field>**, **text** элемента **<item>** текстового селектора для 5 версии ТПО допустимо использовать теги языка разметки html для форматирования выводимого сообщения. Например, **"<html>полной массой до 0.75
 тонн включительно"**. В 7 версии — недопустимо.
4. В 7 версии ТПО для перенаправления на другой сервис используется атрибут **redirect** элемента **<item>**. В 5 версии ТПО используется атрибут **redirect-params="..."** в элемента **<form>**.
5. В 7 версии ТПО в элементе **<help>** возможно указать тип точки. Например, **<help types="4">** — для электронного кошелька. Если не будет найдена подсказка для типа точки, то будет использована подсказка по умолчанию. В 5 версии ТПО данный функционал не поддерживается.
6. Воспроизведение звуков при открытии экранов ввода реализовано только в 5 версии ТПО.

11 ОСОБЕННОСТИ СИНТАКСИСА ДЛЯ ПРИЛОЖЕНИЯ «ПРИЕМ ПЛАТЕЖЕЙ ANDROID»

Для универсального обработчика реализованы поля ввода `<text-field>` (раздел [6.1.1](#)), `<checkbox-field>` (раздел [6.3](#)), `<text-selector>` (раздел [6.4.2](#)), `<date-field>` (раздел [6.5](#)), `<set-field>` (раздел [6.6](#)).

Если значение поля не соответствует правилам валидации (раздел [6.1.2](#)), то отображается содержимое элемента `<help>` (раздел [7.1](#)). Аналогично для `<checkbox-field>` в зависимости от значения атрибута **required** отображается содержимое элемента `<help>` или `<help-off>`.

В системе обновлений формы должны размещаться в каталоге `<корень CO>/android_im7/forms/universal`.

ПРИЛОЖЕНИЕ А. ПРИМЕР ФОРМЫ С ВЛОЖЕННЫМ СЕЛЕКТОРОМ

В примере осуществляется выбор типа страхования: **добровольное** или **обязательное**.

Внутри типа «**Добровольное** страхование» возможен выбор: страхования жизни; страхования гражданской ответственности; ипотечное кредитование, лизинг.

Внутри типа «**Обязательное** страхование» возможен выбор: страхования жизни; страхования гражданской ответственности для каждого из которых возможен выбор страхования транспорта или имущества.

После выбора типа «**Добровольного** страхования» осуществляется выбор серии сертификата и затем вводится номер и другие реквизиты: ФИО, E-mail, Номер телефона.

После выбора типа «**Обязательного** страхования» осуществляется выбор «По номеру бланка» или «По номеру сертификата» и затем вводится номер бланка или сертификата и другие реквизиты: ФИО, E-mail, Номер телефона.

```
<?xml version="1.0" encoding="UTF-8"?>
<forms>
  <form service-id="229">
    <!--Создание экрана выбора типа страхования-->
    <text-selector name="mcp_type" message="Страхование" title="Страхование">
      <item text="Добровольное страхование" value="Добровольное страхование">
        <text-selector name="mcp_type2" message="Страхование" title="Страхование">
          <item text="Страхование жизни" value="Страхование жизни">
            <text-selector name="mcp_id2" message="Выберите серию сертификатов"
              title="Серия сертификата">
              <item text="I-MIC" value="I-MIC"/>
              <item text="I-PI" value="I-PI"/>
              <item text="TI (52382)" value="TI (52382)"/>
              <item text="I-PA" value="I-PA"/>
              <item text="I-GL" value="I-GL"/>
              <item text="I-CG" value="I-CG"/>
            </text-selector>
          </item>
        </text-selector>
      </item>
    </text-selector>
  </form>
</forms>
```

```
<!--Создание поля ввода, со следующими параметрами: активна цифровая
клавиатура, максимальное количество вводимых символов 9. Название поля ввода:
"Номер сертификата"-->
<text-field name="idl" keyboard="Digital" max-len="9"
            message="Введите номер сертификата"
            title="Номер сертификата">
    <!--Регулярное выражение для валидации вводимого номера-->
    <verify> <rule regex="^[0-9/]{9}$"/> </verify>
    <!--Подсказка, отображается на терминале-->
    <help> 9 символов. Номер: 123456/15 или 000001/15 </help>
</text-field>
</item>
<item text="Страхование гражданской ответственности "
      value="Страхование гражданской ответственности ">
    <text-selector name="mcp_id2" message="Выберите серию сертификатов"
                  title="Серия сертификата">
        <item text="C-MIC" value="C-MIC"/>
        <item text="C-PI" value="C-PI"/>
        <item text="C-MB" value="C-MB"/>
        <item text="C-MI" value="C-MI"/>
        <item text="C-PA" value="C-PA"/>
        <item text="C-FRI" value="C-FRI"/>
        <item text="CC-G" value="CC-G"/>
        <item text="C-CI" value="C-CI"/>
        <item text="C-EE" value="C-EE"/>
        <item text="C-EI" value="C-EI"/>
        <item text="C-GL" value="C-GL"/>
        <item text="C-MB" value="C-MB"/>
        <item text="MED" value="MED"/>
    </text-selector>
    <!--Создание поля ввода, со следующими параметрами: активна цифровая
клавиатура, максимальное количество вводимых символов 9. Название поля ввода:
"Номер сертификата"-->
    <text-field name="idl" keyboard="Digital" max-len="9"
                message="Введите номер сертификата"
                title="Номер сертификата">
        <!--Регулярное выражение для валидации вводимого номера-->
        <verify> <rule regex="^[0-9/]{9}$"/> </verify>
        <!--Подсказка, отображается на терминале-->
        <help> 9 символов. Номер: 123456/15 или 000001/15 </help>
    </text-field>
</item>
<item text="Ипотечное кредитование, лизинг"
```

```
        value="Ипотечное кредитование, лизинг">
<text-selector name="mcp_id2" message="Выберите серию сертификата"
        title="Серия сертификата">
    <item text="M-MIC" value="M-MIC"/>
    <item text="M-PI" value="M-PI"/>
    <item text="MC-MI" value="MC-MI"/>
    <item text="M-PA" value="M-PA"/>
    <item text="M-CI" value="M-CI"/>
    <item text="M-MLV" value="M-MLV"/>
    <item text="M-MI" value="M-MI"/>
    <item text="M-CRI" value="M-CRI"/>
</text-selector>
<!--Создание поля ввода, со следующими параметрами: активна цифровая
клавиатура, максимальное количество вводимых символов 9. Название поля ввода:
"Номер сертификата"-->
    <text-field name="id1" keyboard="Digital" max-len="9"
        message="Введите номер сертификата"
        title="Номер сертификата">
        <!--Регулярное выражение для валидации вводимого номера-->
        <verify> <rule regex="^[0-9/]{9}$"/> </verify>
        <!--Подсказка, отображается на терминале-->
        <help> 9 символов. Номер: 123456/15 или 000001/15 </help>
    </text-field>
</item>
</text-selector>
</item>
<item text="Обязательное страхование" value="Обязательное страхование">
    <text-selector name="mcp_type3" message="Страхование" title="Страхование">
        <item text="Страхование жизни" value="Страхование жизни">
            <text-selector name="mcp_type5" message="Страхование"
                title="Страхование">
                <item text="Транспорт" value="Транспорт">
                    <text-selector name="mcp_id2" message="Выберите серию сертификата"
                        title="Серия сертификата">
                        <item text="I-MLC" value="I-MLC">
                            <text-selector name="Номер" message="Выберите тип платежа"
                                title="Тип оплаты">
                                <item text="По номеру бланка" value="Номер бланка">
                                    <!--Создание поля ввода, со следующими параметрами: со следующими
параметрами: активна цифровая, символьная и буквенная клавиатура, язык ввода
– английский, без возможности смены, максимальное количество вводимых символов
9. Название поля ввода: "Номер бланка"-->
                                    <text-field name="id1" keyboard="any:[en,symb]:lower:true"
```

```
max-len="6" message="Введите номер бланка"
title="Номер бланка">
  <!--Регулярное выражение для валидации вводимого номера-->
  <verify> <rule regex="^\d{6}$"/> </verify>
  <!--Подсказка, отображается на терминале-->
  <help> Номер: 123456 </help>
</text-field>
</item>
<item text="По номеру сертификата" value="Номер сертификата">
  <!--Создание поля ввода, со следующими параметрами: активна
цифровая клавиатура, максимальное количество вводимых символов 13. Название
поля ввода: "Номер сертификата"-->
  <text-field name="id1" keyboard="Digital" max-len="13"
message="Введите номер сертификата"
title="Номер сертификата">
    <!--Регулярное выражение для валидации вводимого номера-->
    <verify> <rule regex="^(PPA|LPA|RPA)\d{10}$"/> </verify>
    <!--Подсказка, отображается на терминале-->
    <help>
      Первые три символа PPA, LPA или RPA. Номер: PPA1234567891
    </help>
  </text-field>
</item>
</text-selector>
</item>
</text-selector>
</item>
<item text="Имущество" value="Имущество">
  <text-selector name="mcp_id2" message="Выберите серию сертификата"
title="Серия сертификата">
    <item text="PIC" value="PIC"/>
    <item text="LIC" value="LIC"/>
  </text-selector>
  <text-selector name="nomer" message="Выберите тип оплаты"
title="Тип оплаты">
    <item text="По номеру бланка" value="Номер бланка">
      <!--Создание поля ввода, со следующими параметрами: со следующими
параметрами: активна цифровая, символьная и буквенная клавиатура, язык ввода
- английский, без возможности смены, максимальное количество вводимых символов
9. Название поля ввода: "Номер бланка"-->
      <text-field name="id1" keyboard="any:[en,symb]:lower:true"
max-len="6" message="Введите номер бланка"
title="Номер бланка">
```

```
<!--Регулярное выражение для валидации вводимого номера-->
<verify> rule regex="^\d{6}$"/> </verify>
<!--Подсказка, отображается на терминале-->
<help> Номер: 123456 </help>
</text-field>
</item>
<item text="По номеру сертификата" value="Номер сертификата">
<!--Создание поля ввода, со следующими параметрами: со следующими
параметрами: активна цифровая клавиатура, максимальное количество вводимых
символов 13. Название поля ввода: "Номер сертификата"-->
<text-field name="id1" keyboard="Digital" max-len="13"
message="Введите номер сертификата"
title="Номер сертификата">
<!--Регулярное выражение для валидации вводимого номера-->
<verify> <rule regex="^(PPA|LPA|RPA)\d{10}$"/> </verify>
<!--Подсказка, отображается на терминале-->
<help>
Первые три символа PPA, LPA или RPA. Номер: PPA1234567891
</help>
</text-field>
</item>
</text-selector>
</item>
</text-selector>
</item>
<item text="Страхование гражданской ответственности"
value="Страхование гражданской ответственности">
<text-selector name="mcp_type4" message="Страхование"
title="Страхование">
<item text="Транспорт" value="Транспорт">
<text-selector name="mcp_id2" message="Выберите серию сертификата"
title="Серия сертификата">
<item text="С-MLC" value="С-MLC">
<text-selector name="nomer" message="Выберите тип оплаты"
title="Тип оплаты">
<item text="По номеру бланка" value="Номер бланка">
<!--Создание поля ввода, со следующими параметрами: со следующими
параметрами: активна цифровая, символьная и буквенная клавиатура, язык ввода
- английский, без возможности смены, максимальное количество вводимых символов
6. Название поля ввода: "Номер бланка"-->
<text-field name="id1" keyboard="any:[en,symb]:lower:true"
max-len="6" message="Введите номер бланка"
title="Номер бланка">
```

```
<verify> <rule regex="^\d{6}$"/> </verify>
<!--Подсказка, отображается на терминале-->
<help> Номер: 123456 </help>
</text-field>
</item>
<item text="По номеру сертификата" value="Номер сертификата">
  <!--Создание поля ввода, со следующими параметрами: со следующими
параметрами: активна цифровая клавиатура, максимальное количество вводимых
символов 13. Название поля ввода: "Номер сертификата"-->
  <text-field name="id1" keyboard="Digital" max-len="13"
    message="Введите номер сертификата"
    title="Номер сертификата">
    <!--Регулярное выражение для валидации вводимого номера-->
    <verify> <rule regex="^(PPA|LPA|RPA)\d{10}$"/> </verify>
    <!--Подсказка, отображается на терминале-->
    <help>
      Первые три символа 3 PPA, LPA или RPA. Номер: PPA1234567891
    </help>
  </text-field>
</item>
</text-selector>
</item>
</text-selector>
</item>
<item text="Имущество" value="Имущество">
  <text-selector name="mcp_id2" message="Выберите серию сертификата"
    title="Серия сертификата">
    <item text="C-PIC" value="C-PIC"/>
    <item text="LIC" value="LIC"/>
  </text-selector>
  <text-selector name="nomer" message="Выберите тип оплаты"
    title="Тип оплаты">
    <item text="По номеру бланка" value="Номер бланка">
      <!--Создание поля ввода, со следующими параметрами: со следующими
параметрами: активна цифровая, символьная и буквенная клавиатура, язык ввода
– английский, без возможности смены, максимальное количество вводимых символов
6. Название поля ввода: "Номер бланка"-->
      <text-field name="id1" keyboard="any:[en,symb]:lower:true"
        max-len="6" message="Введите номер бланка"
        title="Номер бланка">
        <!--Регулярное выражение для валидации вводимого номера-->
        <verify> <rule regex="^\d{6}$"/> </verify>
        <!--Подсказка, отображается на терминале-->
```

```
<help> Номер: 123456 </help>
</text-field>
</item>
<item text="По номеру сертификата" value="Номер сертификата">
  <!--Создание поля ввода, со следующими параметрами: со следующими
параметрами: активна цифровая клавиатура, максимальное количество вводимых
символов 13. Название поля ввода: "Номер сертификата"-->
  <text-field name="id1" keyboard="Digital" max-len="13"
    message="Введите номер сертификата"
    title="Номер сертификата">
    <!--Регулярное выражение для валидации вводимого номера-->
    <verify> <rule regex="^(PPA|LPA|RPA)\d{10}$"/> </verify>
    <!--Подсказка, отображается на терминале-->
    <help>
      Первые три символа 3 PPA, LPA или RPA. Номер: PPA1234567891
    </help>
  </text-field>
</item>
</text-selector>
</item>
</text-selector>
</item>
</text-selector>
</item>
</text-selector>
<!--Создание поля ввода, со следующими параметрами: со следующими
параметрами: активна цифровая, символьная и буквенная клавиатура, язык ввода
- английский, без возможности смены, максимальное количество вводимых символов
6. Название поля ввода: "ФИО"-->
<text-field name="mcp_fio" keyboard="any:[en,symb]:lower:true"
  max-len="60" message="Введите фамилию, имя, отчество"
  title="ФИО">
  <verify> <rule regex="^{3,60}$"/> </verify>
</text-field>
<text-field name="mcp_mail" keyboard="any:[en,symb]:lower:true"
  max-len="40" message="Введите адрес электронной почты e-mail"
  title="E-mail">
  <!--Регулярное выражение для валидации вводимых данных-->
  <verify>
    <rule regex="^([a-z0-9\_\\.\\-]+(\\\_\\.\\-)[a-z0-9]+)*@[a-z0-9][a-z0-9-
9\_\\.\\-]*\\.([a-z]+)?$"/>
  </verify>
  <!--Подсказка, отображается на терминале-->
```

```
<help>E-mail</help>
</text-field>
<text-field name="mcp_nomt" keyboard="Digital" max-len="12"
            message="Введите номер телефона" title="Номер телефона">
  <verify> <rule regex="\d{10,12}$"/> </verify>
  <!--Подсказка, отображается на терминале-->
  <help> Номер: 0701234567 </help>
</text-field>
</form>
</forms>
```