

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное
образовательное учреждение
высшего профессионального образования
ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

А. В. ЩЕГОЛЕВА
А. Н. КИРИЛЕНКО

ПРОЕКТИРОВАНИЕ
ИНФОРМАЦИОННОЙ
СИСТЕМЫ:
СТРУКТУРНЫЙ ПОДХОД

Учебное пособие
для студентов вузов

Петрозаводск
Издательство ПетрГУ
2013

первый из которых связан с обработкой карт брони, а второй — с обработкой карт визита. В этой предметной области возможна только одна передача данных от процесса бронирования процессу работы с картами визита. Для обеспечения этих процессов будет использовано только одно хранилище данных — реляционная база данных. В этом случае для описания проекта информационной системы мы используем следующий инструментарий: дерево функций и диаграммы «Сущность — связь».

Для более сложных предметных областей с большим числом взаимосвязанных процессов, несколькими ролями пользователей, несколькими хранилищами данных следует начать процесс проектирования с построения диаграммы потоков данных (DFD), связывающей все процессы и хранилища данных и постепенно детализирующей каждый процесс. Для случая проектирования систем реального времени также необходимо построить модели, описывающие поведение системы во времени.

Таким образом, начнем с построения модели дерева функций информационной системы, затем перейдем к описанию структуры базы данных и на последнем этапе опишем спецификации функций, представленных в дереве функций, с учетом спроектированной схемы отношений базы данных.

3 ПОСТРОЕНИЕ МОДЕЛИ ДЕРЕВА ФУНКЦИЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Создание модели дерева функций начинается с описания корневой вершины, название которой определяет основное назначение разрабатываемой информационной системы. Далее выделяются основные группы функций в предметной области, они соответствуют основным бизнес-процессам предметной области. Далее в рамках каждого бизнес-процесса выделяются подпроцессы — функции по работе с информационными объектами предметной области. Процесс детализации продолжается до простейших функций, например, занесения фамилии клиента в поле ввода. В последнюю очередь в дерево функций добавляются служебные функции, напрямую несвязанные с бизнес-процессами предметной области. Обычно названия функций в дереве функций являются либо глаголом в неопределенной форме, либо отглагольным существительным.

Для рассматриваемой предметной области корнем дерева функций будет вершина с названием «Автоматизация деятельности дежурного администратора». От этой вершины отходят вершины, представляющие основные группы функций, связанные с информационными объектами предметной области: *Факт бронирования*, *Факт визита*, *Черный список*, *Комната*, *Услуга*. Тогда дочерними вершинами корневой вершины дерева функций будут: *Ведение карт брони*, *Ведение карт визита*, *Работа с Черным списком*. Информационные объекты *Комната* и *Услуга* являются вспомогательными, поэтому работу с ними поместим в вершину *Работа со справочниками*, которая будет включать справочники: *Категории*, *Комнаты*, *Услуги*. Отдельно выделим вершину *Ведение отчетов*. Хотя отчеты связаны с информационным объектом *Факт визита*, но с точки зрения бизнес-процессов предметной области отчеты представляют собой самостоятельный информационный объект. Аналогично в отдельную вершину поместим функцию *Работа с листом заселения*. Добавим еще одну группу функций, которая обязательно присутствует в любой информационной системе, — *Служебные функции*. Эта группа включает функции по администрированию ИС, настройке, ведению архивов и другие вспомогательные функции, которые не связаны с бизнес-процессами предметной области.

Корневая часть получившегося дерева функций представлена на рис. 3.1.



Рис. 3.1. Корневая часть дерева функций

Далее необходимо детализировать каждую группу функций. Так как, во-первых, дерево получится очень большим, а, во-вторых, некоторые функции, например для карт визита и карт брони, звучат одинаково, то не будем приводить полное дерево функций, а опишем подробно только функции, связанные с картой визита. Графическое представление этого поддерева не поместится на одном листе, поэтому заменим графическое представление иерархическим списком, представленным ниже. Глубина дерева должна быть максимальной. *В дерево необходимо включить все функции.* Если какая-то функция не будет включена, она не будет и реализована в информационной системе. Описание каждого листа дерева должно быть достаточно подробным.

Рассмотрим вершину дерева функций *Ведение карт визита*. В первую очередь следует выделить две группы действий: работа со списком карт визита и работа с одной картой визита. Работа со списком карт визита включает просмотр списка, сортировку карт визита, фильтрацию карт визита и предоставление возможности выбора одной карты визита. Можно также предусмотреть выбор нескольких карт визита, но мы не будем усложнять проектируемую информационную систему. Каждая из этих операций должна быть описана более подробно: в каком виде представлять список, по каким полям выполнять фильтрацию и т. д. Фрагмент дерева функций для вершины *Ведение карт визита* представлен на рис. 3.2.

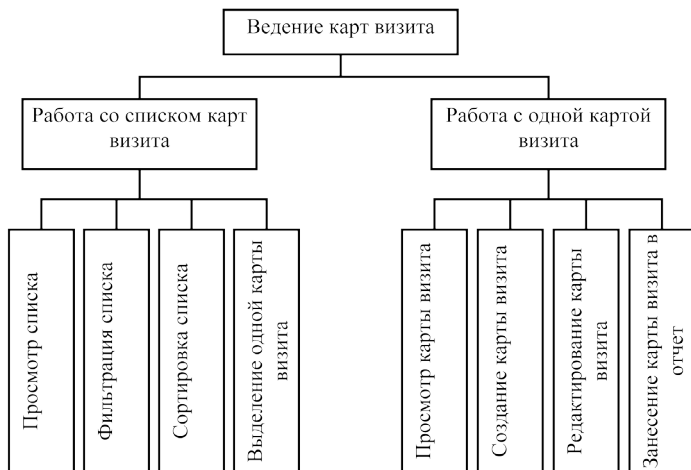


Рис. 3.2. Фрагмент дерева функций от вершины *Ведение карт визита*

Более подробное описание каждой функции представлено ниже в виде списка.

1. Ведение карт визита (второй уровень дерева):

1) Работа со списком карт визита (третий уровень дерева):

а) Просмотр списка карт визита в табличном виде (четвертый уровень дерева). Таблица содержит столбцы: *Номер карты визита, Номер комнаты, ФИО клиента, Дата приезда, Дата отъезда.*

б) Фильтрация списка (поиск карты визита):

- по состоянию карты визита (активные, закрытые, все) (пятый уровень дерева);
- по фамилии клиента;
- по дате приезда;
- по номеру карты визита;
- по номеру комнаты;
- по номеру карты брони;
- по номеру счета.

в) Сортировка карт визита (по возрастанию, по убыванию):

- по номеру карты визита;
- по ФИО клиента;
- по номеру комнаты;
- по дате приезда;
- по дате отъезда.

г) Выделение одной карты визита.

2) Работа с одной картой визита (третий уровень дерева):

а) Создать карту визита (четвертый уровень дерева) — карта визита не сохраняется, если не заполнено хотя бы одно из обязательных полей, в этом случае выводится сообщение «Заполните, пожалуйста, ...» с перечислением незаполненных полей:

- 1 Номер карты визита (пятый уровень дерева) — значение генерируется автоматически и недоступно для редактирования. Если карта визита находится в состоянии «Активна» надпись имеет зеленый фон, если в состоянии «Закрыта» — красный фон.
- 2 Номер карты брони заполняется автоматически, доступное заполнение через выбор карты брони из списка карт брони.

- 3 Фамилия — поле ввода — количество символов не более 40 — первая буква автоматически переводится в верхний регистр — разрешен ввод только букв (русские и латинские) и дефиса — поле обязательно для заполнения.
- 4 Имя — поле ввода — количество символов не более 35 — первая буква автоматически переводится в верхний регистр — разрешен ввод только букв (русские и латинские) и дефиса — поле обязательно для заполнения.
- 5 Отчество — поле ввода — количество символов не более 35 — первая буква автоматически переводится в верхний регистр — разрешен ввод только букв (русские и латинские) и дефиса.
- 6 Серия и номер паспорта — поле ввода — количество символов не более 15 — разрешен ввод только цифр и букв (русских и латинских) — поле обязательно для заполнения.
- 7 Кем выдан паспорт — поле ввода — разрешен ввод только букв (русских), цифр, точки и дефиса.
- 8 Когда выдан паспорт — календарь с маской ввода «__.__.____».
- 9 Контактная информация — поле ввода — количество символов не более 255 — допустим ввод любых символов.
- 10 Дата приезда — календарь с маской ввода «__.__.____». По умолчанию записана текущая дата — поле обязательно для заполнения. При попытке ввести дату, которая является более поздней по отношению к дате отъезда, выводится сообщение «Дата приезда должна быть раньше даты отъезда», при этом автоматически заносится предыдущая по отношению к дате отъезда дата.
- 11 Дата отъезда — календарь с маской ввода «__.__.____». По умолчанию записана следующая за текущей дата — поле обязательно для заполнения. При попытке ввести дату, которая является более ранней по отношению к дате приезда, выводится сообщение «Дата отъезда должна быть позднее даты приезда», при этом автоматически заносится следующая за датой приезда дата.
- 12 Количество суток — поле ввода — рассчитывается автоматически по дате приезда и дате отъезда — недоступно для редактирования.

- 13 Номер комнаты — раскрывающийся список — выбор только из списка — поле обязательно для заполнения — список фильтруется в зависимости от значения поля *Категория комнаты*.
 - 14 Категория — раскрывающийся список — выбор только из списка — по умолчанию «2А» — при выборе категории список комнат автоматически обновляется, заносится первая в списке комната в поле *Номер комнаты*.
 - 15 Стоимость — поле ввода, заполняется автоматически при смене категории номера — недоступно для редактирования — отображение значения в формате «____, __р.» с пробелами для тысяч.
 - 16 Сумма — поле ввода, заполняется автоматически при смене дат, категории номера — недоступно для редактирования — отображение значения в формате «____, __р.» с пробелами для тысяч.
 - 17 Номер счета заполняется автоматически при выписке счета на проживание — недоступно для редактирования.
 - 18 Оплачено — заполняется автоматически при выписке счета на проживание — недоступно для редактирования.
- б) Просмотр карты визита (четвертый уровень дерева) — все поля недоступны для редактирования.
- в) Редактировать карту визита:
- 1 Редактировать основные поля карты визита (пятый уровень дерева):
 - Фамилия (шестой уровень дерева);
 - Имя;
 - Отчество;
 - Дата приезда;
 - Дата отъезда;
 - Номер комнаты;
 - Категория;
 - Серия и номер паспорта;
 - Кем выдан паспорт;
 - Когда выдан паспорт;
 - Контактная информация.
 - 2 Работа с оказанными услугами (пятый уровень дерева):
 - 1) Работа со списком оказанных услуг (шестой уровень дерева):

- а) Просмотр списка оказанных услуг (седьмой уровень дерева).
 - б) Сортировка:
 - по дате оказания услуги (восьмой уровень дерева);
 - по названию услуги;
 - по стоимости;
 - по количеству;
 - по сумме;
 - по номеру счета.
 - в) Выделение оказанной услуги.
 - г) Выделение списка оказанных услуг.
 - д) Занесение услуг в счет.
 - е) Выбор неоплаченных услуг.
- 2) Работа с одной оказанной услугой (шестой уровень дерева):
- а) Создать услугу (седьмой уровень дерева):
 - 1 Дата оказания услуги — календарь с маской ввода «__.__.____». По умолчанию записана текущая дата — поле обязательно для заполнения (восьмой уровень дерева).
 - 2 Название услуги — раскрывающийся список — выбор только из списка — по умолчанию значение первой услуги в таблице tblService — при выборе услуги обновляется информация о стоимости и сумме.
 - 3 Стоимость — поле ввода — заполняется автоматически — поле недоступно для редактирования.
 - 4 Количество — поле ввода — допускается ввод только вещественных чисел, включающих не более 5 цифр, из них две цифры после запятой — значение по умолчанию 1 — при изменении обновляется информация о сумме — поле обязательное для заполнения.
 - 5 Сумма — поле ввода — заполняется автоматически по количеству и стоимости.
 - б) Редактировать услугу (седьмой уровень дерева).
 - в) Удалить услугу.
 - г) Просмотр услуги.
 - д) Сохранить услугу.

- е) Занесение услуги в счет — выполняется при создании счета.
 - ж) Занесение отметки об оплате — выполняется на уровне триггера.
 - з) Удаление отметки об оплате.
3. Работа со счетами (пятый уровень дерева):
- 1) Работа со списком счетов (шестой уровень дерева):
 - а) Просмотр списка счетов (седьмой уровень дерева).
 - б) Сортировка:
 - по дате выписки счета;
 - по сумме;
 - по номеру счета.
 - в) Выделение счета.
 - 2) Работа с одним счетом (шестой уровень дерева):
 - а) Создать счет на проживание (седьмой уровень дерева):
 - 1 Номер счета (восьмой уровень дерева) — поле ввода — значение генерируется автоматически по номеру последнего созданного счета в системе.
 - 2 Дата создания счета — календарь с маской ввода «__.__.____». По умолчанию записана текущая дата — поле обязательно для заполнения.
 - 3 Сумма — поле ввода — значение заполняется автоматически по сумме проживания — недоступно для редактирования.
 - б) Создать счет на услуги (седьмой уровень дерева):
 - 1 Номер счета — поле ввода — значение генерируется автоматически по номеру последнего созданного счета в системе.
 - 2 Дата создания счета — календарь с маской ввода «__.__.____». По умолчанию записана текущая дата — поле обязательно для заполнения.
 - 3 Сумма — поле ввода — значение заполняется автоматически при добавлении/удалении услуг, включаемых в счет — недоступно для редактирования.
 - в) Удалить счет.
 - г) Просмотр счета — просмотр печатной формы счета.
 - д) Печать счета.

- 4) Закрывать карту визита — выполняется проверка на наличие неоплаченных услуг или проживания, автоматически формируется счет со всеми неоплаченными услугами, в случае сохранения счета, карта визита переводится в состояние «Закрота», иначе закрытие карты отменяется (пятый уровень дерева).
- г) Удалить карту визита (четвертый уровень дерева) — выполняется проверка на состояние карты визита, если карта закрыта, то выводится запрос на подтверждение, при положительном ответе пользователя карта удаляется. В случае если карта находится в состоянии «Активна», проверяется наличие счетов и оказанных услуг, если есть хотя бы один счет, то операция удаления отменяется с выводом сообщения «Удаление активной карты невозможно при наличии оплаченных счетов». Если есть оказанные услуги, то операция удаления отменяется с выводом сообщения «Удаление активной карты невозможно при наличии оказанных услуг».
- д) Занести карту визита в отчет — выполняется на уровне триггера.

Рассмотрим еще одну группу функций.

1 Работа с листом заселения (второй уровень дерева):

- 1) Просмотр листа заселения в виде таблицы, в строках которой представлена информация по одной комнате, а в столбцах — по одной дате (третий уровень дерева). По горизонтали даты находятся в диапазоне от самой ранней даты приезда визита (брони), хранящегося в базе данных, до самой поздней даты отъезда брони (визита), хранящегося в базе данных. По вертикали представлены все комнаты гостиницы. Начиная с текущей даты, клетки раскрашены следующим образом: красный — номер занят, желтым — номер забронирован. В клетке представлена информация: фамилия, имя клиента.
- 2) Фильтрация:
 - а) по категории номеров (четвертый уровень дерева);
 - б) по номеру комнаты;
 - в) по дате.
- 3) Поиск:
 - а) по фамилии клиента.
- 4) Перемещение:
 - а) по горизонтали — по датам;

- б) по вертикали — по номерам.
- 5) Выделение ячейки — вывод информации:
 - для бронирования: ФИО клиента, контактная информация, дата приезда, дата отъезда.
 - для визита: ФИО клиента, контактная информация, дата приезда, дата отъезда, сумма проживания, сумма всех оказанных услуг, факт оплаты проживания, сумма оплаченных услуг.

Одной из служебных функций является функция авторизации пользователя.

- 1 Авторизация дежурного администратора (третий уровень дерева). При авторизации дежурного администратора, его идентификатор запоминается и подставляется автоматически при сохранении данных в тех таблицах базы данных, где это требуется. В строке состояния отображается ФИО дежурного администратора.

Аналогичным образом описываются остальные функции дерева функций.

Можно было бы добавить еще одну корневую группу функций для удобства работы пользователей в некоторых редко выполняемых бизнес-процессах. Это функции *Работа со счетами*. Информационный объект *Счет* является частью информационного объекта *Факт визита*, и работа со счетами, относящимися к одному визиту, учтена в дереве функций. Но в некоторых ситуациях возникает необходимость видеть все выписанные счета без разбивки их по картам визита. Именно для этого можно предусмотреть функции для работы со списком всех счетов.

Полезным делом при проектировании будет создание пользовательских сценариев — описаний выполнения пользователем бизнес-процессов с помощью информационной системы. Опишем для примера несколько бизнес-процессов предметной области. В качестве упражнения попробуйте проверить: для всех ли действий пользователя в дереве функций найдется соответствующая функция.

- 1 *Создание карты брони*: пользователь проверяет наличие свободных номеров указанной клиентом категории номеров, проверяет по ФИО клиента наличие клиента в черном списке, заполняет карту брони.

- 2 *Аннулирование карты брони:* пользователь выполняет поиск карты брони (по ФИО, дате приезда, дате оформления), выполняет операцию аннулирования.
- 3 *Внесение изменений в карту брони:* пользователь выполняет поиск карты брони (по ФИО, дате приезда, дате оформления), вносит изменения в карту брони.
- 4 *Создание карты визита:* пользователь выполняет поиск карты брони (по ФИО, дате приезда, дате оформления), если клиент ранее забронировал номер; для найденной карты брони копирует данные из карты брони в новую карту визита; если клиент ранее не бронировал номер, то пользователь проверяет наличие свободных номеров указанной пользователем категории номеров, заполняет карту визита.
- 5 *Редактирование карты визита:* пользователь выполняет поиск карты визита (по ФИО, номеру комнаты), вносит изменения в карту визита.
- 6 *Заказ услуги:* пользователь выполняет поиск карты визита (по ФИО, номеру комнаты), добавляет информацию об услуге.
- 7 *Редактирование услуги:* пользователь выполняет поиск карты визита (по ФИО, номеру комнаты), вносит изменения в информацию об услуге, если она не оплачена.
- 8 *Оплата услуг:* пользователь выполняет поиск карты визита (по ФИО, номеру комнаты), получает список неоплаченных услуг, выбирает из полученного списка услуги, которые клиент собирается оплатить, формирует новый счет, печатает счет, заносит в карту визита отметки об оплате услуг.
- 9 *Закрытие карты визита:* пользователь выполняет поиск карты визита (по ФИО, номеру комнаты), проверяет отсутствие оплаты проживания, формирует новый счет на проживание, печатает счет, заносит в карту визита отметку об оплате проживания, получает список всех неоплаченных услуг, формирует новый счет на все неоплаченные услуги, печатает счет, заносит в карту визита отметки об оплате услуг, заносит отметку о закрытии карты визита.
- 10 *Поиск просроченных карт брони:* пользователь получает список карт брони, для которых дата приезда ранее текущей даты, пользователь выбирает из полученного списка карты брони, которые будут занесены в черный список, или выделяет все карты брони,

заносит отметки о вхождении в черный список тем клиентам, для которых были выделены карты брони, заносит отметку о просроченной карте брони для всех карт брони из полученного списка.

- 11 *Просмотр листа заселения гостиницы:* пользователь выбирает дату начала просмотра листа заселения или категорию комнат, или номер комнаты, или ФИО клиента, или занятые/забронированные/свободные и формирует лист заселения в соответствии с заданными условиями.

Анализируя пользовательские сценарии, можно оценить скорость выполнения действий пользователем, выявить неэффективные моменты в распределении функций между пользователем и информационной системой.

Например, обратим внимание на действия пользователя в рамках бизнес-процесса *Оплата услуг*. Пользователь заносит в карту визита отметку об оплате услуги. В отсутствие информационной системы пользователю, действительно, необходимо внести соответствующие отметки после оформления счета на оплату услуг. При проектировании информационной системы мы должны облегчить работу дежурного администратора, запроецировав автоматическое выполнение некоторых действий. К таким действиям, которые в рассматриваемой предметной области можно выполнять автоматически без участия пользователя, как раз относится внесение отметок об оплате. Поэтому это действие будет выполнять не пользователь, а информационная система при занесении услуги в счет. Удобно реализовать это действие в виде триггера для соответствующей таблицы базы данных.

Аналогично можно поступить и с другими действиями пользователя. В качестве примера хотелось бы еще отметить функцию дежурного администратора по составлению им сменного отчета. Создание отдельной функции информационной системы для формирования отчета является излишним действием, ведь отчет является вторичным документом предметной области, создаваемым на основе первичных документов — карты визита и счета, поэтому в информационной системе формирование отчета (заполнение отчета) будет происходить автоматически при создании новой карты визита и при создании нового счета. Автоматическое формирование отчета можно реализовать в виде триггеров на операцию Insert для соответствующих таблиц базы данных. При создании новой записи в такой таблице проверяется существование отчета на текущую дату для текущего администратора.

Если отчета пока нет, то он создается, для создаваемой записи заполняется поле, содержащее отметку о включении карты визита или счета в отчет.

4 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

4.1 Построение инфологической модели предметной области

В первую очередь построим инфологическую модель предметной области, описывающую предметную область без учета используемой в дальнейшем модели данных. Это означает, что по инфологической модели в дальнейшем может быть построена как реляционная, так и объектная база данных. Для описания инфологической модели воспользуемся моделью «сущность-связь» в нотации Баркера.

На первом шаге выделим классы (сущности) предметной области. Под классом понимается совокупность объектов предметной области, обладающих одинаковыми свойствами.

В рассматриваемой предметной области выделим следующие классы: *Клиент* — человек, который либо забронировал номер в гостинице, либо проживает в гостинице. Свойствами класса являются: ФИО, паспортные данные, контактная информация, наличие в черном списке (рис. 4.1);

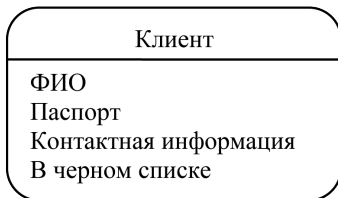


Рис. 4.1. Класс *Клиент*

Комната — номер в гостинице. Так как слово «номер» может иметь много значений в рассматриваемой предметной области, например номер номера, то вместо словосочетания «номер гостиницы» будем использовать для названия класса слово «комната». Свойствами класса являются: номер комнаты, этаж (рис. 4.2);