

Шаблоны проектирования

[В начало](#) / [Мои курсы](#) / [ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ](#) / [Дневная форма получения образования](#)
/ [Первая ступень](#) / [Специальность: Прикладная информатика \(программное...](#) / [Учебные дисциплины](#) / [3 КУРС](#) / [ПИ\(ПОКС\) 6. ШП](#)
/ [Модуль 2. «Общие шаблоны распределения ответственн...](#) / [Лабораторная работа 2. Шаблоны распределения ответ...](#)

Лабораторная работа 2. Шаблоны распределения ответственности

Лабораторная работа №2

Шаблоны распределения ответственности

(6 часов)

Задание:

Разработайте приложение-игру пасьянс, позволяющее раскладывать один из карточных пасьянсов (см. свой вариант). Приложение должно выполнять первоначальный расклад, давать пользователю возможность совершать ход (проверяя, является ли он корректным), проверять, есть ли ещё возможные ходы для пользователя, а также выдавать сообщение о том, что пасьянс сошёлся или не сошёлся. При проектировании структуры приложения используйте следующие шаблоны распределения ответственности:

- информационный эксперт;
- создатель;
- контроллер;
- слабая связанность;
- сильное зацепление.

Процесс проектирования проиллюстрируйте UML-диаграммами классов и состояний.

Варианты:

1. Баян.
2. Дважды два.
3. Карлтон.
4. Косынка.
5. Монте-Карло.
6. Пары.
7. Пирамида.
8. Перекресток.
9. Солитер.
10. Тузы.

Указания:

В данной лабораторной работе предполагается изучение основ шаблонов проектирования GRASP (General Responsibility Assignment Software Patterns). Рассматривается лишь пять основных шаблонов.

Шаблон «**информационный эксперт**» принято первым применять при проектировании сложных систем. Основная цель этого шаблона выделить классы, которые будут хранить информацию, и классы, которые будут ее обрабатывать. Под обработкой понимается не только работа алгоритмов, реализующих основную логику приложения, но и взаимодействие (добавление, чтение, изменение, удаление данных) с системой хранения данных, а также визуализация данных, проверка их корректности и т.д. Таким образом, согласно данному шаблону необходимо выделить классы, являющиеся информационными экспертами. Поля таких классов хранят нужную информацию, а методы лишь предоставляют к ней доступ (зачастую это просто методы чтения и записи значений данных полей). После того, как информационные эксперты выделены, эти классы не реализуют никакой логики обработки хранимой информации (даже проверку на корректность данных). Для реализации логики используются другие классы, которые будут обрабатывать данные, хранимые информационными экспертами. По большому счету, все остальные шаблоны регламентируют распределение ответственности между этими оставшимися классами.

Шаблон «**сильное зацепление**» используется при разделении обязанностей между всеми классами и регламентирует количество задач, возлагаемых на каждый класс. Данный шаблон практически полностью повторяет принцип единственной ответственности. Единственное различие принципа единственной ответственности и шаблона проектирования «сильное зацепление» в том, что принцип единственной

ответственности формулируется исходя из цели проектирования (обеспечение гибкой расширяемости приложения), а шаблон проектирования «сильное зацепление» способ достижения этой цели – максимальная конкретизация поставленной перед классом задачи. Один из признаков нарушения этого шаблона – название класса, не вполне адекватно отражающее то, что данный класс делает.

Шаблон «**слабая связанность**» также используется при разделении обязанностей между всеми классами, но регламентирует он количество связей класса с другими классами. Согласно данному шаблону проектирования при проектировании классов нужно стараться минимизировать его связи с другими классами. Практически, этот шаблон декларирует еще один способ соблюдения принципа единственной ответственности. Также этот шаблон связан с шаблоном «сильное зацепление». Действительно, если класс имеет четко обозначенный круг обязанностей (то есть «сильно зацеплен» за свои обязанности), то для решения этих задач ему достаточно минимального количества связей с другими классами. Явными признаками связи между класса являются:

- наследование;
- реализация интерфейса;
- наличие в классе поля-ссылки на другой класс;
- наличие в классе метода, принимающего в качестве параметра ссылку на другой класс;
- наличие в классе метода, возвращающего в качестве результата ссылку на другой класс;
- наличие в методе класса локальной переменной-ссылки на другой класс.

Для ослабления некоторых зависимостей может использоваться другой шаблон – «создатель». Следует, однако, заметить, что зависимости между информационными экспертами и другими классами, как правило, не учитываются. Это происходит от того, что информационные эксперты используются для транспорта информации от одних классов, реализующих обработку данных, другим. Точнее было бы сказать, что не учитывается степень связанности информационного эксперта с другими классами, но для этих классов связи с информационными экспертами могут учитываться. Хотя в такой ситуации проще использовать шаблон «сильное зацепление». А вот степень связанности нескольких информационных экспертов – важный показатель, так как от этого зависит простота расширения предметной области. Но зачастую основным фактором при моделировании предметной области все же является не шаблоны «сильное зацепление» или «слабая связанность», а адекватность создаваемой модели.

Шаблон «создатель» и «контроллер» используется при распределении определенного рода обязанностей.

Шаблон «**создатель**», в частности, – это класс, который берет на себя ответственность за создание (инстанцирование) экземпляра некоторого класса (как правило не информационного эксперта). Чаще всего методы класса-создателя возвращают ссылку на созданный объект, инкапсулируя процесс создания и инициализации этого объекта. В общем случае классом-создателем можно назвать любой класс, который неким образом создает любой объект. Так, например, класс, который читает некоторую информацию из файла, а затем создает на основе этой информации объект информационного эксперта (или список таких объектов) и возвращает ссылку на этот объект, уже можно назвать создателем.

Шаблон «**контроллер**» – это класс, который берет на себя ответственность взаимодействия с внешними по отношению к системе факторами. Чаще всего это взаимодействие с пользователем, но также это может быть обработка некоторых событий, возникающих в определенное время, или обработка сигналов, приходящих с различных датчиков. Мы будем рассматривать только контроллеры, обрабатывающие пользовательские запросы. Такие контроллеры условно можно разделить на две группы:

- единая точка входа (Front Controller);
- контроллер прецедента (Page Controller).

К первой группе относятся контроллеры, которые обрабатывают абсолютно все запросы пользователя, извлекают из такого запроса все необходимые данные, принимают решения, кому перенаправить этот запрос для обработки, после чего либо обработка запроса заканчивается (то есть часть системы, которой передали запрос на обработку, выполнила все необходимые действия и сама сообщила пользователю о результатах), либо контроллер может принимать результат обработки запроса и сообщать его пользователю. Такие контроллеры удобно использовать при создании сетевых серверов, получающих запросы от клиента по некоторому прикладному протоколу (HTTP, FTP и т.д.); при создании настольных приложений с использованием непосредственно Windows API (при наличии единой оконной функции для приложения); при разработке консольных приложений, работающих в командном режиме (когда пользователь сначала полностью набирает текст команды, которая содержит всю необходимую информацию, а затем ее обрабатывает, как, например, консольный клиент для MS SQL Server).

Ко второй группе относятся контроллеры, которые обрабатывают специализированные запросы (например, запрос на создание новой записи в базе данных обрабатывает один контроллер, а запрос на удаление записи – другой). Такие контроллеры удобно использовать при разработке web-приложений, работающих под управлением стандартного web-сервера (Apache, Small HTTP Server, и т.д.). В таком случае общую обработку запроса осуществляет сам web-сервер, а затем вызывается определенная часть нашего приложения (например, определенный PHP-скрипт). Также такие контроллеры удобно использовать в настольных приложениях с использованием высокоуровневых библиотек (VCL, MFC или QT для C++; AWT, Swing или SWT для Java и т.д.). В таком случае один обработчик некоторого события (нажатия на кнопку, выбора элемента из списка и т.д.) является отдельным контроллером. Еще подобные контроллеры используются при разработке консольных приложений, в которых пользователь сначала выбирает некоторое общее действие в меню, а затем в диалоговом режиме приложение уточняет нужные данные. Тогда каждому действию в меню ставится в соответствие свой контроллер, который затем начинает свой диалог с пользователем.

Примечания:

1. Правила пасьянса «Баян»

Пасьянс раскладывается колодой из 52 карт. Карты раскладываются в 4 горизонтальных ряда по 13 карт в каждом. Каждая карта – это стопка из одной карты. Далее стопки начинают объединяться по следующему правилу: на любую стопку можно положить стопку правее данной или стопку, располагающуюся справа от данной через одну стопку, при этом верхние карты перекладываемых стопок должны быть одинаковой масти или одинакового достоинства. Для крайней правой стопки в ряду правой стопкой считается крайняя левая стопка в следующем ряду. После перекладывания стопки на освободившееся место сдвигаются все стопки правее. Цель пасьянса сложить все карты в одну стопку.

Пример начального расклада см. на рис. 1.1

6♠ 5♦ K♥ K♣ 6♥ 7♦ 5♥ 9♦ 10♣ 2♥ 9♣ 3♥ 2♣
 4♣ K♦ 8♣ K♣ B♣ 4♥ B♣ T♣ Д♦ 8♣ Д♣ 6♣ Д♣
 8♥ Д♥ B♦ 5♣ 10♦ 7♣ 3♣ 6♦ 10♥ 8♦ 2♦ 5♣ T♦
 3♣ 10♣ 2♣ 4♦ 7♥ 3♦ T♥ T♣ 9♥ 9♣ B♥ 4♣ 7♣

Рисунок 1.1

Для этого начального расклада можно рассмотреть третью карту в первом ряду K♥, на эту стопку можно положить либо четвёртую карту K♣, так как совпадают их достоинства, либо пятую 6♥, так как совпадают их масти. Если же рассмотреть вторую карту во втором ряду K♦, то на эту стопку можно положить только четвёртую карту второго ряда K♣. Или на первую карту третьего ряда 8♥ можно положить только вторую карту этого же ряда Д♥. Для демонстрации правил сдвига всех карт после перекладывания, рассмотрим перекладывание на последнюю карту второго ряда Д♣ второй карты третьего ряда Д♥. После чего расклад примет вид как на рис. 1.2

6♠ 5♦ K♥ K♣ 6♥ 7♦ 5♥ 9♦ 10♣ 2♥ 9♣ 3♥ 2♣
 4♣ K♦ 8♣ K♣ B♣ 4♥ B♣ T♣ Д♦ 8♣ Д♣ 6♣ Д♥
 8♥ B♦ 5♣ 10♦ 7♣ 3♣ 6♦ 10♥ 8♦ 2♦ 5♣ T♦ 3♣
 10♣ 2♣ 4♦ 7♥ 3♦ T♥ T♣ 9♥ 9♣ B♥ 4♣ 7♣

Рисунок 1.2

После чего переложим стопку Д♥ на Д♣, а следующим ходом Д♥ на Д♦. Итоговый расклад будет иметь вид как на рис. 1.3

6♠ 5♦ K♥ K♣ 6♥ 7♦ 5♥ 9♦ 10♣ 2♥ 9♣ 3♥ 2♣
 4♣ K♦ 8♣ K♣ B♣ 4♥ B♣ T♣ Д♥ 8♣ 6♣ 8♥ B♦
 5♣ 10♦ 7♣ 3♣ 6♦ 10♥ 8♦ 2♦ 5♣ T♦ 3♣ 10♣ 2♣
 4♦ 7♥ 3♦ T♥ T♣ 9♥ 9♣ B♥ 4♣ 7♣

Рисунок 1.3

Цель пасьянса – собрать всю колоду в одну стопку.

2. Правила пасьянса «Дважды два»

Пасьянс раскладывается колодой из 52 карт. В начале из колоды выкладываются верхние четыре карты, затем на любые две из этих четырёх карт, которые имеют одинаковую масть, выкладываются следующие две карты из колоды.

Например, пусть первые четыре карты колоды: B♣ Д♦ 4♣ Д♣. На B♣ и 4♣ кладутся ещё две карты из колоды, например K♣ и 8♥. Получаем верхние четыре карты: K♣ Д♦ 8♥ Д♣. В этом случае все четыре карты имеют разную масть, больше из колоды ни одной карты забрать не получается, пасьянс не сошёлся.

Рассмотри теперь такой начальный расклад: 4♣ 6♦ 4♦ 2♣. Следующие две карты можно выложить только на 6♦ и 4♦. Допустим, это будут K♣ и 5♣. Получим расклад: 4♣ K♣ 5♣ 2♣. В таком случае следующие две карты можно выложить на 4♣ и 5♣ или на K♣ и 2♣. Допустим, мы выбираем 4♣ и 5♣, а сами следующие карты в колоде 3♦ и 6♣. Получим расклад: 3♦ K♣ 6♣ 2♣. В таком случае следующие две карты могут быть выложены на K♣ и 6♣, на K♣ и 2♣ или на 6♣ и 2♣.

Процесс выкладывания карт продолжается, пока все четыре карты не окажутся разной масти (пасьянс не сошёлся), или будет исчерпана вся колода (пасьянс сошёлся).

3. Правила пасьянса «Карлтон»

Пасьянс раскладывается колодой из 52 карт. Карты из колоды раскладываются на 4 стопки, в первой стопке 4 карты, во второй стопке 3 карты, в третьей стопке 2 карты, в четвертой стопке 1 карта. Карты в стопках раскладываются так, чтобы все карты в стопке были видны. Сверху над стопками располагаются 4 основания для собираемых карт. Рядом кладется колода с оставшимися картами. Перекладывать карты со стопки на стопку можно в нисходящей последовательности с чередованием цвета масти (например, на чёрного валета можно перекладывать красную десятку, с одной стопки на другую можно перекладывать сразу несколько карт, если они уже упорядочены в требуемом порядке). Также на пустую стопку можно переложить короля, или несколько карт с нижним королём. В основания сверху над стопками сначала кладутся тузы, а далее карты в восходящей последовательности: 2, 3, 4, 5, 6, 7, 8,

9, 10, валет, дама, король – все карты на основании должны быть одной масти. Когда карт для перекладывания нет, то на каждую стопку (даже если в ней уже нет ни одной карты) берётся по одной карте из колоды. Пасьянс сошелся, если все карты собраны на 4 основаниях.

4. Правила пасьянса «Косынка»

Пасьянс раскладывается колодой из 52 карт. Карты из колоды раскладываются на 7 стопок, в первой стопке 1 карта, во второй стопке 2 карты и так далее. Верхняя карта стопки остаётся открытой, остальные перевёрнуты. Сверху над стопками располагаются 4 основания для собираемых карт. Рядом кладется колода с оставшимися картами. Перекладывать карты со стопки на стопку можно в нисходящей последовательности с чередованием цвета масти (например, на чёрного валета можно перекладывать красную десятку, с одной стопки на другую можно перекладывать сразу несколько открытых карт, если они уже упорядочены в требуемом порядке). Также на пустую стопку можно переложить короля, или несколько карт с нижним королём. В основания сверху над стопками сначала кладутся тузы, а далее карты в восходящей последовательности: 2, 3, 4, 5, 6, 7, 8, 9, 10, валет, дама, король – все карты на основании должны быть одной масти. Когда карт для перекладывания нет, то из колоды снимается стопка из трёх карт (по одной) в отдельную резервную стопку. Из резервной стопки можно снимать верхнюю карту и класть на любую основную стопку с соблюдением нисходящей последовательности и чередования цвета. Также из резервной стопки можно перекладывать карты сразу на основание с соблюдением масти. Когда вся колода переложена в резервную стопку, все карты из резервной стопки возвращаются в колоду (без перемешивания, просто стопка переворачивается вверх рубашкой). Пасьянс сошелся, если все карты собраны на 4 основаниях.

5. Правила пасьянса «Монте-Карло»

Пасьянс раскладывается колодой из 52 карт. Карты раскладываются в 4 горизонтальных ряда по 5 карт в каждом. Далее две карты одного достоинства, лежащие рядом (по горизонтали, вертикали или диагонали) убираются (по парам). После удаления карт на освободившееся место сдвигаются все карты правее. На освободившиеся крайние правые места в ряду перекладываются карты с левого края нижнего ряда. На освободившиеся крайние правые места нижнего ряда докладываются две карты из колоды. Цель пасьянса сложить все карты в одну стопку.

Пример начального расклада см. на рис. 5.1

6♣ 5♦ К♥ К♠ 6♥
 4♣ К♦ 8♣ К♠ В♣
 8♥ Д♥ В♦ 5♠ 10♦
 3♠ 10♠ 2♠ 4♦ 7♥

Рисунок 5.1

В этом начальном раскладе можно убрать К♥ и К♠ (рядом по горизонтали), или К♣ и К♠ (рядом по вертикали), или К♥ и К♦ (рядом по диагонали). Если убрать двух красных королей и произвести сдвиг карт на освободившиеся места, получим расклад как на рис. 5.2

6♣ 5♦ К♠ 6♥ 4♣
 8♣ К♠ В♣ 8♥ Д♥
 В♦ 5♠ 10♦ 3♠ 10♠
 2♠ 4♦ 7♥

Рисунок 5.2

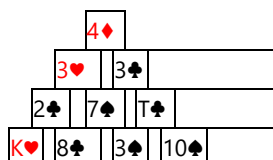
На два освободившиеся места из колоды дополнительно извлекаются ещё две карты. Если в колоде закончились карты, то пустые позиции в конце так и остаются. Цель пасьянса – убрать все карты.

6. Правила пасьянса «Пары»

Пасьянс раскладывается колодой из 36 карт. Все карты раскладываются в 9 стопок по 4 карты в каждой. Верхние карты стопок открываются. Далее за каждый ход убирается одна пара карт одного достоинства. После чего в стопках, с которых удалялись карты, открываются оставшиеся верхние карты. Цель пасьянса – убрать все карты.

7. Правила пасьянса «Пирамида»

Пасьянс раскладывается колодой из 52 карт, при этом 28 карт раскладываются в виде пирамиды в 7 рядов: в первом ряду 1 карта, во втором – 2, в третьем – 3, и так далее. Карты раскладываются таким образом, что карты нижних рядов перекрывают две карты верхнего ряда (кроме крайних, которые перекрывают только одну верхнюю карту). Пример изначального расклада карт представлен на рис. 7.1



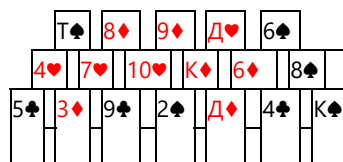


Рисунок 7.1

Далее из колоды вынимается в отдельную стопку одна карта, на которую из пирамиды кладутся свободные (не закрытые другими) карты на одно очко больше или меньше. Например, если в отдельной стопке находится туз, то на него можно положить 2♣, затем 3♦, 4♣, 5♣. Снятые 3♦ и 5♣ освобождают 4♥, которую тоже можно положить на 5♣. Когда все возможные карты из пирамиды переложены в отдельную стопку, из колоды достаётся следующая карта в эту же стопку. Этот процесс продолжается, пока не будет исчерпана пирамида (пасьянс сошёлся) или не закончится колода (пасьянс не сошёлся).

8. Правила пасьянса «Перекресток»

Пасьянс раскладывается колодой из 52 карт. Пять карт выкладываются в виде креста. Колода кладётся рядом, в колоде верхняя карта переворачивается. Таким образом, игровыми являются 6 карт, из которых попарно убираются карты одинакового достоинства. Если одинаковых карт нет, то верхняя карта из основной колоды перекладывается в центральную стопку. Если в боковых стопках (выше, ниже, левее и правее центральной) нет карт, то в эти стопки перекладываются карты из центральной стопки. Если в центральной стопке нет карт, то в неё перекладывается карта из основной колоды. Цель пасьянса – убрать все карты.

9. Правила пасьянса «Солитер»

Пасьянс раскладывается колодой из 52 карт. Колода раскладывается в 8 стопок. В первых 4 стопках по 7 карт, в остальных 4 стопках по 6 карт. При этом стопки раскладываются «лесенкой», чтобы все карты в стопке были видны. Отдельно резервируется место под четыре стопки, в которых необходимо собрать всю колоду (каждую масть в отдельную стопку, в каждой стопке карты идут по возрастанию: туз, двойка, тройка, и так далее до короля). Для перекладывания карт в пасьянсе резервируется ещё четыре запасных стопки, в которых может быть не более одной карты в каждой. В эти запасные стопки можно положить любую верхнюю карту любой из 8 основных стопок. В основных стопках карты перекладываются по следующему правилу: на старшую карту кладутся младшие (например, на десятку можно положить девятку, но нельзя восьмёрку или валета) с чередованием черной и красной масти.

10. Правила пасьянса «Тузы»

Пасьянс раскладывается колодой из 36 карт. Карты раскладываются в 12 стопок (3 ряда по 4 стопки в ряду), в каждой стопке по 3 карты (так, чтобы все карты в стопке были видны). Далее необходимо переложить все карты в четыре стопки по одной масти в каждой стопке начиная с шестёрок, далее семёрки и так далее до тузов. Забирать можно только верхние карты стопки, при этом можно переложить карту с одной стопки в другую на карту того же достоинства, но в каждой стопке не должно быть более трёх карт. Перекладывать карты на пустые стопки нельзя. Если при изначальном раскладе ни в одной стопке сверху нет шестёрок, то одну любую шестёрку можно достать из середины стопки.

Состояние ответа

Состояние ответа на задание	Ни одной попытки
Состояние оценивания	Не оценено
Последнее изменение	-

Комментарии к ответу

► [Комментарии \(0\).](#)

Добавить ответ на задание

Вы пока не предоставили ответ на задание

Перейти на...



Вы зашли под именем [Беляев Максим](#) (Выход)

[ПИ\(ПОКС\) 6. ШП](#)

[Русский \(ru\)](#)

[Русский \(ru\)](#)

[English \(en\)](#)

[Скачать мобильное приложение](#)