

МОДЕРНИЗАЦИЯ СТРУКТУРЫ УПРОЩЁННОЙ СИСТЕМЫ  
ТЕХНИЧЕСКОГО МЕНЕДЖМЕНТА СУДОВ

В работах [1 – 3] предложена упрощённая система технического менеджмента для судоходных компаний Planner, которая выполняет следующие функции:

- планирование и учёт работ по техническому обслуживанию судов; складской учёт;
- контроль и учёт трудовой деятельности экипажа;
- организация закупок для судов;
- учёт расходов средств на техническое обслуживание и закупки;
- отправка отчётных данных в офис судоходной компании.

Сопровождение этой системы показало, что в каждом конкретном случае по желанию пользователей возникает необходимость развить отдельные функции системы. Однако применённая в системе традиционная структура затрудняла привлечение к развитию системы новых программистов, так как им было сложно разобраться в ранее осуществлённых изменениях.

Поэтому была поставлена задача сделать архитектуру системы более гибкой и удобной в сопровождении или, другими словами, более адаптивной к изменениям. Также было решено максимально сохранить, или даже улучшить, основное преимущество системы – простоту её освоения пользователями.

Для решения этой задачи были использованы паттерны проектирования [4], под которыми понимают модули с типовыми решениями проектных задач. Существует каталог паттернов, где приводится подробная информация о цели каждого паттерна, его функциях и ситуациях, в которых они уместны, а также возможных последствиях (как положительных, так и отрицательных).

В модернизированной системе используется стандартная трёх-уровневая архитектура:

- база данных;
- уровень обработки данных;
- пользовательский интерфейс.

База данных является реляционной и реализована с помощью системы MS SQL Server.

На уровне обработки данных задачи разбиты на подзадачи, а те в свою очередь, – на ещё более мелкие подзадачи до тех пор, пока зада-

чи не станут атомарными. В теории баз данных под атомарными задачами подразумеваются операции выборки и обновления данных.

Чтобы внести изменения в базу или сделать выборку, необходимо осуществить подключение к базе данных, создать и открыть транзакцию, выполнить работу с данными, закрыть транзакцию. Выборка является атомарной операцией. Обновление данных сводится к трём атомарным операциям: добавление, редактирование, удаление записи в конкретной таблице. В этой цепочке действий изменяемым звеном являются только операции выборки и обновления данных, а открытие-закрытие транзакции и подключение к БД – это стандартные процедуры. В соответствии с этим на уровне обработки данных построена иерархия классов, показанная на рис. 1.

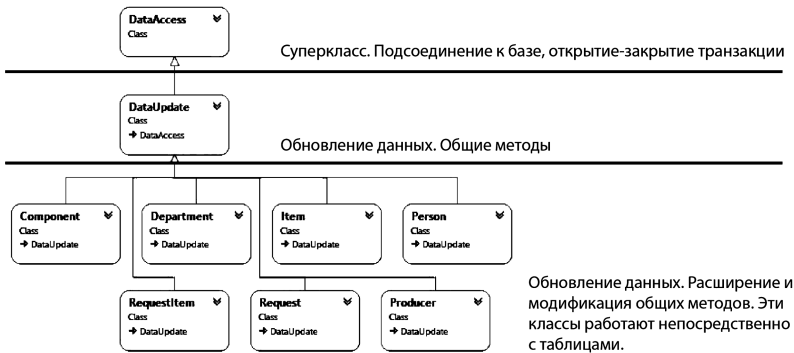


Рис 1. Иерархия классов уровня обработки данных.

DataAccess – это суперкласс, на который возложены функции подключения к базе и открытие-закрытие транзакции. Кроме того, в нём описаны методы выборки и обновления данных. Выборка данных, как самая простая операция, может вызываться из любого наследуемого класса.

Класс DataUpdate унаследован от DataAccess. В нём прописаны методы обновления данных, которые при необходимости могут быть переопределены в наследуемых классах.

Классы для работы с таблицами предназначены для атомарных действий работы с базой данных (выборка, добавление, редактирование, удаление записей).

Необходимо отметить, что переопределять методы обновления в наследуемых классах иерархии третьего уровня необязательно. Это основной принцип паттерна Декоратор (Decorator) – возможность расширения кода без его модификации. На практике это выглядит

следующим образом: добавляя таблицу в базу данных и соответствующий ему класс в систему, разработчик уже имеет методы вставки и обновления данных, ему остаётся лишь при необходимости динамически наделить этот класс новыми возможностями.

Для уровня пользовательского интерфейса существует множество программ, в основу которых заложена грамотно спроектированная архитектура и большой функционал. Однако такие программы часто имеют недостаточно понятный пользователю интерфейс. Чтобы избежать этой проблемы, в системе PlannerS были разработаны такие стандарты:

1) Списки – самый простой стандарт, в котором присутствуют две формы – собственно сам список и окно для добавления/редактирования элемента (рис. 2). Применяется для ведения основной справочной информации (персонал, механизмы, производители, единицы измерения и прочее).



Рис 2. Уровень пользовательского интерфейса: списки

2) Операции – более сложный стандарт, который применяется, если необходимо выполнить какие-то более сложные логически связан-

ные операции (работа с заявками, приход, расход, инвентаризация, планирование, наработка). В нём часто используются фильтры, режимы сортировки и действия, специфические для данной операции (рис. 3).

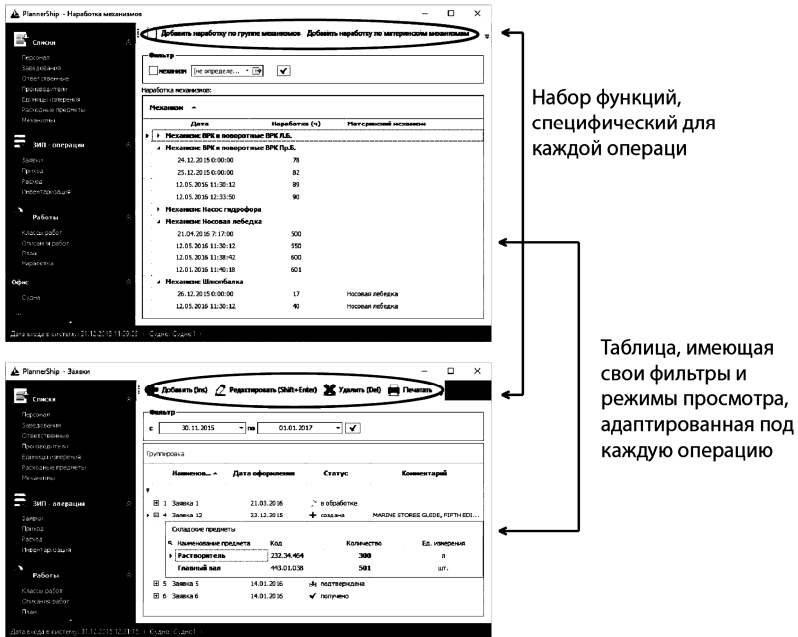


Рис. 3. Уровень пользовательского интерфейса: операции

3) Отчёты. Вся работа по построению отчёта осуществляется программистом "за кадром", поэтому этот стандарт является простым для освоения. В нём реализованы функции, привычные для пользователя из офисных приложений, а также экспорт в популярные форматы (.txt, .pdf, .csv, HTML, Excel, .png, .jpg и др.) (рис. 4).

Каждый стандарт имеет свой специфический набор функций. В процессе программирования все бизнес-требования "подгоняются под стандарт" и реализуются. Это позволяет сделать работу программы предсказуемой, что облегчает работу и программисту, и пользователю.

Освоив работу одного элемента из каждого стандарта, пользователь может уверенно работать с другими элементами этой категории. Обратимся к конкретному примеру: в системе существует около 10

списков. Если пользователь изучит работу со списком Персонал, то он сможет работать и с остальными списками. При расширении системы и добавлении новых элементов программы не нужно тратить время на дополнительное обучение персонала.

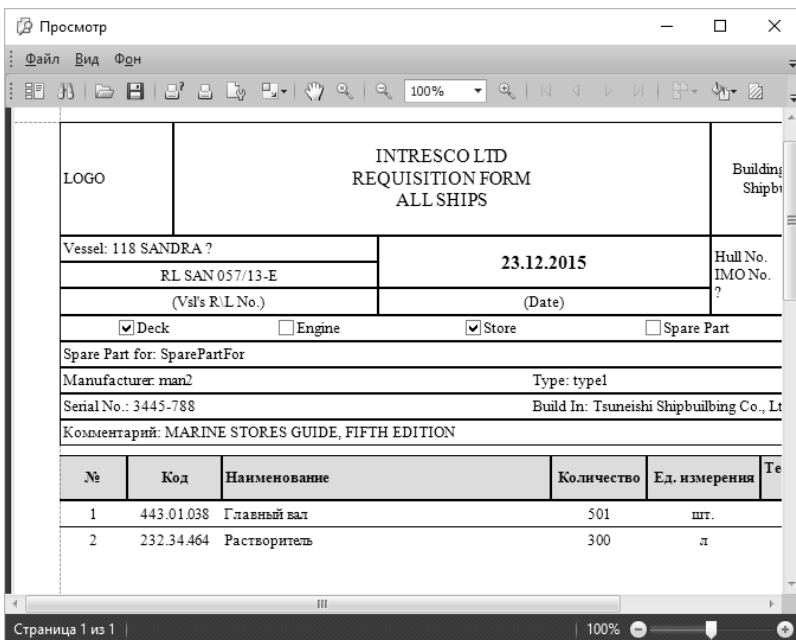


Рис 4. Уровень пользовательского интерфейса: отчёты

Таким образом, предложена архитектура системы технического менеджмента судов, которая может легко адаптироваться под любые бизнес-требования и при этом является интуитивно понятной для пользователя. В основе архитектуры использованы паттерны проектирования, которые облегчают "командную" поддержку системы в случаях добавления в систему новых функций.

## СПИСОК ЛИТЕРАТУРЫ

1. Горб С.И., Каменева А.В. Упрощённая информационная система технического менеджмента судов //Автоматизация судовых технических средств: науч.-техн. сб. – 2008. – Вып. 14. – Одесса: ОНМА. – С. 35 – 42.

2. Горб С.И., Каменева А.В. Новая версия упрощённой информационной системы технического менеджмента // Автоматизация судовых технических средств: науч.-техн. сборник. – 2010. – Вып. 16. – Одесса: ОНМА. – С. 19 – 25.

3. Горб С.И., Каменева А.В. Модель базы данных для модуля идентификации пользователей в системе технического менеджмента судов // Автоматизация судовых технических средств. – 2015. – Вып. 21. – С. 54 – 59.

4. Эрик Фримен, Элизабет Фримен. Паттерны проектирования. – Beijing, Cambridge, Sebastopol, Tokyo: O'Reilly, 2015. – 656 с.

5. Michael C. Feathers. Working Effectively with Legacy Code. – Prentice Hall Professional Technical Reference Upper Saddle River. – Pearson Education, Inc., 2005. – 434 p.