**Репозитарий**

Репозитарий включает в себя все объекты разработки системы: программы, функциональные модули, определения таблиц базы данных и другие объекты.

Репозитарий подразделяется согласно прикладным компонентам.

В пределах прикладного компонента (например, MM) имеется несколько пакетов, содержащих соответствующие объекты для более детального логического подразделения данных.

**Инструменты поиска в репозитарии**

Информационная система репозитария используется для произвольного (т.е. не зависящего от приложения) поиска объектов репозитария, таких как все программы определенного разработчика или все функциональные модули, которые были изменены после определенной даты.

Иерархия приложений используется для поиска объектов репозитария

в зависимости от приложения. Иерархия прикладных компонентов представлена в форме древовидной структуры.

**Работа с навигатором по объектам.**

ABAP-инструментальные средства включают в себя все инструменты, необходимые для создания и обработки объектов репозитария. Эти инструментальные средства охватывают весь цикл разработки программного обеспечения.

**Работа с областью навигации.**

Списки объектов выводятся на экран в области навигации. Например, если для просмотра выбран пакет, то в этой области перечисляются все объекты репозитария, принадлежащие к указанному пакету.

**Организация разработок**

Проекты разработок выполняются в системе разработок. Объекты разработки, обработанные или созданные в проекте, переносятся в последующие системы (тестовую и/или продуктивную систему) по завершении проекта.

**Создание пакета**

1. Перейдите к *навигатору по объектам.* Выберите один из двух вариантов:
   1. В области навигации выберите тип объекта *Пакет* и введите имя пакета в поле ввода внизу. (Удостоверьтесь в соблюдении соглашений области имен клиента.) Выберите Enter. Если указанный пакет еще не существует, на экран выводится диалог для создания пакета.
   2. Выберите *Обработать объект* на первом экране *навигатора по объектам*. В диалоговом окне найдите поле определения пакета и введите имя пакета. Затем нажмите кнопку создания объекта (*F5*).
2. Установите атрибуты для создаваемого пакета.

**Создание ABAP-программы**

1. Перейдите к навигатору по объектам. Выберите один из трех вариантов:

* В области навигации выберите тип объекта *Программа* и введите имя программы в поле ввода внизу. (Удостоверьтесь в соблюдении соглашений области имен клиента.) Подтвердите ввод с помощью Enter. Если программа еще не существует, на экран выводится диалог для создания программы.
* Выведите на экран пакет, в котором необходимо создать программу. Перейти к диалогу для создания программы можно через контекстное меню для пакета или путем выбора узла *Программы*.
* Выберите *Обработать объект* на первом экране навигатора по объектам. В диалоговом окне найдите поле определения программы и введите имя программы. Затем нажмите кнопку создания объекта *(F5).*

1. В настоящем учебном курсе следует удалить индикатор *С TOP-INCLUDE*. (В противном случае исходный текст будет распределен по нескольким программам.)
2. Измените заголовок на понятный краткий текст. В качестве типа программы в этом курсе всегда следует выбирать Исполняемая программа. Все другие классы разработок необязательны. Для получения подробной информации вызовите справку F1.

**Создание транзакций**

1. В *навигаторе* по объектам вызовите список объектов для своей программы.
2. В области навигации в контекстном меню программы выберите

*Создать → Больше → Транзакция.*

1. Введите необходимый код транзакции. (Удостоверьтесь в соблюдении соглашений области имен клиента!) Присвойте краткий текст и выберите идентификатор *Программа и экран выбора* *(транзакция отчета).*
2. На следующем экране введите имя программы и выберите *Транзакция профессионального пользователя.* Выберите *GUI* активирован и установите индикатор *SAP GUI* для *Windows.*
3. Сохраните транзакцию.
4. Так как каждая транзакция является объектом репозитария, необходимо присвоить ее пакету и запросу на изменение на следующих экранах.

**Упражнения к 2 главе**

**Упражнение 1: Организация разработок**

Цели задания:

Сделав это упражнение, вы сможете:

- присваивать изменения в объектах репозитария запросу на изменение

- создавать пакеты

Условие:

Вы являетесь разработчиком в группе авиакомпаний и получаете задание разработать программы оценки для нескольких авиакомпаний. Для логического группирования объектов репозитария и их переноса необходимо создать пакет.

Решение:

- Вход в систему

1. Войдите в операционную систему, а затем в указанную систему обучения с использованием имени пользователя, указанного преподавателем.
2. Введите новый личный пароль

- Создание пакета

1. Создайте пакет ZBC400\_##. Присвойте его запросу на изменение, указанному преподавателем.
2. Присвойте все свои объекты репозитария в следующих упражнениях этому пакету и запросу на изменение.

**Упражнение 2: Разработка ABAP-программ**

Цели упражнения:

Сделав это упражнение, вы сможете:

- создавать, изменять, сохранять и обрабатывать ABAP-программы

- проверять ABAP-программы на правильность синтаксиса

- тестировать и активировать ABAP-программы

Условие:

Необходимо создать ABAP-программу, отображающую информацию относительно времени вылета для всех рейсов в списке.

Решение:

- Создание программы

1. Создайте исполняемую программу ZBC400\_##\_GS\_LIST без ”TOP-INCLUDE”.

- Изменение программы.

2. Определите рабочую область для сбора и просмотра данных (рекомендуемое имя: wa\_flight):

DATA wa\_flight TYPE sflight.

3. Запрограммируйте циклический доступ ко всем записям данных в таблице базы данных SFLIGHT:

    SELECT \* FROM sflight INTO wa\_flight.

     ...

ENDSELECT.

4. Для каждой записи данных выведите содержимое полей carrid, connid, fldate, seatsocc и seatsmax в списке:

    NEW-LINE.

  WRITE wa\_flight-carrid.

  WRITE ...

5. Проверьте программу на наличие синтаксических ошибок. Протестируйте и активируйте программу.

**Упражнение 3: Создание транзакций**

Цели упражнения:

- создавать транзакции

- определять коды транзакций в качестве личных фаворитов

Условие:

Пользователям на предприятии необходимо запускать ABAP-программы как транзакции или определять их в качестве личных фаворитов.

Решение:

- Задание 1: создание транзакции

1. Создайте код транзакции ZBC400\_##\_GS для исполняемой программы ZBC400\_##\_GS\_LIST.
2. Выполните программу путем ввода нового кода транзакции в поле команды.

- Задание 2: определение фаворитов

1. Создайте код транзакции ZBC400\_##\_GS для исполняемой программы ZBC400\_##\_GS\_LIST.
2. Выполните программу путем ввода нового кода транзакции в поле команды.

**Глава 3**

**Преобразование с помощью прозрачных таблиц**

Для каждой сущности, зафиксированной в модели данных, разработчик создает прозрачную таблицу в ABAP-словаре. Сначала она представляет собой только не зависящее от платформы описание таблицы базы данных, а не фактическую таблицу базы данных, однако при активации прозрачной таблицы в базе данных автоматически создается таблица с тем же именем.

**Определение объектов данных через ABAP-словарь. Термины**

* **DATA mycarrid TYPE s\_carr\_id.**

Возврат объекта данных (переменной) типа, указанного в элементе данных *s\_carr\_id*.

* **DATA myconnid TYPE sbc400focc-connid.**

Возврат объекта данных (переменной) того же типа, что и поле структуры sbc400focc-connid.

* **PARAMETERS pa\_carr TYPE spfli-carrid.**

Возврат поля ввода на экране выбора, а также переменной того же типа, что и поле таблицы spfli-carrid.

* **DATA wa\_focc TYPE sbc400focc.**

Возврат структурной переменной того же типа, что и структура словаря данных sbc400focc.

* **DATA wa\_spfli TYPE spfli.**

Возврат структурной переменной того же типа, что и строка таблицы spfli.

**Упражнения к 3 главе**

**Упражнение 1: Использование глобальных структур для объектов данных и диалогов**

Цели задания:

Выполнив это упражнение, вы сможете:

- определять технические имена типов данных (= описания) и объектов данных (= переменные), используемых в диалогах пользователя;

- вызывать ведомости использования для ABAP-программы.

Условие:

Необходимо проверить указанную ABAP-программу и выяснить, используются ли в ней глобальные типы данных (= описания, определенные в словаре), и если да, то какие именно.

Решение:

Шаг 1: поиск и выполнение программы

1. Просмотрите список объектов для пакета BC400 в навигаторе по объектам. Найдите программу SAPBC400WBT\_GETTING\_STARTED и просмотрите список ее объектов. Во время выполнения упражнения не выходите из режима просмотра.
2. Выполните программу для изучения принципов ее работы.

Шаг 2: анализ экрана выбора

1. Какую информацию необходимо предоставить для программы на экране выбора? (Используйте справку по полю F1.)
2. Какие значения можно ввести? (Используйте справку по вводу F4.)
3. Каково техническое имя поля ввода (имя определения в программе)?
4. Найдите эту информацию в определении в исходном тексте, а затем попробуйте найти ту же информацию в поле Техническая информация в справке к полю F1. Какая информация также содержит техническое имя поля ввода?

Шаг 3: анализ экрана

1. Каков номер просматриваемого экрана? (Для ответа на этот вопрос выберите опцию *Статус* в меню *Система* на экране.)
2. Какая информация выводится на экран? (Воспользуйтесь справкой к полю F1.)
3. Каковы технические имена полей экрана?
4. Относятся ли эти поля к структуре с типом, определенным в ABAP-словаре? (См. техническую информацию в справке по полю F1; также выполните навигацию путем двойного щелчка).

Шаг 4: анализ исходного текста программы

1. Какие объекты данных (переменные) определены в программе (см. список объектов программы)? В каком месте программы они определены?
2. Где в программе используются эти объекты данных? (Выберите опцию Ведомость использования в ABAP-редакторе.)
3. Под каким именем определено поле ввода на экране выбора программы? Включено ли это имя в список объектов программы?