Введение

Приложение «Санаторий» разрабатывалось специально для санатория «Берёзка» ОАО «Беларуськалий». То есть принцип работы ведомственного санатория изучался на примере работы «Берёзки». Впоследствии была разработана эталонная база по автоматизации работы санатория, где установлены все необходимые пространства имен (ПИ) (эти ПИ также можно найти здесь <https://code.google.com/p/gedemin-apps/source/browse/>, ветка «Санаторий»).

Приложение «Санаторий» спроектировано на основе имеющегося приложения «Гостиница». Так как предполагалось использовать имеющийся механизм работы с номерным фондом (заселение/переселение/выселение), а разрабатывать только медицинскую часть.

Однако в процессе разработки были выявлены ситуации несоответствия терминологии и механизмов работы Гостиницы и Санатория. Поэтому было принято решение сделать эти приложения независимыми друг от друга и развивать их каждый в своем направлении. При этом общие моменты всё же остались (о таких объектах свидетельствует префикс «HT» - по названию ПИ Гостиница - Hotel). Объекты, удаленные из приложения Санаторий, которые характерны для Гостиницы, описаны в . Этот перечень может быть полезен при добавлении нового функционала в приложение: во-первых, если объект имеется в перечне, его всегда можно проанализировать в контексте Гостиницы (так как он там точно присутствует), а во-вторых, объяснение отсутствия данного объекта может убедить отказаться от него и в этот раз.

В данном руководстве будет представлена структура ПИ «Санаторий» с описанием функционала и используемых объектов.

# Общая структура ПИ «Санаторий»

ПИ GS.Санаторий зависит только от ПИ Общие данные.

На рисунке 1 представлена общая схема ПИ GS.Санаторий.



Рисунок 1 – Общая схема ПИ «Санаторий»

Как видно из рисунка 1, «Санаторий» состоит из следующих подсистем:

* (содержит общие метаданные, общие справочники и документы, вспомогательные функции и пользовательские формы);
* (содержит документ тарифа и справочники, необходимые для создания нового тарифа);
* (здесь собраны все справочники для номерного фонда);
* Подсистема «Путевка» (содержит всё, что относится к созданию документа Путевки или Бронирования Путевки);
* (содержит документы по манипуляциям над путевками);
* (здесь всё, что касается медицинской части Санатория + связующая часть между Путевками и медицинской частью);
* (здесь всё, что связано со столовой в плане взаимодействия с отдыхающими, например, рассаживание в зале, назначения диет);
* (содержит основную форму приложения и обработчики её событий);
* Подсистема «Импорт из Оракла» (содержит функции по импорту данных о сотрудниках предприятия из базы данных Оракл в Гедымин) – эта подсистема опциональна;
* (здесь всё, что связано со столовой в плане калькуляций блюд, формирования заказов, составления меню и т.д.) – эта подсистема зависит от стандартного ПИ «Общепит».

На рисунке стрелками показаны зависимости внутри подсистемы, указывающие на группу подсистем, зависят от каждой из этих подсистем.

Рассмотрим каждую подсистему в отдельности.

# Подсистема «Общее»

Структура подсистемы представлена на рисунке 2.



Общее

Рисунок 2 – Структура ПИ GS.Санаторий.Общее

На рисунке здесь и далее красными стрелками представлены зависимости внутри подсистемы, желтые блоки отсутствуют в ПИ физически, но для наглядности выделены на схеме.

**Состоит из:**

* GS.Санаторий.Общее.Метаданные;
* GS.Санаторий.Общее.Справочники;
* GS.Санаторий.Общее.Документы;
* GS.Санаторий.Общее.Скрипты;
* GS.Санаторий.Общее .Формы**.**

**Зависит от:** «Общие данные».

**От данного ПИ зависят:** все ПИ из приложения «Санаторий».

**Описание:** ПИ *GS.Санаторий.Общее* предназначено для хранения общих данных, таких как домены, справочники, документы, вспомогательные пользовательские формы и функции. Такие объекты могут использоваться в любой другой подсистеме приложения Санаторий.

Ночной аудит

От приложения Гостиница сохранился функционал Ночного Аудита. Это функция, запускаемая менеджером в конце рабочего дня. Её основное предназначение – контроль над оплатой бронирований, контроль над выселением гостей, у которых подошёл день выезда и т.д. Только при выполненных действиях логическая дата системы может быть переведена на следующий день. Информация о логической дате и дате последнего аудита содержится в документе «Служебное» (HT\_SETTING): поля LOGICDATE и LASTAUDITDATE соответственно. В приложении Санаторий возможность использования этого функционала сохранена, однако он не используется в данный момент. Логическая дата соответствует текущей дате, установленной на компьютере.

Настройки приложения

Окно настроек (форма ht\_options) вызывается из Подсистема «Форма менеджера», раздел Менеджер. В настройках указываются основные объекты из справочников для правильной работы выпадающих списков в документах и правильного построения отчётов. Кроме того, настраиваются округления сумм (округляются все цены в приложении до указанных здесь значений), нумерация документов и права доступа. Для реализации прав доступа в событии формы настроек OnCreate в элемент управления usrg\_clbRights добавляются записи с уникальными идентификаторами, которые описаны в файле констант htConst. Затем в скриптах, используя объект класса разграничения прав доступа (Класс THT\_RightsHelper) и его метод CanExecuteAction, происходит проверка этого уникального идентификатора как возможность использовать ту или иную функцию. Если результат выполнения метода *True* – пользователю разрешается доступ к функции.

Описание вспомогательных функций представлено в .

Описание всех классов – .

Описание всех хранимых процедур –.

# Подсистема «Тариф»

Структура ПИ GS.Санаторий.Тариф отображено на рисунке 3.



Тариф

Рисунок 3 – Структура ПИ GS.Санаторий.Тариф

**Состоит из:** внутренних ПИ не имеет.

**Зависит от:**

**От данного ПИ зависят:**

* ;
* ».

**Описание:** ПИ *GS.Санаторий.Тариф* предназначено для хранения документа Тарифов и всех справочников, которые с ним связаны.

Типовой документ «04.Тариф» предназначен для внесения информации о стоимости путевок. Стоимость путевок варьируется в зависимости от следующих значений:

* категория отдыхающего (справочник SAN\_CLIENTCATEGORY: работник предприятия, ребенок работника предприятия, участник ВОВ и т.д.),
* тип тарифа (справочник HT\_TARRIFTYPE: разделение отдыхающих по возрасту: от 0 до 6 лет, от 6 до 18 лет, старше 18 лет);
* категория номера (справочник HT\_NUMBERTYPE: блочный с двумя комнатами, люкс и т.д.);
* дополнительное ли место в номере (за этот выбор сейчас отвечают услуги «Путевка» и «Путевка (доп.место)»). TODO: переделать так: на форме настроек добавить 2 выпадающих списка для указания из списка услуг тех, которые будем считать за «Путевку» и «Путевку на доп. Место». Там, где переменная класса HT\_Options заполняется id по руиду брать вместо руида значения из этих Lookup-ов.

Кроме того в позициях документа Тариф есть возможность указать скидку на путевку, она указывается путем ввода процента скидки, количества дней проживания по этой скидки и периода действия скидки. Например, сотрудникам предприятия предоставляется скидка 25% на 21 день в течение двух лет.

Процесс расчета стоимости Путевки описан в разделе .

Доступ к документу Тариф осуществляется из раздела Менеджер (). Права на доступ к этому документу настраиваются в *Настройках* приложения (). Пользовательская форма usrf\_ht\_tarrif\_grid применяется для указания нужного тарифа вручную. Пока в её использовании не было необходимости, так как цена у путевки строго определяется её параметрами, однако при желании можно использовать эту форму.

# Подсистема «Номерной фонд»

Содержание ПИ GS.Санаторий.Номерной фонд отображено на рисунке 4.



Номерной фонд

Рисунок 4 – Структура ПИ GS.Санаторий.Номерной фонд

**Состоит из:** внутренних ПИ не имеет.

**Зависит от:**

* ;
* .

**От данного ПИ зависят:**

* ;
* Подсистема «Медицинские услуги» (причина – справочники медицинских кабинетов);
* Подсистема «Форма менеджера».

**Описание:** ПИ *GS.Санаторий.Номерной фонд* предназначено для хранения справочных объектов для работы с номерным фондом.

Подсистема включает в себя такие справочники как «Корпуса», «Этажи», «Номера», «Места в номерах». Заселение происходит на конкретное место в номере с возможностью подселения гостей в номера. У места может быть установлен признак «дополнительное место». Такие места не отображаются в таблицах состояния мест (например, при просмотре состояния номерного фонда на дату или при заселении гостя в номер), но информация о них отображается пользователю и при отсутствии свободных «основных» мест заселение может быть произведено на дополнительное место.

В подсистеме также имеется справочник состояния номеров (USR$HT\_NUMBERSTATEDICT). В нем хранятся статусы двух видов: состояние занятости (свободен, занят, бронь) и состояние уборки номера (грязный, чистый, в процессе уборки, на ремонте и т.д.). Первый вид характерен для мест в номерах, в то время как второй – для номера в целом.

Учет состояний мест номеров ведется в журнале занятости мест номеров (USR$HT\_RES\_NUMBERSTATE), состояния уборки номера – в документе «Состояние номера» (USR$HT\_NUMBERSTATEDOC). Они будут рассмотрены в далее (Подсистема «Путевка»)

Надо отметить, что данные справочников доступны из глобального объекта класса справочников (Класс THT\_Dict).

# Подсистема «Путевка»

Структура ПИ Путевка отображена на рисунке 5.



Путевка

Рисунок 5 – Структура ПИ GS.Санаторий.Путевка

**Состоит из:**

* GS.Санаторий.П.Справочники
* GS.Санаторий.П.Документы
* GS.Санаторий.П.Отчеты
* GS.Санаторий.П.Формы

**Зависит от:**

* ;
* ;
* .

**От данного ПИ зависят:**

* ;
* ;
* ;
* .

**Описание:** ПИ *GS.Санаторий.Путевка* включает в себя все объекты, необходимые для создания документа Путевки.

Документ Путевка является базовой единицей работы санатория. Путевку выписывают, оплачивают, заселяют, по ней создают *Санаторную Книжку* и *Историю Болезни* (), выселяют и по путевкам просматривают историю посещений гостем санатория.

Для хранения Путевок используется таблица HT\_RESERVATION (используемая в приложении Гостиница как «Бронирование»). Кроме того, в санатории можно создать бронирование путевки. Бронирование путевки по хранимым данным очень похоже на саму Путевку с небольшими отличиями, поэтому Бронирования Путевок также хранятся в таблице HT\_RESERVATION.

Различаются эти документы по статусу (справочник HT\_RES\_STATEDICT, доступен через глобальный объект класса справочников – Класс THT\_Dict). Для бронирования статус заполняется значением «Бронирование», для Путевки характерны статусы:

* «Не сохранено» - используется при отмене только что созданной Путевки для информации в журнал событий (глобальный объект Класс THT\_ReservationLog, таблица HT\_RES\_LOG).
* «Новая путевка» - документ заполняется этим значением сразу при создании.
* «Ожидание заезда» - статус, в который переводится любая успешно созданная Путевка.
* «Отменено» - отмененная Путевка (удаление Путевки не производится).
* «Заселено» - статус, который получает путевка при назначении ей места в номере.
* «На выезд» (не используется) – статус, который получает путевка в день окончания периода пребывания в санатории.
* «Выселено» - статус, который получает путевка, когда текущая дата больше даты выезда, указанной в документе.

Общими признаками (свойствами) Путевки и Бронирования являются следующие:

* Заезд (ссылка на справочник SAN\_RACE);
* Гость (ссылка на справочник гостей GD\_CONTACT);
* Тип номер (ссылка на справочник HT\_NUMBERTYPE);
* Признак дополнительное ли место.

Каждый документ имеет свой номер. Причем нумерация различна для Бронирований и Путевок, она задается в Настройках Санатория ().

Для документа Путевка дополнительно характерны свойства:

* Заявка (ссылка на документ SAN\_PERSONAL\_APPLICATION);
* Бронирование (ссылка на самого себя, таблица HT\_RESERVATION);
* Категория отдыхающего (ссылка на справочник SAN\_CLIENTCATEGORY);
* Место продажи (выписки) (ссылка на GD\_CONTACT);
* Валюта;
* Тип тарифа (ссылка на HT\_TARRIFTYPE – это значение заполняется автоматически и скрыто от пользователя).

При указании Заявки или Бронирования некоторая информация автоматически заполняется значениями из этих документов.

Для документа Бронирования характерны свойства:

* Дата подтверждения брони;
* Номер (ссылка на справочник HT\_NUMBER).

Дата подтверждения брони путевки автоматически заполняется значением даты создания документа + 5 дней. Каждый раз при открытии формы менеджера () осуществляется проверка, дат подтверждения бронирований и если они меньше текущей даты, то бронь снимается.

Расчет стоимости Путевки (производится автоматически):

1. При изменении данных гостя (CONTACTKEY), заезда (RACEKEY), категории отдыхающего (CLIENTCATEGORYKEY), категории номера (NUMBERTYPEKEY) и признака дополнительного места (ISADDITIONALPLACE) происходит поиск актуального тарифа (это определяется по соотношению дат создания путевки и дат действия тарифа, при этом, если дата окончания действия не указана, тариф считается действительным) - .
2. Если тариф найден, в Путевке заполняются следующие поля:
   1. DAYRATE – стоимость одного дня проживания (соответствует значению одного из полей COSTNCU, COSTEUR, COSTRUR, COSTUSD позиции тарифа в зависимости от указанной в Путевке валюты);
   2. DAYRATEPART – стоимость одного дня проживания с учетом скидки, указанной в позиции документа тарифа (если скидки не предоставляется, это значение совпадает с DAYRATE);
   3. DAYRATESUM – стоимость всего периода проживания (соответствует значению DAYRATE, умноженному на количество дней, рассчитанное по дням заезда);
   4. TOTALPARTSUM – стоимость всего периода проживания с учетом скидки (рассчитывается процент от DAYRATESUM);
   5. PARTPRICEDAYNUMBER – количество дней, рассчитанных по скидке.

К примеру, путевка может быть выписана на 14 дней проживания, а скидка предоставляется на 21 день. Тогда при выписке путевки количество дней по скидке заполнится значением «14». При последующей выписке путевки, попадающей в период предоставления скидки (например, скидка предоставляется 1 раз за 2 года) эти 14 дней будут учтены и скидка будет предоставлена только на 7 дней, а остальные 7 дней (если заезд новой путевки также составляет 14 дней) будут рассчитаны по полной цене.

При сохранении только что созданной Путевки её стоимость автоматически заносится в документ Оплаты (). В Путевке также имеется поле INCOMETAX, оно заполняется пользователем вручную и если оно отлично от нуля, оно заносится в документ Оплаты отдельной записью.

Создание Путевки/Бронирования осуществляется из формы менеджера (), раздел Гости, пункт Действия на панели инструментов.

Изменения в Путевке

В случаях, когда в выписанную (оплаченную) путевку необходимо внести изменения, предусмотрена функция «Внести изменения в путевку» (Форма Менеджера – раздел Гости – пункт Доп.действия) – создается запись в таблице SAN\_PERMITCHANGES. С помощью этой функции можно:

* уменьшить или увеличить период пребывания гостя в санатории;
* сменить категорию номера проживания;
* сменить основное место в номере на дополнительное и наоборот;
* оформить частичное выселение (например, гость находился в больнице в течение трех дней в середине своего пребывания в санатории).

Изменения необходимы для расчета суммы к доплате/возврату (сумма заносится в документ оплат – SAN\_PAYMENT), для корректного отображения свободных мест в номерах и салфеток в столовой, а также для корректного отображения состояния путевки (важно сохранять все данные путевки на протяжении всех изменений). Для получения последних актуальных изменений предназначена хранимая процедура SAN\_P\_LASTPERMITCHANGES.

Операции над Путевкой (заселение/переселение/выселение)

Путевку со статусом «Ожидание заезда» можно заселить, со статусом «Заселено» - переселить или выселить, используя соответствующие функции на форме менеджера. Переселение и выселение можно осуществить «на будущее», указав дату переселения/выселения. При описанных действиях изменяется статусы мест номеров в журнале (HT\_RES\_NUMBERSTATE)

Телефонные переговоры

Функционал остался от приложения Гостиница, однако в Санатории пока не используется. Для учета переговоров предполагается использовать документ HT\_PHONEDOC, для хранения внешних телефонов - справочник HT\_PHONEDICT.

Список услуг

Функционал остался от приложения Гостиница, однако в Санатории пока не используется. Может быть полезным при работе с дополнительными платными услугами. Для учета услуг для путевки предполагается использовать документ HT\_RES\_SERVICELIST.

Учет состояние номеров (чистый/грязный/на ремонте)

Функционал остался от приложения Гостиница, однако в Санатории пока не используется. Для учета состояний номеров предполагается использовать документ HT\_NUMBERSTATEDOC и для изменения статусов рекомендуется работать из формы менеджера – (раздел Статусы Номеров, смотреть скрытые элементы управления).

# Подсистема «Учет путевок»

Структура ПИ Учет Путевок отображена на рисунке 6.



Учет путевок

Рисунок 6 – Структура подсистемы «Учет путевок»

**Состоит из:**

* GS.Санаторий.УП.Справочники
* GS.Санаторий.УП.Документы
* GS.Санаторий.УП.Отчеты

**Зависит от:**

* **;**
* **.**

**От данного ПИ зависит:** .

**Описание:** ПИ GS.Санаторий.Учет Путевок содержит документы по работе с путевками.

Заявки на путевки

Персональные путевки (типовой документ SAN\_PERSONAL\_APPLICATION) в основном создают для пенсионеров, в то время как сводные (типовой документ SAN\_APP) – на подразделение предприятия. По персональной заявке может быть выписана путевка. На форме менеджера (раздел Гости) в списке персональных путевок выделяются заявки-дубликаты за год (поле IYEARDUPLICATE).

Для работы над путевками разработаны документы (вызов из Формы Менеджера, раздел Менеджер):

* Договоры (типовой документ SAN\_CONTRACT);
* Счета-фактуры (типовой документ SAN\_INVOICE);
* Сведения по иностранцу (типовой документ SAN\_FOREIGNERINFO) – этот документ содержит всего два поля: номер сведения и ссылку на путевку. Документ был разработан для печати отчета, но вводить каждый раз номер сведений вручную ненадежно, к тому же со временем документ можно расширить данными;
* Товарные накладные (типовой документ SAN\_WAYBILL);
* Оплаты (типовой документ SAN\_PAYMENT).

Для построения некоторых отчетов (отчеты о сумме помощи) из данной подсистемы разработан вспомогательный документ (таблица SAN\_ACCOUNTANTDEPARTMENT), где указаны бухгалтеры и подразделения, за которые они несут ответственность (то есть по которым строят отчеты). Указав в параметрах отчета нужного бухгалтера, он построится только по необходимым подразделениям, указанным в этой таблице (альтернатива ссылки на множество).

Кроме того в отчетах о сумме помощи учитываются скидки 1 раз в месяц и в год, предоставляемые санаторием в определенной сумме. Эти суммы хранятся в константах системы Гедымин и могут быть изменены по необходимости с определенной даты. Эти константы (обращение к ним из скриптов производится по их РУИДам):

* Льготы по подоходному налогу 1 р/мес;
* Льготы по подоходному налогу 1 р/год.

# Подсистема «Санаторное питание»

Структура ПИ GS.Санаторий.Санаторное питание отображена на рисунке 7.



Санаторное питание

Рисунок 7 – Структура подсистемы «Санаторное питание»

**Состоит из:**

* GS.Санаторий.СП.Справочники
* GS.Санаторий.СП.Документы
* GS.Санаторий.СП.Формы
* GS.Санаторий.СП.Отчеты

**Зависит от:**

* **;**
* **.**

**От данного ПИ зависит:** .

**Описание:** ПИ *GS.Санаторий.Санаторное Питание* объединяет в себе объекты, относящиеся к административной работе со столовой: назначение диеты, рассаживание отдыхающих в зале питания, печать соответствующих отчетов.

Для работы со столовой предусмотрены справочники:

* Смена питания (*HT\_SHIFT*);
* Стол в зале (*HT\_TABLE*);
* Салфетка – место за столом (*HT\_DININGROOMPLACE*).

А также два документа:

* Столовая (таблица *SAN\_DINNING*);
* Журнал столовой (таблица *HT\_RES\_CANTEEN*).

При успешном заселении гостя в санаторий (оформление заселения путевки), в документе столовой (*SAN\_DINNING*) создается запись со ссылкой на эту путевку. В документе также имеется ссылка на диету (поле *DIETKEY* – ссылка на справочник диет *SAN\_DIET*) и ссылка на салфетку в зале питания (поле *DININGPLACE –* ссылка на справочник салфеток *HT\_DININGROOMPLACE*). Это поле отображает актуальную салфетку для гостя на текущую дату. Все пересаживания (пересадить за другую салфетку могут «на будущее» - датой, которая ещё не наступила, пересаживания могут быть несколько раз за период пребывания) хранятся в журнале столовой (*HT\_RES\_CANTEEN*).

Работа со столовой осуществляется из Формы Менеджера, раздел «Санаторное питание» чаще всего диет-сестрой. Она указывает необходимую гостю диету и назначает салфетку. Она же может очистить назначения и оформить «пересаживание» на другое место. Чаще всего рассаживание в зале питания осуществляется в соответствии с назначенными диетами.

В подсистеме используются следующие отчеты:

* Карточка гостя (личные данные гостя, на путевке которого установлен курсор);
* Справка на питание (отчет строится за период и представляет собой запись о количестве проживающих на каждую дату из периода);
* Состояние мест в столовой (печатается схема зала питания на указанный день: номера столов и даты, до которой занято каждое место, а также указываются места со строгими диетами - № 9 и 15). В шапке этого отчета указывается период, в течение которого отчет действителен. Действительным он считается до ближайшей даты изменения в состоянии хотя бы одно места и эта дата рассчитывается как минимальная дата из всех дат, до которой заняты места в зале.

Бывают ситуации непредвиденных изменений в проживании путевки (оформление частичного выселения или изменение дат заезда администратором санатория). В этом случае необходимо сообщить работникам столовой, что их отчеты могли устареть, так как были изменения в проживании гостей. Для этого в журнал *HT\_RES\_LOG* заносится сообщение с информацией о салфетке и причине изменений дат, а также у такой записи заполняется поле-флаг отображения для диет-сестры (поле *ISDININGROOMSHOW*). При каждом обновлении данных в разделе «Столовая» Формы Менеджера вызывается функция (ShowDinningRoomMessage) отображения сведений об изменениях для диет-сестры. Когда диет-сестра приняла к сведению предназначенное для неё сообщение (распечатала отчеты или просто запомнила изменения) она больше не нуждается в нем и по нажатию на кнопку «ОК», расположенную справа от сообщения поле отображения очищается, а флаг отображения в журнале *HT\_RES\_LOG* сбрасывается и в дальнейшем сообщения не отображаются.

Весь перечень событий, о которых сообщается диет-сестре:

* Отмена заселения (очищаются все занятости мест номеров в журнале, статус путевки становится «Ожидание Заезда»);
* Частичное выселение;
* Отмена частичного выселения (очищаются все занятости мест номеров в журнале, статус путевки становится «Ожидание Заезда»);
* Выселение (выселение оформляется на дату, то есть может быть раньше даты выезда по заезду);
* Отмена выселения (очищаются все занятости мест номеров в журнале, статус путевки становится «Ожидание Заезда»).

# Подсистема «Медицинские услуги»

Структура ПИ *GS.Санаторий.Медицинские услуги* отображена на рисунке 8.



Рисунок 8 – Структура подсистемы «Медицинские услуги»

**Состоит из:**

* GS.Санаторий.МУ.Лечащие врачи
* GS.Санаторий.МУ.Процедурные аппараты
  + GS.Санаторий.МУ.ПА.Справочники
  + GS.Санаторий.МУ.ПА.Документы
* GS.Санаторий.МУ.Учет талонов
* GS.Санаторий.МУ.План лечения
  + GS.Санаторий.МУ.ПЛ.Справочники
  + GS.Санаторий.МУ.ПЛ.Документы
* GS.Санаторий.МУ.История болезни
  + GS.Санаторий.МУ.ИБ.Справочники.Диагнозы
  + GS.Санаторий.МУ.ИБ.Справочники.Обследования
  + GS.Санаторий.МУ.ИБ.Справочники.Вспомогательные
  + GS.Санаторий.МУ.ИБ.Документы
  + GS.Санаторий.МУ.ИБ.Формы
* GS.Санаторий.МУ.Назначения процедур
  + GS.Санаторий.МУ.НП.Документы
  + GS.Санаторий.МУ.НП.Формы
* GS.Санаторий.МУ.Медикаментозное лечение
* GS.Санаторий.МУ.Санаторная книжка
* GS.Санаторий.МУ.СК.Отчеты

**Зависит от:**

* ;
* ;
* .

**От данного ПИ зависит: .**

**Описание:** ПИ *GS.Санаторий.Медицинские Услуги* содержит объекты для работы с медицинской составляющей санатория. В данной подсистеме реализованы такие функции как:

* Ведение санаторной книжки;
* Назначение лечащего врача;
* Ведение Истории Болезни гостя;
* Назначение и контроль за прохождением медицинских процедур.

Документы *Санаторная Книжка* (SAN\_SANATORIUMCARD) автоматически создается при создании документа Путевка ()

Подсистема «Путевка»и после документа История Болезни (SAN\_MEDICAL\_HISTORY), так как имеет ссылки на Путевку и Историю Болезни, являясь связующим звеном между административной частью работы санатория и медицинской частью. Таким образом эти три документа имеют связь между собой 1:1.

Из Формы Менеджера (раздел «Санаторная книжка») медицинский статист указывает первичный прием у лечащего врача в конкретной санаторной книжке, назначая тем самым одновременно лечащего врача и дату первичного приема. Назначение происходит из формы выбора доступных дней работы врачей (форма usrf\_san\_available\_doctime).

При назначении первичного приема в шапке документа *История Болезни* автоматически заполняется поле (DOCTORKEY) лечащего врача, а в позициях добавляется запись с типом приема «Первичный» (поле RECEIVINGTYPEKEY).

Данные для формы выбора первичного приема (лечащего врача) хранятся в журнале работы лечащих врачей (SAN\_DOCTIMETABLE), который заполняется из раздела *Менеджер*. Структура Журнала работы лечащих врачей следующая:

* Ссылка на лечащего врача (DOCTORKEY);
* Дата работы (DATE);
* Количество приемов на дату (RECEPTIONNUMBER);
* Ссылка на медицинский кабинет (CABINETKEY).

При подсчете доступного количества назначений на дату происходит подсчет ссылок на данную запись журнала (данный подсчет используется просто для информации, поэтому реально значение может быть меньше нуля).

История болезни

В этом документе шапка используется для статической информации (такой как обследования, анамнез и т.д.) и назначений плана лечения (медикаментозное, процедурное), а позиции представляют собой приемы врачей (узких специалистов, консультации и т.д.)

Документ История Болезни (SAN\_MEDICAL\_HISTORY) использует большое количество справочных данных, поэтому было принято решение разделить все справочники на 3 вида (на 3 ПИ):

* Диагнозы;
* Обследования;
* Вспомогательные.

Большинство из этих справочников вызывается из Исследователя (ветка «Медицинские справочники»). Значения всех справочников используются в документе *История Болезни* на разных вкладках. Значения некоторых справочников объединены в один и имеют признаком различия поле TYPE, по которому нужные значения отображаются в нужных выпадающих списках (это экономит ресурсы на создание разных справочников) и упрощает добавление новых значений.

На *Форме Менеджера* для каждого лечащего врача настроено ограничение на доступ только к своим пациентам. Если есть необходимость получить доступ к другим Историям Болезни (например, лечащий врач пациента заболел, или гость обращается к дежурному врачу), тогда пользователь обращается к функции «Поиск истории болезни», в форме поиска (usrf\_san\_search\_medhistories) врач указывает гостя, историю болезни которого он ищет и причину поиска и только после заполнения этих полей осуществляется поиск. Сам факт поиска сохраняется в документе «Журнал историй боолезни» (san\_medhistory\_log) – это необходимо, чтобы исключить и проконтролировать непредвиденные обращения врачей к историям болезни не своих пациентов.

План лечения пациента представлен на вкладке «План лечения» шапки документа История Болезни и осуществляется следующими назначениями:

* Прием настоев трав и минеральной воды (таблицы SAN\_HERBSAPPOINTMENT и SAN\_MIN\_WATER\_APPOINTMENT соответственно);
* Медикаментозное лечение (таблица SAN\_MEDICATION);
* Назначение медицинских процедур:
  + Без указания времени – перечисляются процедуры с количеством посещений за заезд (таблица SAN\_TIMELESSPROCEDURES);
  + По времени (документ «Назначения и прохождения процедур» SAN\_PROCAPPOINTMENT).

Назначения процедур по времени

Эта информация хранится в документе с позициями SAN\_PROCAPPOINTMENT. Шапка документа представляет собой один рабочий день (поле даты DATE) процедурного аппарата (поле-ссылка на справочник аппаратов PROCAPPARATUS), а позиции – время (поле TIME) с назначенным гостем на это время (поле-ссылка на документ истории болезни MEDHISTORYKEY).

В позиции также указывается следующая информация для медицинской сестры, контролирующей процесс проведения процедуры:

* продолжительность (DURATION);
* вид процедуры (поле-ссылка PROCEDUREKINDKEY на справочник видов процедур);
* часть тела применения процедуры (поле-ссылка BODYREGIONKEY на справочник зоны/части тела);
* условные единицы – информация для мед. сестры (CONVENTIONALUNITS);
* дополнительная информация (COMMENT);
* талон, выданный на данную процедуру (поле-ссылка COUPONLINEKEY на позицию документа *Учета Талонов*).

Документ назначения процедур доступен в разделе «Процедуры» Формы Менеджера как рабочее место процедурной медицинской сестры. При этом каждой медсестре (пользователю с настроенными правами на этот документ) доступны только те процедурные аппараты, которые назначены ей в документе *Ответственности за процедурный аппарат* (таблица SAN\_APPARATUSLIABILITY) – документ доступен в разделе Менеджер.

Назначения пациенту процедур по времени происходит из раздела План Лечения в документе история болезни. На форме назначения (usrf\_san\_procappointment) отображается следующая информация:

* Данные назначения (представлены в виде компонентов TgsIBGrid табличек и соответствующих бизнес объектов, связанных между собой по Master-Detail):
  + Тип процедур (справочник SAN\_PROCEDURETYPE);
  + Процедура (справочник HT\_SERVICE);
  + Вид процедуры (справочник SAN\_PROCEDUREKIND);
  + Процедурный аппарат для процедуры (справочник SAN\_PROCAPPARATUS);
  + Область (часть тела), для которой назначается процедура (справочник SAN\_BODYREGION);
  + Произвольный комментарий.
* Вспомогательная информация по назначению (представлена в виде компонентов TStaticText):
  + Количество доступных талонов для текущего лечащего врача для указанной процедуры (информация из типового документа Учета талонов SAN\_PROCCOUPON, подсчитывается с помощью представления SAN\_V\_DOCCOUPONS);
  + Количество доступных процедур на заезд для данного отдыхающего (задается двумя константами в системе Гедымин: «Курс мед. процедуры (14 дней)» и «Курс мед. процедуры (21 день)»);
  + Количество отмеченных позиций на шахматке состояния занятости процедурного аппарата;
* Шахматка состояния занятости выбранного процедурного аппарата (представлена компонентом TgsDBGrid и набором данных TClientDataSet).

На шахматке отображается следующая информация (за её вывод отвечает хранимая процедура SAN\_P\_PROCCHECKERBOARD -):

* Рабочее/нерабочее время аппарата по его актуальному графику работы.
* Есть ли закрепленный за аппаратом ответственный сотрудник санатория на данный день.
* Назначены ли другие пациенты на это время
* Назначены ли у этого пациента другие процедуры на это время.
* Назначены ли до или после этого процедуры не совместимые с назначаемой в течение времени, которое указано в справочнике совместимости процедур (SAN\_PROCEDURECOMPATIBILITY) – доступен в разделе Менеджер.

Процедурные аппараты

Каждый процедурный аппарат привязан к определенной процедуре, их справочник хранится в таблице SAN\_PROCAPPARATUS. Для каждого процедурного аппарата характерен график рабочего времени. Для графиков предназначен типовой документ «11. График работы процедурных аппаратов» (таблицы SAN\_PROCAPPWORKSCHEDULE и SAN\_PROCAPPWORKSCHEDULELINE), он доступен в разделе Менеджер. В шапке документа задается наименование графика и режим работы по дням недели, далее генерируются позиции документа (дата – рабочие интервалы). Назначение определенного графика процедурному аппарату происходит в документе «12. Работа процедурных аппаратов» (таблицы SAN\_APPSCHREL и SAN\_APPSCHRELLINE), он также доступен в разделе Менеджер.

Для процедурных аппаратов характерно назначение ответственного, обычно это процедурная медсестра, которая контролирует проведение процедур на аппарате. Ответственный за аппарат назначается в соответствующем документе (таблица SAN\_APPARATUSLIABILITY), он доступен в разделе Менеджер.

Учет талонов

Учет талонов на медицинские процедуры ведется в соответствующем документе (таблицы SAN\_PROCCOUPON и SAN\_PROCCOUPON LINE), он доступен в разделе Менеджер. В шапке документа указывается дата основного заезда (основным считается заезд, который имеет в справочнике номер), процедура и общее количество талонов на эту процедуру. В позициях расписывается, какому врачу сколько выделяется талонов из общего числа. В соответствии с решаемой задачей в документе предусмотрены следующие ограничения:

* В позициях общее число талонов не может превышать количества из шапки соответствующего документа;
* В шапке документа добавлено рассчитываемое поле BALANCE, в котором отображается доступное количество талонов;
* В случае, когда общее число талонов в позициях соответствует количеству из шапки, запись выделяется цветом.

Часто лечащему врачу при входе в систему Гедымин необходимо получить информацию о том, что у его пациентов были проведены какие-либо изменения другими пользователями (срочные консультации других специалистов, отмена приема медицинского препарата и т.д.). Для такого сообщения предусмотрено представление SAN\_V\_MEDHISTORY\_META. В нем формируются данные об изменениях, проведенных с момента предыдущего входа в систему текущего пользователя до текущего момента.

Сообщение пользователю выдается при следующих изменениях:

* Документ История болезни;
* Позиции документа История Болезни, то есть если были приемы врачей;
* Назначения процедур;
* Медикаментозное лечение.

# Подсистема «Форма менеджера»

Форма разработана для удобства и абстрагирования от стандартного интерфейса системы Гедымин, что помогает более детально разграничить права доступа к элементам базы данных, а иногда и вовсе скрыть от пользователей те или иные стандартные механизмы (такие как удаление документа Путевки, или создание документов Санаторной Книжки и Истории Болезни). Для большинства пользователей приложения Санаторий возможна работа только из данной формы.

Разделы на форме представлены вертикальным меню с наименованием предметной области (программно это меню представляет собой переключение вкладок компонента TPageControl на Форме Менеджера).

Следует отметить, что на Форме Менеджера существуют неиспользуемые разделы, например, учет состояния уборки номеров или статистические данные по номерному фонду. Они скрыты от пользователей, так как пока не используются, но при необходимости можно доработать эти разделы.

# Подсистема «Импорт из Оракла»

Данная подсистема является опциональной (необязательна к установке в пакете ПИ Санаторий). Она разрабатывалась для импорта данных о сотрудниках, пенсионерах и детях сотрудников из базы данных предприятия в Гедымин. Импорт происходит из диалоговой формы Гостя санатория по введенному табельному номеру сотрудника. Для подключения к системе Оракла предназначена функция ConnectToOracle, где указываются логин, пароль и наименование базы данных.

# Подсистема «Санаторный общепит»

Данная подсистема разработана на основе двух приложений: «Санаторий» и «Общепит», поэтому ПИ *GS.Санаторий.Санаторное Питание* зависит от них обоих.

Данная подсистема предназначена для ведения питания в столовой санатория, учитывая все особенности работы санатория.

Основные особенности работы санаторного общепита:

* Отсутствие переходящих остатков продуктов на кухне;
* Учет норм потребления продуктов питания на одного человека.

### Приложение 1

**Перечень того, что было удалено из Гостиницы для Санатория**

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Наименование функционала | Объекты | Пояснения |
| 1 | Трейсы (примечания к брони) | Таблица (usr$res\_tracelist), домен-ссылка на неё | В ходе работы выяснилось, что данный функционал избыточен |
| 2 | Гарантированность бронирования. | Справочник, домен-ссылка, поле-ссылка в документе Бронирования/Путевки | Для ведомственного санатория не характерно |
| 3 | Канал бронирования | Справочник, домен-ссылка, поле-ссылка в документе Брониварония/Путевки | Для ведомственного санатория не характерно |
| 4 | Групповое бронирование | Таблица, поле-ссылка в документе Бронирования/Путевки, функции характерные для группы броней | Для санатория не характерно |
| 5 | Список гостей | Таблица (usr$ res\_guestlist), домен, объекты на диалоговой форме Бронирования/Путевки | Для отеля характерно разделение данных о человеке, на имя которого забронирован номер и списка проживающих в этом номере (т.е. по этой брони). Для путевок же в санатории – 1 путевка=1 гость, соответственно, список гостей избыточен |
| 6 | Шахматка занятости номеров | Файл flash с отрисованной шахматкой, вкладка на Форме Менеджера, все функции и обработчики связанные с объектом web-страницы на этой вкладке | Не использовалась в санатории с самого начала. Требует доработок в соответствии с заселением не в номер, а на место в номере. Доработки необходимо производить в сторонней программе, что не удобно. |
| 7 | Форма usrf\_ht\_logview | Форма usrf\_ht\_logview | Нигде не используется |
| 8 | Отчеты | «Подтверждения бронирования», «регистрационная карточка», «баланс по номерам», «Неоплаченные бронирования», «HK статусы номеров», «Акт о выполненных работах», «Брони с доступом из внешней системы», «Журнал начислений», «Журнал отмененных бронирований», «Журнал примечаний», «Журнал статусов номеров», «Завтраков за период», «История и прогноз», «Матрица номеров», «Наличие номеров (по типам)», «Начисления по статьям», «Оплаты по типам», «Отчёт в ОВИР», «Отчет по заселениям/проживаниям», «Проверка тарифов», «Свободные номера», «Сводный отчет за день», «Статистика по компании», «Счет EN», «Счет RU», «Счет-фактура» | Не характерно для санатория или не используется |

### Приложение 2

**Вспомогательные функции приложения Санаторий**

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Входные параметры | Тип возвращаемого значения | Описание |
| hlp\_WeekDayNameRus | **ByVal** InWeekDayNumber, **ByVal** InShort | string | Возвращает день недели на русском языке по номеру *InWeekDayNumber* и в зависимости от *InShort* в сокращенной или полной форме |
| hlp\_ParamWindow | **ByVal** InParamArray | variant | Возвращает результат выполнения окна параметров *System.GetNewParamWindow(0)* |
| hlp\_NumPluralForm | **ByVal** Num,  **ByVal** Form1,  **ByVal** Form2,  **ByVal** Form5 | string | Возвращает существительное в нужном падеже в соответствии с числом *Num*. Например, hlp\_NumPluralForm(Num,”этаж”, “этажа”,”этажей”). |
| hlp\_MonthNameRus | **ByVal** InMonthNumber, **ByVal** InShort,  **ByVal** InGenitiveCase | string | Возвращает название месяца на русском языке по номеру *InMonthNumber,* в зависимости от *InShort* полную или короткую форму, в зависимости от *InGenitiveCase* в родительном или именительном падеже |
| hlp\_LAT2CYR | **ByVal** sCYR | string | Преобразовывает строку латинских символов в строку кирилличных символов |
| hlp\_iif | **ByVal I**nCondition**,**  **ByVal** InFirst**,**  **ByVal** InSecond | variant | Возвращает значение *InFirst*,если *InCondition* верно и *InSecond* – в противном случае |
| hlp\_GetStrFieldValue | **ByVal** TableName**, ByVal** ListFieldName**, ByVal** KeyFieldName**, ByVal** KeyValue | string | Возвращает первое значение результата выполнения запроса:  SELECT *ListFieldName*  FROM *TableName*  WHERE *KeyFieldName = KeyValue* |
| hlp\_GetIntFieldValueTr | **ByVal** TableName**, ByVal** ListFieldName**, ByVal** KeyFieldName**, ByVal** KeyValue**,**  **ByVal** Tr | integer | Возвращает результат выполнения запроса:  SELECT *ListFieldName*  FROM *TableName*  WHERE *KeyFieldName = KeyValue*  в транзакции *Tr* |
| hlp\_GetIntFieldValue | **ByVal** TableName**, ByVal** ListFieldName**, ByVal** KeyFieldName**, ByVal** KeyValue | integer | Возвращает результат выполнения запроса:  SELECT *ListFieldName*  FROM *TableName*  WHERE *KeyFieldName = KeyValue* |
| hlp\_GetDateTimeFieldValueTr | **ByVal** TableName**, ByVal** ListFieldName**, ByVal** KeyFieldName**, ByVal** KeyValue**,**  **ByVal** Tr | Date | Возвращает первое значение результата выполнения запроса:  SELECT *ListFieldName*  FROM *TableName*  WHERE *KeyFieldName = KeyValue*  в транзакции *Tr* |
| hlp\_GetDateTimeFieldValue | **ByVal** TableName**, ByVal** ListFieldName**, ByVal** KeyFieldName**, ByVal** KeyValue | Date | Возвращает первое значение результата выполнения запроса:  SELECT *ListFieldName*  FROM *TableName*  WHERE *KeyFieldName = KeyValue* |
| hlp\_FormatNumber | **ByVal** NumericValue | string | Преобразовывает число *NumericValue* в отформатированную строку |
| hlp\_FormatDateTime | **ByVal** InFormat**,**  **ByVal** InDateTime | string | Преобразует дату *InDateTime* в формат, указанный в *InFormat* |
| hlp\_CYR2LAT | **ByVal** sCYR | string | Преобразует строку кирилличных символов в латинские символы |
| hlp\_AppendString | **ByRef** InSourceString**,**  **ByVal** InAppendedString**,**  **ByVal** InDivider | Результат в строке *InSourceString* | Соединяет строки *InSourceString* и *InAppendedString* через разделитель *InDivider* |

### Приложение 3

**Классы приложения Санаторий**

## Класс THT\_Dict

Справочники, которые часто используются в процессе работы Санатория. Класс разработан для более быстрого доступа к данным.

Свойства

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Доступ | Тип данных | Описание |
| FFloorDict | private | Dictionary | Этажи. В словаре key – id из справочника, item – объект класса Класс THT\_Floor |
| FNumberTypeDict | private | Dictionary | Типы номеров. В словаре key – id из справочника, item – объект класса Класс THT\_NumberType |
| FNumberDict | private | Dictionary | Номера. В словаре key – id из справочника, item – объект класса Класс THT\_Number |
| FPayTypeDict | private | Dictionary | Типы оплаты. В словаре key – id из справочника, item – объект класса Класс THT\_PayType |
| FTarrifTypeDict | private | Dictionary | Типы тарифов. В словаре key – id из справочника, item – объект класса Класс THT\_TarrifType |
| FloorDict | public (property get) | Dictionary | Возвращает свойство Этажи |
| NumberTypeDict | public (property get) | Dictionary | Возвращает свойство Типы номеров |
| NumberDict | public (property get) | Dictionary | Возвращает свойство Номера |
| PayTypeDict | public (property get) | Dictionary | Возвращает свойство Типы оплаты |
| TarrifTypeDict | public (property get) | Dictionary | Возвращает свойство Типы тарифов |

Методы

|  |  |  |
| --- | --- | --- |
| Наименование | Доступ | Описание |
| Refresh |  | Обновление всех данных в свойствах-справочниках |
| Class\_Initialize | private | Инициализация свойств объектов |
| Class\_Terminate | private | Завершение работы объекта |

## Класс THT\_Floor

Этаж. Структура класса соответствует структуре справочника HT\_Floor. Используется в классе справочников (Класс THT\_Dict)

Свойства

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Доступ | Тип данных | Описание |
| Key | public | integer | Ключ записи. Аналог полю ID в таблице справочника HT\_Floor |
| Name | public | string | Наименование этажа. Аналог полю USR$NAME в таблице справочника HT\_Floor |
| Disabled | public | integer (0,1) | Признак неактивности записи. Аналог полю Disabled в таблице справочника HT\_Floor |

## Класс THT\_NumberType

Тип номера. Структура класса соответствует структуре справочника HT\_NumberType. Используется в классе справочников (Класс THT\_Dict)

Свойства

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Доступ | Тип данных | Описание |
| Key | public | integer | Ключ записи. Аналог полю ID в таблице справочника HT\_NumberType |
| Name | public | string | Наименование типа. Аналог полю USR$NAME в таблице справочника HT\_ NumberType |
| ShortName | public | string | Краткое наименование типа. Аналог полю USR$SHORTNAME в таблице справочника HT\_ NumberType |
| CountRoom | public | integer | Количество комнат в номере. |
| CountPlace | public | Integer | Количество мест в номере. |
| CountAddPlace | public | Integer | Количество дополнительных мест в номере. |
| IsVirtual | public | Integer(0,1) | Виртуальный ли номер |

## Класс THT\_Number

Номер. Структура класса соответствует структуре справочника HT\_Number. Используется в классе справочников (Класс THT\_Dict)

Свойства

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Доступ | Тип данных | Описание |
| Key | public | integer | Ключ записи. Аналог полю ID в таблице справочника HT\_Number |
| Name | public | string | Наименование номера. Аналог полю USR$NAME в таблице справочника HT\_ Number |
| TypeKey | public | integer | Тип номера. Аналог полю-ссылке USR$TYPEKEY в таблице справочника HT\_ Number |
| CountRoom | public | integer | Количество комнат в номере |
| CountPlace | public | Integer | Количество мест в номере |
| CountAddPlace | public | Integer | Количество дополнительных мест в номере |
| IsVirtual | public | Integer(0,1) | Виртуальный ли номер |

## Класс THT\_PayType

Тип оплаты. Структура класса соответствует структуре справочника HT\_PAYTYPE. Используется в классе справочников (Класс THT\_Dict)

Свойства

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Доступ | Тип данных | Описание |
| Key | public | integer | Ключ записи. Аналог полю ID в таблице справочника HT\_PAYTYPE |
| Name | public | string | Наименование типа оплаты. Аналог полю USR$NAME в таблице справочника HT\_PAYTYPE |
| Kind | public | Integer (1,2,3) | Тип платежного средства. Аналог полю-ссылке USR$TYPE в таблице справочника HT\_ PAYTYPE. Значения:   * 1 – Наличный * 2 – Пластиковая карта * 3 – Безналичный. |

## Класс THT\_TarrifType

Тип тарифа. Структура класса соответствует структуре справочника HT\_TARRIFTYPE. Используется в классе справочников (Класс THT\_Dict)

Свойства

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Доступ | Тип данных | Описание |
| Key | public | integer | Ключ записи. Аналог полю ID в таблице справочника HT\_TARRIFTYPE |
| Name | public | string | Наименование типа оплаты. Аналог полю USR$NAME в таблице справочника HT\_TARRIFTYPE |
| OnlyAdmin | public | integer | Признак доступности только администратору. Аналог полю USR$ ONLYADMIN в таблице справочника HT\_ TARRIFTYPE |

## Класс THT\_NumberStateDict

Справочник состояния номеров.

Свойства

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Доступ | Тип данных | Описание |
| FDictionary | private | Dictionary | Состояния номеров (мест номеров). В словаре key – id из справочника, item – объект Класс THT\_NumberState |

Методы

|  |  |  |
| --- | --- | --- |
| Наименование | Доступ | Описание |
| GetFreeState | public | Возвращает объект Класс THT\_NumberState, соответствующий состоянию «Свободно» |
| GetBookState | public | Возвращает объект Класс THT\_NumberState, соответствующий состоянию «Забронировано» |
| GetBusyState | public | Возвращает объект класса Класс THT\_NumberState, соответствующий состоянию «Занят» |
| GetMoveState | public | Возвращает объект Класс THT\_NumberState, соответствующий состоянию «Переселен» |
| GetCleanState | public | Возвращает объектКласс THT\_NumberState, соответствующий состоянию «Чистый» |
| GetDirtyState | public | Возвращает объект Класс THT\_NumberState, соответствующий состоянию «Грязный» |
| GetRepairState | public | Возвращает объект Класс THT\_NumberState, соответствующий состоянию «На ремонте» |
| GetInspectedState | public | Возвращает объект Класс THT\_NumberState, соответствующий состоянию «Инспектирован» |
| GetCleaningState | public | Возвращает объект Класс THT\_NumberState, соответствующий состоянию «Чистый» |
| GetByKey (ByVal InKey) | public | Возвращает объект состояния номера (Класс THT\_NumberState) по ключу |
| Count | public | Количество состояний номеров в списке |
| GetItemArray | public | Получить массив объектов из списка |
| Refresh | public | Обновление всех данных в свойствах-справочниках |
| RevRGB (red, green, blue) | private | Возвращает длинное целое число (Long) собранное по значениям Red, Green, Blue |
| Class\_Initialize | private | Инициализация свойств объектов |
| Class\_Terminate | private | Очищение словаря для справочника при уничтожении объекта класса |

## Класс THT\_NumberState

Состояние номера. Структура класса соответствует структуре справочника HT\_NUMBERSTATEDICT. Используется в классе справочника статусов номеров (Класс THT\_NumberStateDict).

Свойства

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Доступ | Тип данных | Описание |
| Key | public | Integer | Ключ записи. Аналог полю ID в таблице справочника HT\_NumberStateDict |
| Name | public | String | Наименование состояния. Аналог полю USR$NAME в таблице справочника HT\_NumberStateDict |
| Color | public |  | Цвет для VBScript |
| FlashColor | public |  | Цвет для Flash (не используется) |

## Класс THLP\_ChooseItems

Хелпер для выбора записей из таблицы.

Свойства

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Доступ | Тип данных | Описание |
| FItems | private | Array | Выбранные значения (записи) |
| Items | public (property get) | Array | Возвращает выбранные записи |
| Items (ByVal Value) | public (property let) | Array | Устанавливает значение FItems = Value |
| FCanCreate | private | Integer (0,1) | Флаг, позволять ли создавать записи в таблицу на форме |
| CanCreate | public (property get) | Integer (0,1) | Возвращает флаг создания записей |
| CanCreate(ByVal Value) | public (property let) | Integer (0,1) | Устанавливает значение FCanCreate = Value |
| FAllowNothingSelected | private | Integer (0,1) | Флаг, разрешать ли отсутствие выбора |
| AllowNothingSelected | public (property get) | Integer (0,1) | Возвращает флаг разрешения на отсутствие выбора |
| AllowNothingSelected(ByVal Value) | public (property let) | Integer (0,1) | Устанавливает флаг FAllowNothingSelected = Value |
| SingleItem | public (property get) |  | Возвращает первое значение из массива выбранных (FItems(0)) |

Методы

|  |  |  |
| --- | --- | --- |
| Наименование | Доступ | Описание |
| ChooseSingle(ByVal FormCaption,  ByVal RelationName,  ByVal KeyField,  ByVal NameField,  ByVal WhereClause) | public | Открывает вспомогательную форму для выбора записей usrf\_hlp\_chooseitems по переданным настройкам. Возвращает *True*, если выбор на форме был подтвержден. Разрешает выбрать только одно значение |
| ChooseMulti (ByVal FormCaption,  ByVal RelationName,  ByVal KeyField,  ByVal NameField,  ByVal WhereClause) | public | Открывает вспомогательную форму для выбора записей usrf\_hlp\_chooseitems по переданным настройкам. Возвращает *True*, если выбор на форме был подтвержден. Разрешает выбрать множество значений |
| Class\_Initialize | private | Инициализация свойств объектов |
| Class\_Terminate | private | Завершение работы объекта |

## Класс THT\_Const

Класс для доступа к некоторым константам приложения Санаторий.

Свойства

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Доступ | Тип данных | Описание |
| FReservationNumberSeparator | private | string | Разделитель номера бронирования (по умолчанию «/») |
| ReservationNumberSeparator | public (property get) | string | Возвращает разделитель номера |
| FSettleNumber | private | integer | Размещение (не используется) |
| SettleNumber | public (property get) | Integer | Возвращает размещение |
| FSettlePlace | private | Integer | Не используется |
| SettlePlace | public (property get) | Integer | Не используется |
| FSRVFrequencySingle | private | integer | Периодичность оказания услуг «Разово» |
| SRVFrequencySingle | public (property get) | Integer | Возвращает периодичность оказания услуг «Разово» |
| FSRVFrequencyDaily | private | Integer | Периодичность оказания услуг «Каждый день» |
| SRVFrequencyDaily | public (property get) | Integer | Возвращает периодичность оказания услуг «Каждый день» |
| FSRVFrequencyPerHour | private | Integer | Периодичность оказания услуг «Каждый час» |
| SRVFrequencyPerHour | public (property get) | Integer | Возвращает периодичность оказания услуг «Каждый час» |
| FSRVFrequencyPerMinute | private | Integer | Периодичность оказания услуг «Каждую минуту» |
| SRVFrequencyPerMinute | public (property get) | Integer | Возвращает периодичность оказания услуг «Каждую минуту» |
| FSRVFrequencyPerSecond | private | Integer | Периодичность оказания услуг «Каждую секунду» |
| SRVFrequencyPerSecond | public (property get) | integer | Возвращает периодичность оказания услуг «Каждую секунду» |
| FReservationLogFields | private | Dictionary | Поля для журналирования изменений в бронировании |
| ReservationLogFields | public (property get) | Dictionary | Возвращает поля для журналирования |

Методы

|  |  |  |
| --- | --- | --- |
| Наименование | Доступ | Описание |
| FillReservationLogFields | private | Заполнение словаря FReservationLogFields значениями Поле – Нарименование |

## Класс THT\_Log

Класс для удобного логирования в журнал HT\_LOG.

Методы

|  |  |  |
| --- | --- | --- |
| Наименование | Доступ | Описание |
| InternalAdd (ByVal InRecordKey,  ByVal InDescription,  ByVal LogType,  ByVal InOwnerform) | private | Внутренний для класса метод сохранения записи в таблицу HT\_LOG. |
| Add (ByVal InRecordKey,  ByVal InDescription,  ByVal InOwnerform) | public | Добавляет запись в журнал |
| AddDetail(ByVal InRecordKey,  ByVal InDescription,  ByVal InOwnerform) | public | Добавить дочернюю запись в журнал. Используется в функции Ночной Аудит |
| AddSystem(ByVal InRecordKey,  ByVal InDescription,  ByVal InOwnerform) | public | Добавить запись в журнал. Чаще всего используется с кодом ошибки. |

## Класс THT\_NumberStateHelper

Хелпер по работе с номерным фондом (состояния уборки номеров).

Свойства

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Доступ | Тип данных | Описание |
| FWorkTransaction | private |  | Разделитель номера бронирования |

Методы

|  |  |  |
| --- | --- | --- |
| Наименование | Доступ | Описание |
| AddHKRepairState(ByVal InNumberKey,  ByVal InDocumentKey,  ByVal InDateTimeBegin,  ByVal InDateEnd,  ByVal Description,  ByVal InTr) | Public | Добавление состояния ремонта номера |
| AddHKState(ByVal InNumberKey,  ByVal InNumberState,  ByVal InDocumentKey,  ByVal InDateTime,  ByRef InTr) | Public | Добавление произвольного состояния номера |
| Internal\_AddHKState(ByVal InNumberKey,  ByVal InDocumentKey,  ByVal InStateKey,  ByVal DateTimeBegin,  ByVal DateEnd,  ByVal Description, ByVal InTr) | Private | Внутренняя процедура добавления состояния |
| AddHKState\_OnExcept(ByRef Tr) | private | Функция отката транзакции для функции *Internal\_AddHKState* |
| GetHKState(ByVal InNumberKey) | public | Получить ID состояния номера по его ID |
| GetNumberUsedBy(ByVal InNumberKey,  ByVal InDateBegin,  ByVal InDateEnd) | public | Используется ли переданный номер в переданном периоде. На номер может быть бронь или он может быть заселен, тогда метод вернет ID брони. Если метод вернет -1, значит на этот номер можно делать новую бронь |
| Class\_Initialize | private | Инициализация транзакции, используемой в объекте |
| Class\_Terminate | private | Удаление объекта транзакции, созданной для обекта |

## Класс THT\_Options

Глобальные настройки Санатория, задаваемые на форме Настроек в Форме Менеджера.

Свойства

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Доступ | Предполагаемый тип данных | Описание |
| STORAGE\_PATH | private | string | Ветка в хранилище для хранения глобальных настроек приложения (по умолчанию "Options\Sanatorium") |
| FInstitution | private | integer | Ведомство |
| Institution | public (property get) | integer |  |
| FProcFolder | private | integer | Папка для медицинских процедур |
| ProcFolder | public (property get) | integer |  |
| FDoctorsUserGroup | private | integer | Группа пользователей лечащих врачей |
| DoctorsUserGroup | public (property get) | integer |  |
| FDefaultTime | private | date | Время заселения по умолчанию |
| DefaultTime | public (property get) | date |  |
| FDepartureTime | private | date | Время выселения по умолчанию |
| DepartureTime | public (property get) | date |  |
| FWarnLogicDateNotDate | private | integer | Предупреждать о несовпадении реальной и логической даты |
| WarnLogicDateNotDate | public (property get) | integer |  |
| FReservationNumberMask | private |  | Маска номера нового бронирования |
| ReservationNumberMask | public (property get) |  |  |
| FRoundBYRToValue | private | currency | Округление в НДЕ |
| RoundBYRToValue | public (property get) | currency |  |
| FRoundUSDToValue | private | currency | Округление в USD |
| RoundUSDToValue | public (property get) | currency |  |
| FRoundEURToValue | private | currency | Округление в EUR |
| RoundEURToValue | public (property get) | currency |  |
| FRoundRURToValue | private | currency | Округление в RUR |
| RoundRURToValue | public (property get) | currency |  |
| FResidenceServiceKey | private | integer | Ключ услуги проживания |
| ResidenceServiceKey | public (property get) | integer |  |
| FAddResidenceServiceKey | private | integer | Ключ услуги дополнительного проживания |
| AddResidenceServiceKey | public (property get) | integer |  |
| FAddPlaceServiceKey | private | integer | Ключ услуги «дополнительное место» |
| AddPlaceServiceKey | public (property get) | integer |  |
| FBreakfastServiceKey | private | integer | Ключ услуги «Завтрак» |
| BreakfastServiceKey | public (property get) | integer |  |
| FPhoneCallServiceKey | private | integer | Ключ услуги звонков |
| PhoneCallServiceKey | public (property get) | integer |  |
| FPermitServiceKey | private | integer | Ключ услуги «Путевка» |
| PermitServiceKey | public (property get) | integer |  |
| FPermitAddPlaceServiceKey | private | integer | Ключ услуги «Путевка (доп. место)» |
| PermitAddPlaceServiceKey | public (property get) | integer |  |
| FTeleProviderDataFolderPath | private |  | Путь к папке с данными от телефонной компании |
| TeleProviderDataFolderPath | public (property get) |  |  |
| FTeleProviderDataFileMask | private |  | Маска для поиска файлов от телефонной компании |
| TeleProviderDataFileMask | public (property get) |  |  |
| FTeleInternalDataFolderPath | private |  | Путь к папке с данным от внутренней системы учета звонков |
| TeleInternalDataFolderPath | public (property get) |  |  |
| FTeleInternalDataFileMask | private |  | Маска для поиска файлов с данными от внутренней системы учета звонков |
| TeleInternalDataFileMask | public (property get) |  |  |
| FRestFrontConnection | private |  | Строка подключения к БД кассовой системы |
| RestFrontConnection | public (property get) |  |  |
| FRestFrontUser | private |  | IB пользователь |
| RestFrontUser | public (property get) |  |  |
| FRestFrontPassword | private |  | IB пароль |
| RestFrontPassword | public (property get) |  |  |
| FPermitLastNumber | private | integer | Текущий номер путевки |
| PermitLastNumber | public (property get) | integer |  |
| FPermitNumberInc | private | integer | Прирост для номера путевки |
| PermitNumberInc | public (property get) | integer |  |
| FPermitCheckNumber | private | integer | Контролировать номер путевки |
| PermitCheckNumber | public (property get) | integer |  |
| FReservationLastNumber | private | integer | Текущий номер для бронирования |
| ReservationLastNumber | public (property get) | integer |  |
| FReservationNumberInc | private | integer | Прирост для номера бронирования |
| ReservationNumberInc | public (property get) | integer |  |
| FReservationCheckNumber | private | integer | Контролировать номер бронирования |
| ReservationCheckNumber | public (property get) | integer |  |
| FCurrentUserRights | private |  | Права доступа |
| CurrentUserRights | public (property get) |  |  |
| FStatNumberAvailDayCount | private | integer | Сколько дней отображать в наличии номеров |
| StatNumberAvailDayCount | public (property get) | integer |  |

Методы

|  |  |  |
| --- | --- | --- |
| Наименование | Доступ | Описание |
| ShowOptions | Public | Отобразить окно настроек |
| TestCorrect (ByVal AForm) | Public | Проверка правильности введенных на форме значений |
| LoadOptionsToForm(ByVal AForm) | Private | Заполнить форму настроек текущими значениями |
| SaveOptionsFromForm(ByVal AForm) | private | Считать из формы настроек измененные значения |
| LoadUserRightsToForm(ByVal AForm) | private | Загрузить права пользователей в форму |
| SaveUserRightsFromForm(ByVal AForm) | private | Сохранить настройки прав пользователей из формы |
| LoadFromStorage | private | Загрузить настройки из хранилища |
| SaveToStorage | private | Сохранить настройки в хранилище |
| RefreshDependentVars | private | Обновить переменные, зависимые от настроек |
| Class\_Initialize | private | Инициализация объекта: значение STORAGE\_PATH, считывание данных из хранилища |

## Класс THT\_ReservationLog

Логирование действий, произведенных над путевкой/бронированием.

Методы

|  |  |  |
| --- | --- | --- |
| Наименование | Доступ | Описание |
| InternalAdd(ByVal InDocumentKey,  ByVal InDescription,  ByVal InStateKey,  ByVal InCancelReasonKey,  ByVal InActionType) | Private | Внутренний для класса метод создания записи в журнале бронирования (HT\_RES\_LOG) |
| Add(ByVal InDocumentKey,  ByVal InDescription,  ByVal InStateKey) | Public | Добавить запись в журнал |
| AddTyped(ByVal InDocumentKey,  ByVal InDescription,  ByVal InStateKey,  ByVal InActionType) | Public | Добавить типизированную запись в журнал. В качестве типа может выступать одно из значений: проживание продлено, проживание сокращено |
| AddWithCancelReason(ByVal InDocumentKey,  ByVal InDescription,  ByVal InStateKey,  ByVal InCancelReasonKey) | Public | Добавить запись в журнал с указанием ссылки на причину отмены создания путевки |

## Класс THT\_ReservationStateDict

Справочник состояний путевки/бронирования HT\_RES\_STATEDICT.

Свойства

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Доступ | Тип данных | Описание |
| FDictionary | private | Dictionary | Словарь всех значений из справочника, где Key – ID записи из справочника, Item – объект класса Класс THT\_ReservationState |

Методы

|  |  |  |
| --- | --- | --- |
| Наименование | Доступ | Описание |
| GetNewState | Public | Новая путевка |
| GetArrivalWaitState | Public | Ожидание заезда |
| GetEvictedState | Public | Выселено |
| GetSettledState | Public | Заселено |
| GetCanceledState | public | Отменено |
| GetNotSavedState | public | Не сохранено |
| GetBookedState | Public | Забронировано |
| GetDepartureWaitState | Public | На выезд |
| GetPermitBookingState | Public | Бронирование путевки |
| GetSettleStateList | Public | Получить список ключей состояния проживания |
| GetNumberInUseStateList | Public | Получить список ключей состояний, в которых номер считается используемым |
| GetByKey(ByVal InKey) | Public | Получить объект состояния номера по ключу |
| Count | Public | Количество состояний номеров в списке |
| GetItemArray | Public | Получить массив объектов из списка |
| Refresh | Public | Обновить данные в кэше |

## Класс THT\_ReservationState

Состояние номера. Структура класса соответствует структуре справочника HT\_RES\_STATEDICT. Используется в классе справочника статусов путевок/бронирований (Класс THT\_ReservationStateDict).

Свойства

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Доступ | Тип данных | Описание |
| Key | public | integer | Ключ записи. Аналог полю ID в таблице справочника HT\_RES\_STATEDICT |
| Name | public | string | Наименование состояния. Аналог полю USR$NAME в таблице справочника HT\_RES\_STATEDICT |
| SettleType | public | string | Тип один из: "before", "settle", "after", "cancel", в соответствии со статусом путевки/бронирования. |

## Класс THT\_RightsHelper

Проверка прав доступа, назначенных в настройках санатория.

Методы

|  |  |  |
| --- | --- | --- |
| Наименование | Доступ | Описание |
| CanExecuteAction (ByVal ActionID) | Public | Разрешено ли выполнять действие. ActionID – константа на определенное действие. Проверка осуществляется, если текущий пользователь НЕ администратор |

## Класс THT\_Variables

Переменные приложения Санаторий

Свойства

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Доступ | Тип данных | Описание |
| STORAGE\_PATH | private | string | Ветка в хранилище для хранения переменных (по умолчанию «Options\Sanatorium») |
| FIBSQLReservationNumber | private | Объект TIBSQL | Запрос на получение нового номера бронирования |
| FIBSQLPermitNumber | private | Объект TIBSQL | Запрос на получение нового номера путевки |
| FLogicDate | private | Date | Логическая дата |
| LogicDate | public (property get) | Date | Возвращает логическую дату |
| LogicDate(ByVal Value) | public (property let) | Date | Устанавливает логическую дату |
| FLastNightAuditDate | private | Date | Дата проведения последнего ночного аудита |
| LastNightAuditDate | public (property get) | Date | Возвращает дату последнего ночного аудита |
| LastNightAuditDate(ByVal Value) | public (property let) | Date | Устанавливает дату последнего ночного аудита |

Методы

|  |  |  |
| --- | --- | --- |
| Наименование | Доступ | Описание |
| WriteParamToDB (ByVal ParamName,  ByVal ParamValue) | Private | Запись параметров в БД |
| GetNextReservationNumber | Public | Получить новый номер бронирования |
| GetNextPermitNumber | Public | Получить новый номер путевки |
| Refresh | Public | Загрузка данных |
| Class\_Initialize | private | Инициализация объекта, загрузка данных из хранилища |
| Class\_Terminate | private | Удаление объектов запросов, используемых в классе. |

## Класс TMath

Общие математически действия

Свойства

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Доступ | Тип данных | Описание |
| FRACTION | private | decimal | Вспомогательное значение (очень маленькое) для корректного округления. Например, 0,5 должно быть 1, а не 0. |

Методы

|  |  |  |
| --- | --- | --- |
| Наименование | Доступ | Описание |
| RoundToDecimal  (**ByVal** AValue,  **ByVal** AToDecimal) | public | Округлить *AValue* до *AToDecimal*  знака после запятой |
| RoundToValue  (**ByVal** AValue,  **ByVal** AToValue) | public | Округлить *AValue* до значения *AToValue* |
| Ceil  (**ByVal** AValue) | public | Округлить *AValue* до целого вверх |
| Floor  (**ByVal** AValue) | public | Округлить *AValue* до целого вниз |
| Max  (**ByVal** AFirst,  **ByVal** ASecond) | public | Максимальное число из двух |
| Min  (**ByVal** AFirst,  **ByVal** ASecond) | public | Минимальное число из двух |
| Class\_Initialize | private | Инициализация FRACTION |

### Приложение 4

**Хранимые процедуры**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование | Входные данные | Выходные данные | Описание | ПИ |
| HT\_P\_  GET\_HKSTATE | FORDATE – состояние на дату,  ROOMKEY – ID номера | STATEKEY – ID состояния | Состояние номера (чистый, грязный и т.д.) на дату | GS.Санаторий.П.Документы |
| HT\_P\_  GET\_HKSTATELIST | FORDATE – состояние на дату | ROOMKEY – ID номера,  STATEKEY – ID состояния | Список всех номеров и их состояний на дату | GS.Санаторий.П.Документы |
| HT\_P\_  GET\_LOGIC\_POSTINGSUM | RESERVATIONKEY – ID Путевки,  DATEBEGIN – дата начала периода,  DATEEND – дата окончания периода | POSTSUMNCU – начисления,  PAYSUMNCU – оплата | Начисления и оплаты по бронированию (для Гостиницы, в Санатории пока не используется) | GS.Санаторий.П.Документы |
| HT\_P\_  GET\_NUMBERS | Нет | NUMBERKEY – ID комнаты,  NUMBERNAME – наименование комнаты | Список не виртуальных номеров | GS.Санаторий.Номерной фонд |
| SAN\_P\_  FULLAGE | BDAY – дата рождения,  ONDATE – на дату | FULLYEARAGE – число лет | Возвращает число полных лет человека на дату | GS.Санаторий.Общее.Метаданные |
| SAN\_P\_  GETACTUALPROCWORKSCHE | FORDATE – на дату,  APPKEY – ID процедурного аппарата | SCHEDULEKEY – ID графика | Актуальный график работы процедурного аппарата | GS.Санаторий.МУ.ПА.Документы |
| SAN\_P\_  GETDEPARTMENTPATH | ID – ID подразделения,  NOTSHOWNKEY – отображать ли само это подразделение | PATH – путь к подразделению | Полный путь к подразделению по дереву справочника (разделитель «-->») | GS.Санаторий.Общее.Метаданные |
| SAN\_P\_  GETLASTVISITDATE | CONTACTKEY – ID человека, CANCELSTATEKEY – ID статуса «Отменено» путевки,  FORDATE – на дату | LASTVISITDATE – дата последнего посещения | Дата последнего посещения санатория (учитывается соответствующая дата из карточки гостя и все неотмененные путевки человека) | GS.Санаторий.П.Документы |
| SAN\_P\_  GETMINDATEDRAPP | FREESTATEKEY – ID состояния «Свободно» места в столовой,  DOCUMENTKEY – ID документа путевки | MINDATE – минимальная дата назначения места в столовой | Самая ранняя дата назначения места в столовой | GS.Санаторий.СП.Документы |
| SAN\_P\_  GETPREVIOUSLOGON\_ALL | нет | PREVIOUSDATE – дата и время входа в систему,  CONTACTKEY – ID контакта | Дата и время последнего входа в систему пользователей, которых отслеживаем в системе по этой теме (тема – вход в систему Log on) | GS.Санаторий.Общее.Метаданные |
| SAN\_P\_  GETWORKINTERVALS | SCHEDULEKEY – ID графика работы проц. аппарата,  THEDAY – дата | TBEGIN – время начала периода,  TEND – время окончания периода | Рабочие интервалы из графика работы процедурных аппаратов | GS.Санаторий.МУ.ПА.Документы |
| SAN\_P\_  LASTPERMITCHANGES | USR$PERMITKEY – ID путевки,  FORDATE – дата, на которую определяются последние изменения | Данные после изменений:  USR$DATEBEGIN – дата заезда,  USR$DATEEND – дата выезда,  USR$CHANGESBEGINDATE – дата начала действия изменений,  USR$CHANGESENDDATE – дата окончания действия измеенний,  USR$CLIENTCATEGORYKEY – категория гостя (не используется),  USR$NUMBERTYPEKEY – тип номера,  USR$ISADDITIONALPLACE – дополнит. ли место,  CHANGESKEY – ID записи об изменениях,  USR$ISPARTIALEVICTION – частичное ли выселение | Последние изменения путевки.  Категорию гостя изменить нельзя, так как сложно рассчитать использованные и доступные дни по скидке и за полную стоимость до и после изменений. | GS.Санаторий.П.Документы |
| SAN\_P\_  PERMITDAYCOUNTPERIOD | BCHECKPERIOD – начало периода для проверки,  ECHECKPERIOD – окончание периода для проверки,  BPERIOD – начало периода, в котором проверяем,  EPERIOD – окончание периода, в котором проверяем | ALLDAYCOUNT – всего дней заезда,  PERIODDAYCOUNT – дни заезда в периоде, BEFOREPERIODDAYCOUNT – дни заезда до периода, AFTERPERIODDAYCOUNT – дни заезда после периода | Разбивает нужные даты (чаще всего это даты заезда путевки) в контексте интересующего периода:   * количество дней заезда (ДЗ) общее; * количество ДЗ внутри периода; * количество ДЗ до периода; * количество ДЗ после периода. | GS.Санаторий.Общее.Метаданные |
| SAN\_P\_  PROCCHECKERBOARD | DATEBEGIN – дата начала периода для расчетов,  DATEEND – дата окончания периода для расчетов,  SERVICEKEY – ID текущей процедуры,  CURMEDHIST – ID документа Истории Болезни | APPKEY – ID процедурного аппарата,  APPNAME – наименование процедурного аппарата,  COLDATE – дата (одна из дат периода),  COLTIME – время,  COMMENT – пояснения к определенному времени,  COLOR – цифра, обозначающая тип информации,  DOCLINEKEY – ID документа назначения процедуры | Набор всей информации для таблицы-шахматки назначения медицинских процедур.  Возможные значения COLOR и COMMENT:  1 – нерабочее время аппарата по графику,  2 – на данное время назначен другой пациент,  3 – у текущего пациента назначена другая процедура,  7 – если это вариант 3 и эта назначенная процедура = входящей SERVICEKEY,  4 – если у текущего пациента уже назначена ДО этого времени процедура, несовместимая с текущей,  5 - если у текущего пациента уже назначена ПОСЛЕ этого времени процедура, несовместимая с текущей,  6 – нет ответственного за аппарат. | GS.Санаторий.МУ.Санаторная книжка |