

## CRUD-матрица

CRUD matrix - таблица, которая позволяет согласовать действия системы с элементами данных, чтобы показать, где каждый элемент создается (Created), читается (Read), обновляется (Updated) и удаляется (Deleted).

4 базовые функции при работе с хранилищами данных:

C – CREATE - создание

R – READ - чтение

U – UPDATE - обновление

D – DELETE – удаление

Прецеденты	Классы						
	Книга	Читатель	Читательский билет	Книжный формуляр	Библиотекарь	Книговыдача	Бронь
Регистрация читателя		C	C				
Поиск книги	R						
Выдача книги	U	U		U	U	C	D
Выдача читательского билета		R	U		U		
Возврат книги	U	R	U	U	U	D	
Утрата книги	D	D	D	U	U	D	
Бронирование книги	R	R	U				C
Пере-регистрация читателя		R/U	U		U		
Поиск читателя		R					
	5	9	6	3	5	3	2

Разработка CRUD-матрицы помогает команде сконцентрироваться на существенных прецедентах, описывающих бизнес-процесс. CRUD-матрица формируется в виде таблицы, в которой в верхней части перечисляются все классы из диаграммы классов, а в левой части отражается список прецедентов. Задача аналитика заполнить пересечения между прецедентами и классами следующими комбинациями доступа к экземплярам классов: создание (Create), чтение (Read), обновление (Update) или удаление (Delete).

Таким образом описываются все прецеденты, которые выполняют создание, чтение, обновление или удаление одного или нескольких экземпляров класса.

## **Анализ CRUD-матрицы производится в семь шагов:**

1. Проверка полноты построения модели
2. Определение зависимостей
3. Определение пакета типовых работ для разработки
4. Оценка времени, необходимого для разработки
5. Проверка модели на непротиворечивость
6. Определение последующей работы и дополнений функционала
7. Определение приоритетов для разработки и поставки

### **Шаг 1. Проверка полноты построения модели**

На основе CRUD-матрицы проверяется полнота построения модели на предмет целесообразности использования классов или прецедентов в разрабатываемой системе.

К примеру, в рамках перечисленных прецедентов не описано создание или удаление класса “Книжный формуляр”. Данный факт может конечно говорить о том, что “Книжный формуляр” создается и удаляется вне разрабатываемой системы или же о том, что аналитик не описал прецедент “регистрация книги” или “списание книги”, а для этого нужно ввести еще один класс “инвентарная книга”.

### **Шаг 2. Определение зависимостей**

При планировании процесса разработки системы CRUD-матрица помогает определить перечень классов, которые разрабатываются в первую очередь, чтобы покрыть максимальное количество прецедентов. Например, прецедент “регистрация читателя” будет реализован первым, потому что класс читатель согласно CRUD-матрицы используется в 8 прецедентах, в отличие от класса “бронь”. Таким образом определяются нужные и актуальные данные.

### **Шаг 3. Определение пакета типовых работ для разработки**

CRUD-матрица помогает определить типовую реализацию и выявить дублирование функционала в системе. К примеру, класс “книжный формуляр” реализуется по типу класса “библиотекарь” и соответственно времени на реализацию класса “книжный формуляр” потребуется меньше. Таким образом, сокращается время разработки системы.

### **Шаг 4. Оценка времени, необходимого для разработки**

CRUD-матрица предоставляет команде простой механизм для оценки времени, необходимого для разработки и тестирования определенной части функциональности.

В первом приближении производится оценка каждой комбинации доступа, а затем производится оценка каждого прецедента. К примеру, создание нового экземпляра класса требуется в 4 прецедентах, чтение информации об экземпляре класса в 7 случаях, в 16 случаях требуется обновлять информацию экземпляра класса, а в 6 случаях необходимо удалять экземпляр класса. Используя плановое время на разработку каждой комбинации доступа формируется таблица сложности реализации каждого класса, а также системы в целом:

	С (1 д.)	Р (1 д.)	U (2 д.)	D (2 д.)	Итого времени:
Книга	0	2	2	1	8 дней
Читатель	1	5	2	1	12 дней
Читательский билет	1	0	4	1	11 дней
Книжный формуляр	0	0	3	0	6 дней
Библиотекарь	0	0	5	0	10 дней
Книговыдача	1	0	0	2	5 дней
Бронь	1	0	0	0	1 день
Итого времени:	4 дня	7 дней	32 дня	10 дней	53 дня

#### Шаг 5. Проверка модели на непротиворечивость

В процессе итеративного развития CRUD-матрицы обеспечивается согласованность между прецедентами и теми моделями, которые являются основой разработки и поставки решений. Например, в рамках модели могут быть классы, которые не используются ни в одном прецеденте, что может являться снижением производительности системы, а следовательно это может напрямую повлиять на удовлетворённость клиента. Таким образом, решение, при разработке которого использовалась технология CRUD-матрицы, будет максимально непротиворечивым, что является хорошей основой для разработки.

#### Шаг 6. Определение последующей работы и дополнений функционала

В CRUD-матрице описываются все ответные действия системы по каждому прецеденту, что позволяет разработчику увидеть работу всей системы в целом. На базе CRUD-матрицы аналитик, в случае необходимости, может проводить уточнение модели и строить диаграммы последовательности, диаграммы коммуникации, диаграммы обзора взаимодействия или диаграммы синхронизации. Также CRUD-матрица может использоваться дизайнером при проектировании интерфейсов или архитектором решений для разработки продуктовой спецификации.

### **Шаг 7. Определение приоритетов для разработки и поставки**

CRUD-матрица позволяет клиентам адекватно оценить приоритеты в реализации какой-либо функциональности, и не позволит переместить её реализацию на следующую итерацию. Например, прецеденты “выдача читательского билета” и “перерегистрация читателя” не могут быть поставлены полностью до тех пор пока не будут разработаны классы: читатель, читательский билет и библиотекарь.

### **Задание: Составить CRUD-матрицу для своего проекта**

На основе всех описанных прецедентов для всех объектов составить CRUD-матрицу для проекта