

Министерство образования Республики Беларусь
Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра информатики

Отчет по производственной практике

Выполнил:
студент группы 953504
Басенко К. А.

Руководитель от БГУИР:
Анисимов В.Я.

Руководитель от
предприятия:
Чичин. Е. В.

Минск 2022

Содержание

1. Введение. 3
2. Индивидуальное задание. 4
3. Общая характеристика предприятия. 5
4. Сведения об используемых технологиях. 6
5. Реализация индивидуального задания. 13
6. Заключение. 23
7. Источники использованной литературы. 24
8. Приложение А - листинг кода. 25

Введение

Предприятие, в процессе своей работы, порождает большие объемы информации. В эти кластеры входят результаты ее деятельности, процессов, ошибки, отклонения, а также наблюдения. В сыром виде все это не представляет никакой ценности, но для роста предприятия и повышения ее эффективности их обработка может принести большую выгоду. Именно поэтому при запуске технологических процессов, крайне важно создать качественную систему классификации данных, которая позволит отслеживать, управлять, прогнозировать эффективность предприятия на каждом этапе его деятельности.

SAP предоставляет такую возможность в множестве своих продуктов, одним из которых является SAP Quality Management(QM). Модуль призван интегрировать процессы по управлению качеством продукции предприятия с другими продуктами SAP, а также предоставляет функционал по сбору, хранению, выводу и анализу данных связанных с качеством на предприятии, что является первостепенным в современном производстве.

Базовой структурой для классификации информации в модуле являются каталоги (catalogs). Они призваны помочь в управлении, группировке и стандартизации информации (к примеру типы дефектов, задачи, атрибуты характеристик) на стороне заказчика или завода. Каталоги помогают записывать и последовательно рассчитывать качественные данные и описывать возникающие проблемы.

Каталоги напрямую привязываются к характеристикам продукта, хранят информацию на уровне характеристик в списках задач по проверке качества, упрощают запись результатов проверки, дефектов и мест дефектов во время проверки, организуют принятие решения пользователем по результатам проверки.

SAP зарезервировал коды 0-9 и A-O для своих коробочных каталогов, но заказчики могут определять свои используя коды от P до Z.

Целью практического задания является анализ и изучения структуры каталогов в модуле SAP QM, а также разработка программного интерфейса (API) для модификации и настройки каталогов заказчиков на предприятии.

Индивидуальное задание

Тема: разработка веб-приложения *Bakery* (пекарня), выполняющее *CRUD*-операции с продуктами.

Используемые технологии:

- *SAP ABAP*

Задачи:

- Создать таблицы для описания используемых сущностей, установить связи между ними.
- Создать функциональную группу для поддержки/обслуживания таблиц, функциональные модули из группы выполняют *CRUD*-операции.
- Создать классы для описания поведения используемых сущностей
- Создать веб-приложение, объединяющее весь созданный функционал в удобный для пользователя *GUI*.

Общая характеристика предприятия

Общество с ограниченной ответственностью «Леверекс Интернешнл» является официальным сервисным партнером SAP и ориентирована на разработку и внедрение программного обеспечения на платформе программных продуктов и средств разработки компании SAP, а также на разработку веб-, мобильных и облачных/серверных программных продуктов. Основная цель деятельности компании — повышение отдачи от инвестиций заказчиков в программное обеспечение и автоматизацию бизнес-процессов. Компания является Стратегическим Технологическим Партнером большого количества компаний и стартапов в США и Европе.

Компания "Леверекс Интернешнл" создана в 2005 году и является частью международной группы компаний LeverX со штаб-квартирой в г. Маунтин-Вью (Калифорния, США), офисами и центрами разработки в Трое (Мичиган, США), Кинг-оф-Пруссия (Филадельфия, США), Риге (Латвия), Киеве (Украина), Москве (Российская Федерация), Ташкенте (Узбекистан), а также международным центром разработки в Минске (Республика Беларусь).

Среди ее услуг имеют место экспертиза в новейших версиях модулей и платформах SAP: ERP и финансы, CRM и интернет-торговля, SCM (управление цепочками поставок), закупки, PLM и управление производством, управление активами, BI аналитические решения, Business Technology Platform, а также свои пакеты решений.

Как и среди продуктов SAP, компания предоставляет глубокую техническую экспертизу среду ее технологий, таких как Front-end Fiori, ABAP, SAP S/4HANA, Cloud Foundry, SAP HANA, Odata, мобильные технологии SAP и др.

Также компания работает над решениями в областях машинного обучения, интернета вещей, блокчейна, виртуальной и дополненной реальности.

На рисунке представлены некоторые из клиентов ООО «Леверекс Интернешнл».



За время существования компании завершили более 550 проектов для клиентов по всему миру. Среди клиентов — постоянно растущее число компаний из списка Fortune 1000, а также средние по размеру игроки рынка.

Сведения об используемых технологиях

ABAP (Advanced Business Application Programming) – внутренний язык высокого уровня программирования в среде SAP. Углубляться в историю особенно не хочется, но отмечу лишь, что синтаксис наиболее близок к языку программирования COBOL.

Язык позволяет работать с внутренними структурами данных, интерфейсами пользователя, транзакциями, отчётами, интерфейсами загрузки, выгрузки данных (word, excel, pdf, ...); стоит отметить, что можно использовать объектно-ориентированные конструкции (ABAP Objects); существует множество технологий связи с другими системами (BAPI, RFC, ...) для выгрузки и загрузки данных (или же сторонней обработки).

Типизация. Переменные в ABAP обычно объявляются с помощью конструкции DATA:

```
DATA: count TYPE i. " объявляем переменную count типа integer
```

Пример со структурой:

```
TYPES: BEGIN OF struct,
        name(10) type C, " символьная переменная длиной 10 символов
        post(15) type C,
END OF struct.

DATA: wa_struct TYPE STANDARD TABLE OF struct. " таблица объявленного нами типа
DATA: name_person LIKE wa_struct-name. " ссылка на элемент name объявленной структуры
```

ABAP имеет свою собственную среду разработки, которая называется ABAP Workbench, имеющую отладчик, средства трассировки, систему контроля версий, средства работы с буфером, журналы использования, и так далее.

В 2012 году разработчики интегрировали ABAP со всем известной IDE Eclipse. Правда работает это на последних версиях ядра, что есть не на всех проектах.

ABAP – это язык программирования, работающий в среде выполнения SAP ABAP, созданный и используемый SAP для разработки прикладных программ, включая:

- Отчеты
- Программирование модульного пула

- Интерфейсы
- формы
- Преобразования данных
- Выход пользователя и BADI

Все приложения R / 3 и даже части его базовой системы были разработаны в АВАР.

АВАР также называется АВАР / 4. «4» в АВАР / 4 означает «язык четвертого поколения» или 4GL.

АВАР Workbench используется SAP для разработки стандартного и специального прикладного программного обеспечения. АВАР Workbench также используется для создания объектов словаря. Он состоит из следующих компонентов –

- **АВАР Editor** используется для поддержки программ.
- **Словарь АВАР** используется для поддержки объектов Dictionary.
- **Браузер репозитория** используется для отображения иерархической структуры компонентов в пакете.
- **Меню Painter** используется для разработки графических пользовательских интерфейсов, включая строки меню и панели инструментов.
- **Screen Painter** используется для поддержки компонентов экрана для онлайн-программ.
- **Информационная система репозитория** содержит информацию об объектах разработки и времени выполнения, таких как модели данных, типы словарей и структуры таблиц, программы и функции.
- **Инструменты тестирования и анализа**, такие как проверка синтаксиса и отладчик.
- **Function Builder**, который позволяет создавать и поддерживать функциональные группы и функциональные модули.
- **Data Modeler**, инструмент, который поддерживает графическое моделирование.
- **Workbench Organizer**, который поддерживает несколько проектов разработки и управляет их распространением.

АВАР Editor используется для поддержки программ.

Словарь АВАР используется для поддержки объектов Dictionary.

Браузер репозитория используется для отображения иерархической структуры компонентов в пакете.

Меню Painter используется для разработки графических пользовательских интерфейсов, включая строки меню и панели инструментов.

Screen Painter используется для поддержки компонентов экрана для онлайн-программ.

Информационная система репозитория содержит информацию об объектах разработки и времени выполнения, таких как модели данных, типы словарей и структуры таблиц, программы и функции.

Инструменты тестирования и анализа, такие как проверка синтаксиса и отладчик.

Function Builder, который позволяет создавать и поддерживать функциональные группы и функциональные модули.

Data Modeler, инструмент, который поддерживает графическое моделирование.

Workbench Organizer, который поддерживает несколько проектов разработки и управляет их распространением.



Программы отчетов создают списки и могут быть разделены на классические отчеты и интерактивные отчеты.

- **Классические отчеты** не допускают взаимодействия с пользователем; поэтому базовый список содержит обширную информацию, которую пользователь должен часто сортировать, чтобы найти соответствующие данные.
- **Интерактивные отчеты** позволяют взаимодействие с пользователем; следовательно, пользователь может создавать

вторичные подробные списки базового списка, выбирая соответствующие данные и запрашивая дополнительную информацию.

- **SAP Query или Ad-hoc Query или InfoSet Query** – это инструмент, который позволяет конечному пользователю создавать различные запросы на основе разных входных и выходных параметров в системе SAP в соответствии с требованиями. Это один из инструментов отчетности, который в основном используется в модуле управления персоналом для извлечения данных из реляционных баз данных. InfoSet Query подходит для создания отчетов во всех областях системы SAP R / 3.

Классические отчеты не допускают взаимодействия с пользователем; поэтому базовый список содержит обширную информацию, которую пользователь должен часто сортировать, чтобы найти соответствующие данные.

Интерактивные отчеты позволяют взаимодействие с пользователем; следовательно, пользователь может создавать вторичные подробные списки базового списка, выбирая соответствующие данные и запрашивая дополнительную информацию.

SAP Query или Ad-hoc Query или InfoSet Query – это инструмент, который позволяет конечному пользователю создавать различные запросы на основе разных входных и выходных параметров в системе SAP в соответствии с требованиями. Это один из инструментов отчетности, который в основном используется в модуле управления персоналом для извлечения данных из реляционных баз данных. InfoSet Query подходит для создания отчетов во всех областях системы SAP R / 3.

Специальный запрос предоставляет три различных вида отчетов:

- **Основной список** – Простые отчеты.
- **Статистика** – отчеты со статистическими функциями, такими как среднее, процент и т. Д.
- **Ранжированный список** – для аналитических отчетов.

Основной список – Простые отчеты.

Статистика – отчеты со статистическими функциями, такими как среднее, процент и т. Д.

Ранжированный список – для аналитических отчетов.

Код транзакции, используемый для доступа к специальным запросам –

- **SQ01** – Ведение запросов
- **SQ02** – Показать инфо-набор

- **SQ03** – Ведение групп пользователей

База данных SAP должна быть заполнена до того, как конечные пользователи смогут начать работу над бизнес-процессом для анализа и составления отчетов. Различные методы используются для передачи данных в систему на разных этапах в зависимости от сложности и объема передаваемых данных.

Данные могут быть перенесены из SAP в SAP или SAP в не-SAP-системы (устаревшая система). Данные также могут быть переданы с помощью ручного ввода. Инструменты, используемые для передачи данных, следующие:

- BDC (пакетная передача данных)
- IDOC (промежуточный документ)
- LSMW (Legacy System Migration Workbench)
- Интерфейсы через загрузку файлов .txt или Excel, содержащих данные
- Ручной ввод данных с использованием кодов транзакций

SAP создает различные документы, такие как заказы на поставку, заказы на продажу, счета-фактуры, платежные ведомости и т. Д. Вы можете распечатать эти документы, когда это необходимо. На следующем рисунке показано, как работает процесс печати в системе SAP.

В системе SAP требуются различные методы печати, такие как –

- Локальная печать
- Удаленная печать (сеть)
- Фронтальная печать (SAP GUI для Windows)
- Фронтальная печать (SAP GUI для HTML)

Программы могут быть написаны в интерактивном режиме с использованием возможностей системы R / 3. Программы могут выполняться как онлайн, так и в фоновом режиме. Фоновые задания также можно запланировать на определенные интервалы.

Программирование пула модулей (или онлайн-программирование) включает создание пула модулей (набор модулей ABAP) и одного или нескольких экранов. Модули вызываются процессором экрана во время выполнения программы.

Обработка пакетного ввода используется для обеспечения безопасной передачи данных в систему SAP. Этот процесс представляет собой автоматический защищенный перенос данных в систему SAP,

которая использует транзакции SAP для проверки данных при заполнении базы данных SAP.

ABAP содержит операторы, которые соответствуют стандартам CРI-С (Common Program Interface – Communications). Они используются для программирования коммуникационных программ. ABAP может читать и записывать последовательные наборы данных.

Реализация индивидуального задания

Для описания атрибутов/характеристик используемых сущностей, были созданы домены, на которых они основаны:

- ▼ Domains
 - YBASE_D_BAK_ID
 - YBASE_D_BAK_PROD_NUM
 - YBASE_D_BAK_PROD_TYPE

Так, например, один из атрибутов продукта пекарни - это его тип. Благодаря доменам, все возможные типы продуктов определены, что исключает ситуации записи несоответствующих или непонятных продуктов:

Domain		YBASE_D_BAK_PROD_TYPE	Active
Short Description		Bakery product type	
<div>Properties Definition Value Range</div>			
Single Vals			
I	Fix.Val.	Short Descript.	
<input type="checkbox"/>	BREAD		
<input type="checkbox"/>	COOKIES		
<input type="checkbox"/>	CAKES		
<input type="checkbox"/>	PIES		
<input type="checkbox"/>	ECLAIRS		

Для реализации индивидуального задания были созданы таблицы *YBASE_BAK* (таблица для магазина) и *YBASE_BAK_PROD* (таблица для продуктов магазина), где вторая таблица имеет внешний ключ *BAKERY*.

YBASE_BAK:

Field	Key	Inl...	Data element	Data Type	Length	Deci...	Short Description
MANDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3	0	Client
ID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	YBASE_E_BAK_ID	CHAR	20	0	Bakery id
DESCR	<input type="checkbox"/>	<input type="checkbox"/>	YBASE_E_BAK_DES...	CHAR	20	0	Bakery description
CAPACITY	<input type="checkbox"/>	<input type="checkbox"/>	YBASE_E_BAK_CAP...	CHAR	20	0	Bakery product capacity
.INCLUDE	<input type="checkbox"/>	<input type="checkbox"/>	YBASE_TECH_INFO	STRU	0	0	Technical Data
CREATED_BY	<input type="checkbox"/>	<input type="checkbox"/>	USNAM	CHAR	12	0	User Name
CREATED_ON	<input type="checkbox"/>	<input type="checkbox"/>	TIMESTAMP	DEC	15	0	UTC Time Stamp in Short Form (YYYYMMDDhhmmss)

YBASE_BAK_PROD:

Field	Key	Inl...	Data element	Data Type	Length	Deci...	Short Description
MANDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3	0	Client
NUMB	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	YBASE_E_BAK_PRO...	CHAR	20	0	Bakery product number
TYPE	<input type="checkbox"/>	<input type="checkbox"/>	YBASE_E_BAK_PRO...	CHAR	20	0	Bakery product type
WEIGHT	<input type="checkbox"/>	<input type="checkbox"/>	BRGEW	QUAN	13	3	Gross Weight
WEIGHT_UNIT	<input type="checkbox"/>	<input type="checkbox"/>	GEWEI	UNIT	3	0	Weight Unit
BAKERY	<input type="checkbox"/>	<input type="checkbox"/>	YBASE_E_BAK_ID	CHAR	20	0	Bakery id

Для удобства дальнейшего обслуживания таблиц были созданы их представления (views).

YBASE_V_BAK:

Maintenance View	YBASE_V_BAK	Active						
Short Description	Bakeries							
Attributes	Table/Join Conditions	View Fields						
Selection Conditions								
Maint.Status								
Table fields								
View field	Table	Field	Key	Data elem.	M...	DType	Length	Short description
MANDT	YBASE_BAK	MANDT	<input checked="" type="checkbox"/>	MANDT	<input type="checkbox"/>	CLNT	3	Client
ID	YBASE_BAK	ID	<input checked="" type="checkbox"/>	YBASE_F_BAK_ID	<input type="checkbox"/>	CHAR	20	Bakery id
DESCR	YBASE_BAK	DESCR	<input type="checkbox"/>	YBASE_F_BAK_DESCR	<input type="checkbox"/>	CHAR	20	Bakery description
CAPACITY	YBASE_BAK	CAPACITY	<input type="checkbox"/>	YBASE_F_BAK_CAPACITY	<input type="checkbox"/>	CHAR	20	Bakery product capacity
COMMENTARY	YBASE_BAK	COMMENTARY	<input type="checkbox"/>	YBASE_F_BAK_COMMENT	<input type="checkbox"/>	CHAR	20	Bakery comment

YBASE_V_BAK_PROD:

Maintenance View

YBASE_V_BAK_PROD

Active

Short Description

Bakery products

Attributes

Table/Join Conditions

View Fields

Selection Conditions

Maint.Status

Table fields

View field	M...	DTyp	Length	Short description
MANDT	<input type="checkbox"/>	CLNT	3	Client
NUMB	<input type="checkbox"/>	CHAR	20	Bakery product number
TYPE	<input type="checkbox"/>	CHAR	20	Bakery product type
WEIGHT	<input type="checkbox"/>	QUAN	13	Gross Weight
WEIGHT_UNIT	<input type="checkbox"/>	UNIT	3	Weight Unit
BAKERY	<input type="checkbox"/>	CHAR	20	Bakery id

Функциональная группа YBASE_FG_BAK_PROD, содержащая 4 функциональных модуля:

- YBASE_FM_CREATE_BAK_PROD для создания продуктов
- YBASE_FM_READ_BAK_PROD для просмотра имеющихся, записанных продуктов
- YBASE_FM_UPDATE_BAK_PROD для обновления информации о продуктах
- YBASE_FM_DELETE_BAK_PROD для удаления продуктов

Каждый модуль выполняет предназначенную работу и в исключительных ситуациях, например, если удалять несуществующий продукт, уведомляет об этом. Пример функции удаления:

```

1 FUNCTION YBASE_FM_READ_BAK_PROD.
2 *-----
3 *""Local Interface:
4 *  IMPORTING
5 *    REFERENCE (IV_BAK) TYPE  YBASE_E_BAK_ID
6 *  EXPORTING
7 *    REFERENCE (ET_BAK_PROD) TYPE  YBASE_TT_BAK_PROD
8 *  EXCEPTIONS
9 *    NO_DATA
10 *-----
11
12      SELECT * FROM ybase_bak_prod as prod
13      INTO CORRESPONDING FIELDS OF TABLE @et_bak_prod
14      WHERE prod~bakery = @iv_bak.
15
16      IF sy-subrc <> 0.
17          MESSAGE e004(ybase_msg_bak) RAISING no_data.
18      ENDIF.
19
20 ENDFUNCTION.

```

Для каждой сущности созданы классы:

- YBASE_CL_BAK для описания магазина/пекарни
- YBASE_CL_BAK_PROD для описания продуктов/изделий пекарни.

Атрибуты класса YBASE_CL_BAK:

Class/InterfaceYBASE_CL_BAKImplemented / Active

Properties

Interfaces

Friends

Attributes

Methods

Events

Types

Aliases

Properties

Filter

Attribute	Level	Visibility	R...	Typing	Associated Type
MS_BAK	Instance Attribute	Private	<input type="checkbox"/>	Type	YBASE_V_BAK
MT_BAK_PROD_REF	Instance Attribute	Private	<input type="checkbox"/>	Type	LTY_TT_BAK_PROD_REF

Методы класса YBASE_CL_BAK:

Class/Interface

YBASE_CL_BAK

Implemented / Active

Properties

Interfaces

Friends

Attributes

Methods

Events

Types

Aliases

Parameters

Exceptions

Sourcecode

</

Атрибуты класса YBASE_CL_BAK_PROD:

считается создание отдельных репозиториев для каждого бизнес-объекта или контекста.

Созданные репозитории реализуют соответствующие им интерфейсы:

YBASE_IF_BAK_REPOSITORY:

Interface		YBASE_IF_BAK_REPOSITORY	Implemented / Active
<div>Properties Interfaces Attributes Methods Events Types Aliases</div>			
<div>Parameters Exceptions</div>			
Method	Level	M...	Description
READ_BAK_BY_ID	Instance Method		

YBASE_IF_BAK_PROD_REPOSITORY:

Interface		YBASE_IF_BAK_PROD_REPOSITORY	Implemented / Active
<div>Properties Interfaces Attributes Methods Events Types Aliases</div>			
<div>Parameters Exceptions</div>			
Method	Level	M...	Description
CREATE	Instance Method		
UPDATE	Instance Method		
DELETE	Instance Method		

Для валидации данных, например, при создании нового продукта, создан *Enhancement Spot*, позволяющий добавлять новые проверки адекватности данных, не изменяя кода классов в дальнейшем:

Enhancement Spot

YBASE_ES_BAK_VALID

Active

Attributes

Enhancem. Implementations

Technical Details

Enh. Spot Element Definitions

BAoI Definitions

YBASE_BD_BAK_VALID Bakery Additional Validation

Interface

Implementations

Description

Bakery Additional Validation

Interface

BAoI Definition

YBASE_BD_BAK_VALID

Interface

YBASE_IF_ADD_BAK_PROD_VALID

Method

ADD_NEW_BAK_PROD_VALID

Description

Enhancement Spot

YBASE_ES_BAK_VALID

Active

Attributes

Enhancem. Implementations

Technical Details

Enh. Spot Element Definitions

BAoI Definitions

YBASE_BD_BAK_VALID Bakery Additional Validation

Interface

Implementations

Description

Bakery Additional Validation

Implementations

BAoI Definition

YBASE_BD_BAK_VALID

1 implementation found

Active

Enhancement Implementation

BaoI Implementation

Software Component

Applic. Component

YBASE_EI_BAK_VALID

YBASE_IM_BAK_VALID

HOME

Итоговым этапом стало создание Web Dynpro приложения.

Web Dynpro – это среда ABAP для веб-разработки, основанная на концепции Model View Controller (MVC) программирования

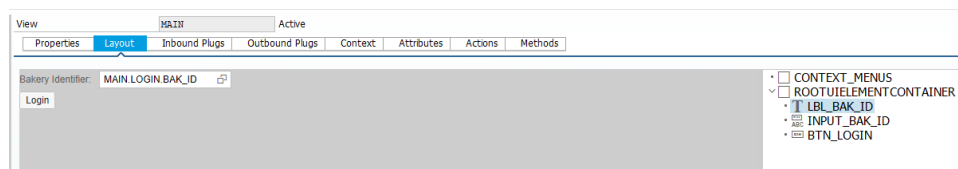
пользовательского интерфейса, стандартная технология SAP UI, которая позволяет разрабатывать веб-приложения с использованием графических инструментов и среды разработки, интегрированной с рабочей средой АВАР. Использование графических инструментов снижает трудозатраты на внедрение, и вы можете лучше повторно использовать и поддерживать компоненты в инструментальных средствах АВАР.

Созданные представления:

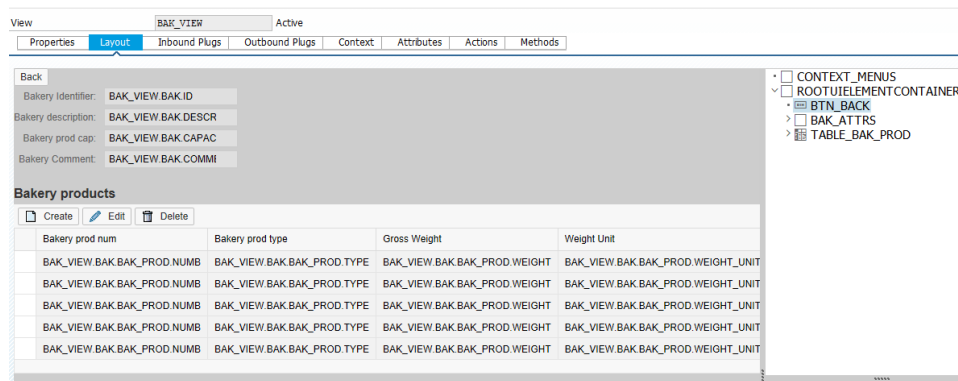
- MAIN - для выбора пекарни
- BAK_VIEW - осуществление CRUD-операций с продуктами
- BAK_PROD_POPUP_VIEW - представление для задания атрибутов для нового продукта, или для изменения атрибутов уже существующего

Предварительный просмотр представлений:

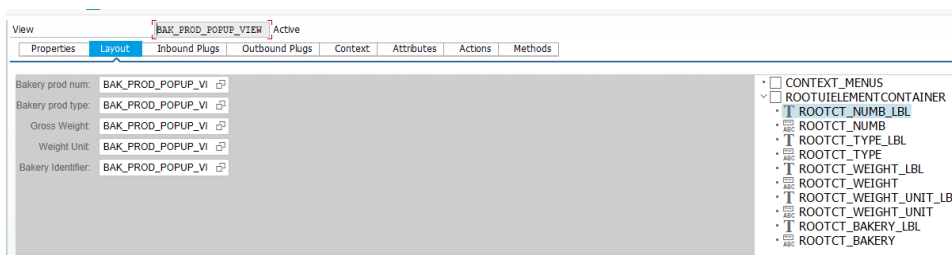
MAIN:



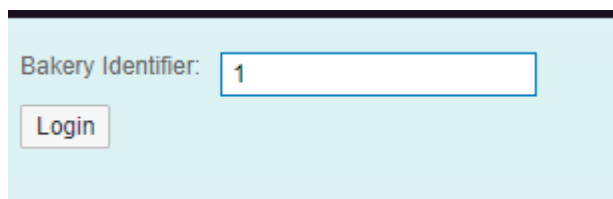
BAK_VIEW:



BAK_PROD_POPUP_VIEW:

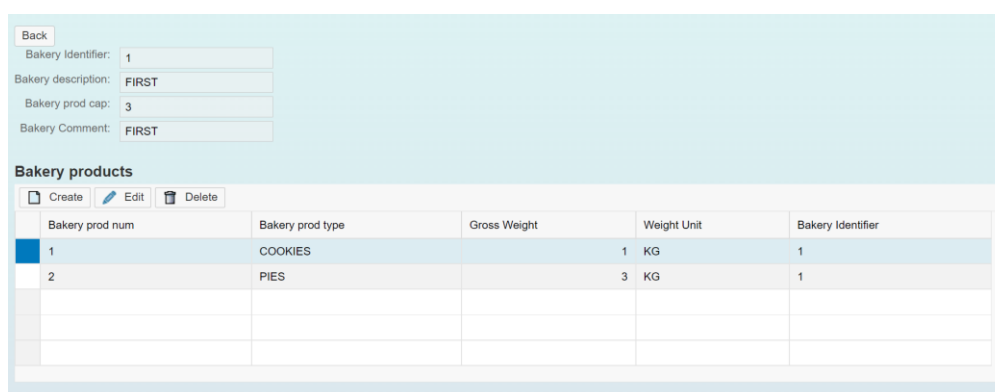


При запуске, первым делом, выбираем пекарню, продукты которой нас интересуют:



A login form with a light blue background. It contains a label "Bakery Identifier:" followed by a text input field containing the number "1". Below the input field is a button labeled "Login".

Далее можем взглянуть на имеющиеся продукты пекарни и ее атрибуты:



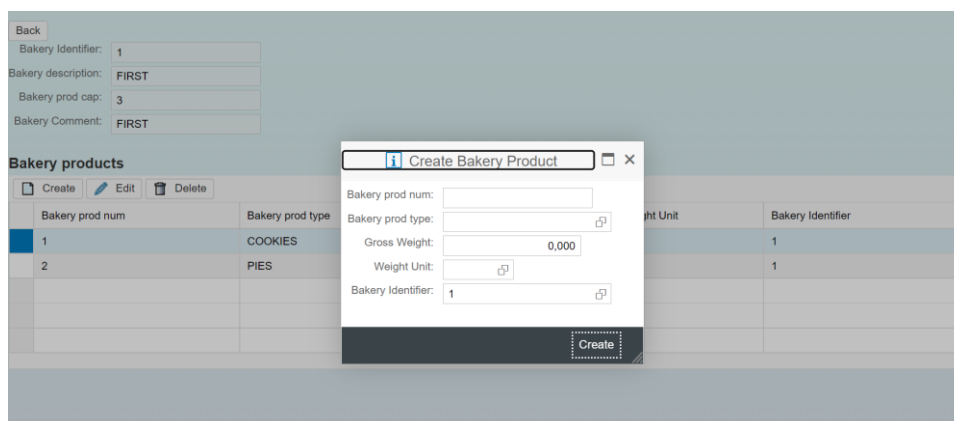
A screenshot of a web application showing bakery products. At the top, there are input fields for "Bakery Identifier: 1", "Bakery description: FIRST", "Bakery prod cap: 3", and "Bakery Comment: FIRST". Below these is a section titled "Bakery products" with buttons for "Create", "Edit", and "Delete". A table lists the products:

Bakery prod num	Bakery prod type	Gross Weight	Weight Unit	Bakery Identifier
1	COOKIES	1	KG	1
2	PIES	3	KG	1

При выборе продукта в таблице, путем нажатия, мы выбираем его для дальнейших действий.

Если наш выбор добавить новый продукт в эту пекарню, нажимаем на кнопку *Create*, расположенную над таблицей. В случае изменения -- Edit, удаления -- Delete.

При добавлении нового продукта требуется определить его атрибуты:



A screenshot of the web application with a modal window titled "Create Bakery Product" open. The modal contains the following fields:

- Bakery prod num: (empty)
- Bakery prod type: (dropdown menu)
- Gross Weight: 0,000
- Weight Unit: (dropdown menu)
- Bakery Identifier: 1

At the bottom of the modal is a "Create" button. The background shows the same bakery products table as in the previous image.

Back

Bakery Identifier: 1

Bakery description: FIRST

Bakery prod cap: 3

Bakery Comment: FIRST

Bakery products

Create Edit Delete

Bakery prod num	Bakery prod type	Gross Weight	Weight Unit	Bakery Identifier
1	COOKIES			1
2	PIES			1

Create Bakery Product

Bakery prod num: 3

Bakery prod type: BREAD

Gross Weight: 0,500

Weight Unit: LB

Bakery Identifier: 1

Create

All 1

Bakery products 3 were created successfully

Back

Bakery Identifier: 1

Bakery description: FIRST

Bakery prod cap: 3

Bakery Comment: FIRST

Bakery products

Create Edit Delete

Bakery prod num	Bakery prod type	Gross Weight	Weight Unit	Bakery Identifier
1	COOKIES	1	KG	1
2	PIES	3	KG	1
3	BREAD	0,500	LB	1

При изменении продукта, появится тоже окно, в котором мы изменяем атрибуты продукта:

Back

Bakery Identifier: 1

Bakery description: FIRST

Bakery prod cap: 3

Bakery Comment: FIRST

Bakery products

Create Edit Delete

Bakery prod num	Bakery prod type	Gross Weight	Weight Unit	Bakery Identifier
1	COOKIES			1
2	PIES			1
3	BREAD			1

Update Bakery Product

Bakery prod num: 3

Bakery prod type: BREAD

Gross Weight: 0,500

Weight Unit: LB

Bakery Identifier: 1

Update

Back

Bakery Identifier: 1

Bakery description: FIRST

Bakery prod cap: 3

Bakery Comment: FIRST

Bakery products

Create Edit Delete

Bakery prod num	Bakery prod type	Gross Weight	Weight Unit	Bakery Identifier
1	COOKIES			1
2	PIES			1
3	BREAD			1

Update Bakery Product

Bakery prod num: 3

Bakery prod type: BREAD

Gross Weight: 0,500

Weight Unit: LB

Bakery Identifier: 2

Update

All ☒ 1

☒ Bakery products 3 were updated successfully

Back

Bakery Identifier: 1

Bakery description: FIRST

Bakery prod cap: 3

Bakery Comment: FIRST

Bakery products

Create Edit Delete

Bakery prod num	Bakery prod type	Gross Weight	Weight Unit	Bakery Identifier
1	COOKIES	1	KG	1
2	PIES	3	KG	1

Так, мы изменили вторичный ключ продукта, значит, перенесли его в другой магазин:

Bakery Identifier: 2

Login

Back

Bakery Identifier: 2

Bakery description: SECOND

Bakery prod cap: 3

Bakery Comment: 2ND

Bakery products

Create Edit Delete

Bakery prod num	Bakery prod type	Gross Weight	Weight Unit	Bakery Identifier
3	BREAD	0,500	LB	2

Для удаления также выделяем продукт путем нажатия по нему и удаляем:

Back

Bakery Identifier: 2

Bakery description: SECOND

Bakery prod cap: 3

Bakery Comment: 2ND

Bakery products

Create Edit Delete

Bakery prod num	Bakery prod type	Gross Weight	Weight Unit	Bakery Identifier
3	BREAD	0,500	LB	2

All ☒ 1

☒ Bakery products 3 were deleted successfully

Back

Bakery Identifier: 2

Bakery description: SECOND

Bakery prod cap: 3

Bakery Comment: 2ND

Bakery products

Create Edit Delete

Bakery prod num	Bakery prod type	Gross Weight	Weight Unit	Bakery Identifier
-----------------	------------------	--------------	-------------	-------------------

Заключение

В результате прохождения практики в ООО «Леверекс Интернешнл» были приобретены теоретические и практические навыки в работе с SAP ABAP.

В результате прохождения производственной практики мной были выполнены следующие задачи:

- Была изучена деятельность и структура ООО «Леверекс Интернешнл»;
- Проведен анализ базовых структур данных в системе SAP
- Успешно выполнено индивидуальное задание, заключающееся в разработке функционала и интерфейса для него

Мои знания в области разработки программных продуктов и пользовательских интерфейсов значительно расширились за время практики в компании. Был получен опыт работы с новейшими технологиями из среды SAP.

Задачи, поставленные руководителем практики от предприятия и руководителем практики от университета, были выполнены в процессе прохождения практики.

Источники использованной литературы

1. SAP AG – The Context in Detail – SAP 2007
2. SAP Help Portal – ABAP Documentation - [Электронный ресурс] SAP 2022 - [режим доступа] <https://help.sap.com/docs/ABAP>

Приложение А - листинг кода

YBASE_CL_BAK.abap:

```
class YBASE_CL_BAK definition
public
final
create private
global friends YBASE_CL_BAK_REPOSITORY .
public section.
.....

"Setters
.....

methods SET_DESCR
importing
    !IV_DESCR type YBASE_E_BAK_DESCR
raising
    YCX_BASE_BAK_PROD_MAINT .
methods SET_CAPACITY
importing
    !IV_CAPACITY type YBASE_E_BAK_CAPACITY
raising
    YCX_BASE_BAK_PROD_MAINT .
methods SET_COMMENT
importing
    !IV_COMMENT type YBASE_E_BAK_COMMENT
raising
    YCX_BASE_BAK_PROD_MAINT .
.....

"Getters
.....

methods GET_ID
returning
    value(RV_ID) type YBASE_E_BAK_ID .
methods GET_DESCR
returning
    value(RV_DESCR) type YBASE_E_BAK_DESCR .
methods GET_CAPACITY
returning
    value(RV_CAPACITY) type YBASE_E_BAK_CAPACITY .
methods GET_COMMENT
returning
    value(RV_COMMENT) type YBASE_E_BAK_COMMENT .
.....

methods ADD_NEW_BAK_PROD
importing
    !IO_BAK_PROD type ref to YBASE_CL_BAK_PROD
raising
    YCX_BASE_BAK_PROD_MAINT .
methods REMOVE_BAK_PROD
importing
    !IV_NUMB type YBASE_E_BAK_PROD_NUM
returning
    value(RO_BAK_PROD) type ref to YBASE_CL_BAK_PROD
raising
    YCX_BASE_BAK_PROD_MAINT .
methods UPDATE_BAK_PROD
importing
```



```

!!IO_BAK_PROD type ref to YBASE_CL_BAK_PROD
raising
  YCX_BASE_BAK_PROD_MAINT .
methods FIND_BAK_PROD
importing
  !IV_NUMB type YBASE_E_BAK_PROD_NUM
returning
  value(RO_BAK_PROD) type ref to YBASE_CL_BAK_PROD
raising
  YCX_BASE_BAK_PROD_MAINT .
methods TO_STRUCT
returning
  value(RS_BAK) type YBASE_V_BAK .
methods BAK_PROD_TABLE
returning
  value(RT_BAK_PROD) type YBASE_TT_BAK_PROD .
methods SET_ID
importing
  !IV_ID type YBASE_E_BAK_ID
raising
  YCX_BASE_BAK_PROD_MAINT .
protected section.
private section.
types: begin of lty_s_bak_prod_ref,
  numb type ybase_e_bak_prod_num,
  ref type ref to ybase_cl_bak_prod,
end of lty_s_bak_prod_ref.
types: lty_tt_bak_prod_ref type standard table of lty_s_bak_prod_ref.
data: ms_bak type ybase_v_bak,
  mt_bak_prod_ref type lty_tt_bak_prod_ref.
methods constructor
importing is_bak type ybase_v_bak
  it_bak_prod type ybase_tt_bak_prod
raising ycx_base_bak_prod_maint.
methods find_bak_prod_ref
importing iv_numb type ybase_e_bak_prod_num
exporting es_bak_prod_ref type lty_s_bak_prod_ref
raising ycx_base_bak_prod_maint.
ENDCLASS.
CLASS YBASE_CL_BAK IMPLEMENTATION.
method add_new_bak_prod.
  data: lv_records_count type i,
    lo_badi_bak_valid type ref to ybase_bd_bak_valid.
  if ms_bak-id <> io_bak_prod->get_bak_id( ).
    raise exception type ycx_base_bak_prod_maint
    exporting
      textid = ycx_base_bak_prod_maint=>cx_wrong_bak
      attr1 = CONV #( io_bak_prod->get_bak_id( ) )
      attr2 = CONV #( ms_bak-id ).
  endif.
  select count( * )
    into lv_records_count
  from ybase_bak_prod
  where numb = ms_bak-id.
  if lv_records_count = ms_bak-capacity.
    raise exception type ycx_base_bak_prod_maint
    exporting

```

```

        textid = ycx_base_bak_prod_maint=>cx_max_capacity_violated
        attr1 = conv #( ms_bak-capacity ).
    endif.
read table mt_bak_prod_ref assigning field-symbol(<ls_bak_prod>)
    with key numb = io_bak_prod->get_numb( ).
if sy-subrc = 0.
    raise exception type ycx_base_bak_prod_maint
        exporting
            textid = ycx_base_bak_prod_maint=>cx_bak_prod_already_exist
            attr1 = CONV #( io_bak_prod->get_numb( ) )
            attr2 = CONV #( me->get_id( ) ).
endif.
try.
    get badi lo_badi_bak_valid.
    call badi lo_badi_bak_valid->add_new_bak_prod_valid
        exporting
            is_to_bak = ms_bak
            is_bak_prod = io_bak_prod->ms_bak_prod.
catch
    cx_badi_context_error
    cx_badi_filter_error
    cx_badi_initial_context
    cx_badi_multiply_implemented
    cx_badi_not_implemented
    cx_badi_unknown_error
    cx_badi_initial_reference
    cx_sy_dyn_call_illegal_method.
endtry.
append value #( numb = io_bak_prod->get_numb( ) ref = io_bak_prod ) to mt_bak_prod_ref.
endmethod.
method bak_prod_table.
    loop at mt_bak_prod_ref assigning field-symbol(<ls_bak_prod_ref>).
        append <ls_bak_prod_ref>-ref->to_struct( ) to rt_bak_prod.
    endloop.
endmethod.
method constructor.
    set_id( iv_id = is_bak-id ).
    set_descr( iv_descr = is_bak-descr ).
    set_capacity( iv_capacity = is_bak-capacity ).
    set_comment( iv_comment = is_bak-commentary ).
    loop at it_bak_prod assigning field-symbol(<ls_bak_prod>).
        data(lo_bak_prod) = NEW ybase_cl_bak_prod( is_bak_prod = <ls_bak_prod> ).
        append value #( numb = <ls_bak_prod>-numb ref = lo_bak_prod ) TO mt_bak_prod_ref.
    endloop.
endmethod.
method find_bak_prod.
    find_bak_prod_ref(
        exporting
            iv_numb = iv_numb
        importing
            es_bak_prod_ref = data(ls_bak_prod_ref)
    ).
    ro_bak_prod = ls_bak_prod_ref-ref.
endmethod.
method find_bak_prod_ref.
    read table mt_bak_prod_ref assigning field-symbol(<ls_bak_prod>)
        with key numb = iv_numb.

```

```

if sy-subrc <> 0.
    raise exception type ycx_base_bak_prod_maint
    exporting
        textid = ycx_base_bak_prod_maint=>cx_bak_prod_not_found
        attr1 = CONV #( iv_numb )
        attr2 = CONV #( ms_bak-id ).
endif.
es_bak_prod_ref = <ls_bak_prod>.
endmethod.
method get_capacity.
    rv_capacity = ms_bak-capacity.
endmethod.
method get_comment.
    rv_comment = ms_bak-commentary.
endmethod.
method get_descr.
    rv_descr = ms_bak-descr.
endmethod.
method get_id.
    rv_id = ms_bak-id.
endmethod.
method remove_bak_prod.
    find_bak_prod_ref(
        exporting
            iv_numb = iv_numb
        importing
            es_bak_prod_ref = data(ls_bak_prod_ref)
    ).
    ro_bak_prod = ls_bak_prod_ref-ref.
    delete mt_bak_prod_ref where numb = iv_numb.
endmethod.
method set_capacity.
    ms_bak-capacity = iv_capacity.
endmethod.
method set_comment.
    ms_bak-commentary = iv_comment.
endmethod.
method set_descr.
    ms_bak-descr = iv_descr.
endmethod.
method SET_ID.
    ms_bak-id = iv_id.
endmethod.
method to_struct.
    rs_bak = ms_bak.
endmethod.
method update_bak_prod.
    read table mt_bak_prod_ref assigning field-symbol(<ls_bak_prod>)
        with key numb = io_bak_prod->get_numb( ).
    if sy-subrc <> 0.
        raise exception type ycx_base_bak_prod_maint
        exporting
            textid = ycx_base_bak_prod_maint=>cx_bak_prod_not_found
            attr1 = CONV #( io_bak_prod->get_numb( ) )
            attr2 = CONV #( ms_bak-id ).
    endif.
    <ls_bak_prod>-ref = io_bak_prod.

```

```

        endmethod.
    ENDCLASS.

YBASE_CL_BAK_PROD.abap:
class YBASE_CL_BAK definition
public
final
create private
global friends YBASE_CL_BAK_REPOSITORY .
public section.
.....

"Setters
.....

methods SET_DESCR
importing
    !IV_DESCR type YBASE_E_BAK_DESCR
raising
    YCX_BASE_BAK_PROD_MAINT .
methods SET_CAPACITY
importing
    !IV_CAPACITY type YBASE_E_BAK_CAPACITY
raising
    YCX_BASE_BAK_PROD_MAINT .
methods SET_COMMENT
importing
    !IV_COMMENT type YBASE_E_BAK_COMMENT
raising
    YCX_BASE_BAK_PROD_MAINT .
.....

"Getters
.....

methods GET_ID
returning
    value(RV_ID) type YBASE_E_BAK_ID .
methods GET_DESCR
returning
    value(RV_DESCR) type YBASE_E_BAK_DESCR .
methods GET_CAPACITY
returning
    value(RV_CAPACITY) type YBASE_E_BAK_CAPACITY .
methods GET_COMMENT
returning
    value(RV_COMMENT) type YBASE_E_BAK_COMMENT .
.....

methods ADD_NEW_BAK_PROD
importing
    !IO_BAK_PROD type ref to YBASE_CL_BAK_PROD
raising
    YCX_BASE_BAK_PROD_MAINT .
methods REMOVE_BAK_PROD
importing
    !IV_NUMB type YBASE_E_BAK_PROD_NUM
returning
    value(RO_BAK_PROD) type ref to YBASE_CL_BAK_PROD
raising
    YCX_BASE_BAK_PROD_MAINT .
methods UPDATE_BAK_PROD

```

```

importing
  !IO_BAK_PROD type ref to YBASE_CL_BAK_PROD
raising
  YCX_BASE_BAK_PROD_MAINT .
methods FIND_BAK_PROD
importing
  !IV_NUMB type YBASE_E_BAK_PROD_NUM
returning
  value(RO_BAK_PROD) type ref to YBASE_CL_BAK_PROD
raising
  YCX_BASE_BAK_PROD_MAINT .
methods TO_STRUCT
returning
  value(RS_BAK) type YBASE_V_BAK .
methods BAK_PROD_TABLE
returning
  value(RT_BAK_PROD) type YBASE_TT_BAK_PROD .
methods SET_ID
importing
  !IV_ID type YBASE_E_BAK_ID
raising
  YCX_BASE_BAK_PROD_MAINT .
protected section.
private section.
types: begin of lty_s_bak_prod_ref,
  numb type ybase_e_bak_prod_num,
  ref type ref to ybase_cl_bak_prod,
  end of lty_s_bak_prod_ref.
types: lty_tt_bak_prod_ref type standard table of lty_s_bak_prod_ref.
data: ms_bak type ybase_v_bak,
  mt_bak_prod_ref type lty_tt_bak_prod_ref.
methods constructor
  importing is_bak type ybase_v_bak
  it_bak_prod type ybase_tt_bak_prod
  raising ycx_base_bak_prod_maint.
methods find_bak_prod_ref
  importing iv_numb type ybase_e_bak_prod_num
  exporting es_bak_prod_ref type lty_s_bak_prod_ref
  raising ycx_base_bak_prod_maint.
ENDCLASS.
CLASS YBASE_CL_BAK IMPLEMENTATION.
method add_new_bak_prod.
  data: lv_records_count type i,
  lo_badi_bak_valid type ref to ybase_bd_bak_valid.
  if ms_bak-id <> io_bak_prod->get_bak_id( ).
    raise exception type ycx_base_bak_prod_maint
    exporting
      textid = ycx_base_bak_prod_maint=>cx_wrong_bak
      attr1 = CONV #( io_bak_prod->get_bak_id( ) )
      attr2 = CONV #( ms_bak-id ).
  endif.
  select count( * )
    into lv_records_count
    from ybase_bak_prod
    where numb = ms_bak-id.
  if lv_records_count = ms_bak-capacity.
    raise exception type ycx_base_bak_prod_maint

```

```

        exporting
            textid = ycx_base_bak_prod_maint=>cx_max_capacity_violated
            attr1 = conv #( ms_bak-capacity ).
    endif.
read table mt_bak_prod_ref assigning field-symbol(<ls_bak_prod>)
    with key numb = io_bak_prod->get_numb( ).
if sy-subrc = 0.
    raise exception type ycx_base_bak_prod_maint
        exporting
            textid = ycx_base_bak_prod_maint=>cx_bak_prod_already_exist
            attr1 = CONV #( io_bak_prod->get_numb( ) )
            attr2 = CONV #( me->get_id( ) ).
endif.
try.
    get badi lo_badi_bak_valid.
    call badi lo_badi_bak_valid->add_new_bak_prod_valid
        exporting
            is_to_bak = ms_bak
            is_bak_prod = io_bak_prod->ms_bak_prod.
catch
    cx_badi_context_error
    cx_badi_filter_error
    cx_badi_initial_context
    cx_badi_multiply_implemented
    cx_badi_not_implemented
    cx_badi_unknown_error
    cx_badi_initial_reference
    cx_sy_dyn_call_illegal_method.
endtry.
append value #( numb = io_bak_prod->get_numb( ) ref = io_bak_prod ) to mt_bak_prod_ref.
endmethod.
method bak_prod_table.
    loop at mt_bak_prod_ref assigning field-symbol(<ls_bak_prod_ref>).
        append <ls_bak_prod_ref>-ref->to_struct( ) to rt_bak_prod.
    endloop.
endmethod.
method constructor.
    set_id( iv_id = is_bak-id ).
    set_descr( iv_descr = is_bak-descr ).
    set_capacity( iv_capacity = is_bak-capacity ).
    set_comment( iv_comment = is_bak-commentary ).
    loop at it_bak_prod assigning field-symbol(<ls_bak_prod>).
        data(lo_bak_prod) = NEW ybase_cl_bak_prod( is_bak_prod = <ls_bak_prod> ).
        append value #( numb = <ls_bak_prod>-numb ref = lo_bak_prod ) TO mt_bak_prod_ref.
    endloop.
endmethod.
method find_bak_prod.
    find_bak_prod_ref(
        exporting
            iv_numb = iv_numb
        importing
            es_bak_prod_ref = data(ls_bak_prod_ref)
    ).
    ro_bak_prod = ls_bak_prod_ref-ref.
endmethod.
method find_bak_prod_ref.
    read table mt_bak_prod_ref assigning field-symbol(<ls_bak_prod>)

```

```

        with key numb = iv_numb.
    if sy-subrc <> 0.
        raise exception type ycx_base_bak_prod_maint
        exporting
            textid = ycx_base_bak_prod_maint=>cx_bak_prod_not_found
            attr1 = CONV #( iv_numb )
            attr2 = CONV #( ms_bak-id ).
    endif.
    es_bak_prod_ref = <ls_bak_prod>.
endmethod.
method get_capacity.
    rv_capacity = ms_bak-capacity.
endmethod.
method get_comment.
    rv_comment = ms_bak-commentary.
endmethod.
method get_descr.
    rv_descr = ms_bak-descr.
endmethod.
method get_id.
    rv_id = ms_bak-id.
endmethod.
method remove_bak_prod.
    find_bak_prod_ref(
        exporting
            iv_numb = iv_numb
        importing
            es_bak_prod_ref = data(ls_bak_prod_ref)
    ).
    ro_bak_prod = ls_bak_prod_ref-ref.
    delete mt_bak_prod_ref where numb = iv_numb.
endmethod.
method set_capacity.
    ms_bak-capacity = iv_capacity.
endmethod.
method set_comment.
    ms_bak-commentary = iv_comment.
endmethod.
method set_descr.
    ms_bak-descr = iv_descr.
endmethod.
method SET_ID.
    ms_bak-id = iv_id.
endmethod.
method to_struct.
    rs_bak = ms_bak.
endmethod.
method update_bak_prod.
    read table mt_bak_prod_ref assigning field-symbol(<ls_bak_prod>)
        with key numb = io_bak_prod->get_numb( ).
    if sy-subrc <> 0.
        raise exception type ycx_base_bak_prod_maint
        exporting
            textid = ycx_base_bak_prod_maint=>cx_bak_prod_not_found
            attr1 = CONV #( io_bak_prod->get_numb( ) )
            attr2 = CONV #( ms_bak-id ).
    endif.

```

```
    <ls_bak_prod>-ref = io_bak_prod.  
endmethod.  
ENDCLASS.
```