

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 3. «Шаблоны проектирования GoF»](#) / [Лабораторная работа 4. Структурные шаблоны проекти...](#)

Лабораторная работа 4. Структурные шаблоны проектирования

Лабораторная работа №4 Структурные шаблоны проектирования (4 часа)

Внимание! В данной лабораторной работе на выбор студента можно выполнять либо 1-ое и 2-ое задание, либо только 3-е.

Выполнение 4-го задания необязательно, но позволяет повысить практическую составляющую оценки на экзамене.

Задание 1:

Дан [исходный код приложения](#), которое в окне строит некоторое графическое изображение. Необходимо выполнить оптимизацию структуры классов приложения в соответствии с шаблоном проектирования **«Decorator»**, с помощью которого необходимо графическим примитивам задавать цвет фона и линий, ширину линий. После чего необходимо, с использованием переработанных классов, разработать ещё одно приложения, которое будет строить в окне рисунок в соответствии с вариантом. При этом в рисунке необходимо использовать все графические примитивы, цвета и способы заливки.

Варианты задания:

1. Домик в деревне.
2. Новогодняя елка.
3. Космический корабль.
4. Автомобиль.
5. Яхта.
6. Подводная лодка.
7. Паровоз.
8. Снеговик.
9. Клоун.
10. Ваза с цветами.

Задание 2:

1. Разработать приложение, генерирующее несколько последовательностей чисел (по заранее заданному алгоритму, например, последовательность чисел Фибоначчи, последовательность простых чисел, последовательность факториалов целых неотрицательных чисел). Генерирование типа последовательности и количество генерируемых элементов должно определяться пользователем. Для каждой последовательности после генерации указать время работы соответствующего алгоритма. Определение этого времени реализовать, используя шаблон проектирования **«Decorator»**.
2. Разработать приложение, создающее список плоских геометрических фигур (круг, квадрат, треугольник и т.д., но не менее 5 фигур). Создание списка должно быть реализовано несколькими способами (ввод с клавиатуры, чтение из файла, генерирование случайным образом). Далее необходимо вывести данные о тех фигурах, площадь которых меньше средней площади всех фигур. При этом вычисление площади фигур может производиться обычным способом, а может округляться по избытку, по недостатку или до ближайшего значения до указанного пользователем количества знаков после запятой (вид округления тоже указывается пользователем). Округление реализовать самостоятельно (через соответствующие методы класса Math), используя шаблон проектирования **«Decorator»**.
3. Разработать приложение, читающее список неких товаров (название, вес одной единицы, стоимость одной единицы, количество единиц товара) из некоторого файла (формат файла можно выбрать произвольно). Далее приложение должно выводить этот список на экран, отсортировав его по суммарному весу всех единиц каждого товара. Затем приложение должно выводить этот же список на экран ещё раз, отсортировав его по суммарной стоимости всех единиц каждого товара. Для сортировки списка товаров использовать возможности класса Collections. После каждого вывода отсортированного списка необходимо вывести количество операций сравнения, использовавшееся при сортировке списка. Подсчёт количества операций сравнения реализовать, используя шаблон проектирования **«Decorator»**.

4. Разработать приложение с графическим пользовательским интерфейсом, представляющее собой простейший калькулятор целых чисел, содержащий одно текстовое поле (заблокированное для непосредственного редактирования) и 16 кнопок: 10 цифр, 4 операции (сложение, вычитание, умножение, деление), кнопку «равно» и кнопку «+/-». Приложение должно работать в демонстрационном режиме, то есть после запуска оно должно считать нажатия пользователя на клавиши, и после 10 нажатий, остальные действия игнорировать. Реализацию ограничения демонстрационной версии организовать через шаблон проектирования **«Decorator»**.
5. Разработать приложение, формирующее прайс-лист товаров некоторого магазина. Список товаров (категория, название, закупочная стоимость) считывается из некоторого файла (формат можно выбрать произвольным образом). Также для каждой категории товаров в отдельном файле задаётся процент наценки, если для некоторой категории процент наценки не указан, то принимается наценка по умолчанию (заданная в том же файле). Ещё в одном файле определяются скидки на некоторые конкретные товары, при чём на некоторые товары скидка определяется в процентах, для некоторых - в рублях. Приложение должно вывести все товары с окончательной ценой. Расчет наценок и скидок реализовать через шаблон проектирования **«Decorator»**.
6. Разработать приложение, формирующее отчёты по расписанию занятий. Расписание храниться в отдельном файле, для каждого пункта расписания определены: название предмета, тип занятия (лекция, практика, лабораторная), дата и время начала, длительность в академических часах, номер аудитории. Приложение должно уметь на основе этих данных генерировать три вида отчетов: расписание по конкретному предмету с сортировкой по типам занятий, расписание на конкретный день с сортировкой по времени, расписание для конкретной аудитории с указанием с сортировкой по дню проведения занятия. При этом отчёт должен выводиться в файл на диске, но при ошибке записи отчёта (при возникновении `IOException` или другой ошибки), в отдельный файл ошибок необходимо вывести сообщение об ошибке, а генерацию отчёта прервать (с удалением ошибочного файла). Обработку ошибок в приложении реализовать, используя шаблон проектирования **«Decorator»**.
7. Разработать приложение, выводящее список товаров мебельного магазина, отсортированного по цене. У каждого товара должны быть заданы: категория, название, артикул (уникальный код) и цена. Для мебели в категории «шкафы» необходимо дополнительно указывать тип дверей (обычные, купе, стеклянные). Для мебели категории «стол» необходимо указывать количество выдвижных ящиков. Для мебели категории «стул» необходимо указывать наличие спинки. Кроме продажи отдельных предметов мебели в магазине могут продаваться наборы со скидкой 5% от общей стоимости всех предметов мебели этого набора. Подсчёт стоимости набора мебели организовать через шаблон проектирования **«Composite»**.
8. Разработать приложение, выводящее список товаров компьютерного магазина, отсортированного по цене. У каждого товара должны быть заданы: категория, название, артикул (уникальный код) и цена. Для комплектующих в категории «процессор» необходимо дополнительно указывать тактовую частоту, количество ядер и тип процессорного разъёма. Для комплектующих в категории «оперативная память» необходимо указывать объём в мегабайтах. Для комплектующих категории «видеокарты» необходимо указывать наличие активного охлаждения и объём памяти. Для комплектующих категории «жёсткие диски» необходимо указывать объём в гигабайтах и скорость вращения шпинделя. Для комплектующих категории «материнская плата» необходимо указывать наличие встроенной видеокарты, максимальный поддерживаемый объём оперативной памяти, тип процессорного разъёма. Для комплектующих категории «корпус» необходимо указывать мощность блока питания и тип корпуса (FullTower, Tower, MiniTower, Desktop). Для комплектующих категории «монитор» необходимо указывать размер диагонали экрана и соотношение сторон экрана. Для комплектующих категории «устройства ввода» необходимо указывать тип клавиатуры (обычная, мультимедиа) и количество кнопок мыши. Кроме продажи отдельных комплектующих магазин может продавать системные блоки (корпус, материнская плата, процессор, оперативная память, жёсткий диск, опционально: видеокарта) с надбавкой 15% за сборку или компьютеры целиком (системный блок, монитор, устройства ввода) с надбавкой 5% за сборку. Подсчёт стоимости товаров организовать через шаблон проектирования **«Composite»**.
9. Разработать приложение, вычисляющее длительность технологических процессов для различных видов деталей. Есть набор элементарных технологических операций, для каждой из которых задаются свои параметры, влияющие на скорость ее выполнения. Операция «Подготовка заготовки»: диаметр стального цилиндра, диаметр заготовки, длина заготовки - время линейно зависит от объёма считываемого металла (по $1 \text{ мм}^3/\text{с}$). Операция «Нарезка резьбы»: диаметр заготовки, длина резьбы, шаг резьбы - время линейно зависит от общей длины резьбы (длина окружности заготовки умноженная на длину резьбы, поделенную на шаг резьбы даёт общую длину резьбы, время операции равно $1 \text{ см}/\text{с}$). Операция «Термообработка»: температура, время выдерживания, скорость охлаждения - время операции складывается из времени разогревания (линейно зависит от температуры: $10^\circ\text{C}/\text{с}$), времени выдерживания и времени охлаждения. Элементарные технические операции используются для производится нескольких видов деталей: винтов (термообработка заготовки при температуре 1000°C с выдерживанием в течение 10 минут и охлаждением со скоростью $0,25^\circ\text{C}/\text{с}$, подготовка заготовки, нарезка резьбы); анкерных болтов (термообработка заготовки при температуре 1250°C с выдерживанием в течение 30 минут и охлаждением со скоростью $0,5^\circ\text{C}/\text{с}$, подготовка заготовки, нарезка резьбы, термообработка заготовки при температуре 1500°C с выдерживанием в течение 30 минут и охлаждением со скоростью $300^\circ\text{C}/\text{с}$); гусеничные траки (подготовка заготовки, термообработка заготовки при температуре 1000°C с выдерживанием в течение 60 минут и охлаждением со скоростью $300^\circ\text{C}/\text{с}$, термообработка заготовки при температуре 1000°C с выдерживанием в течение 30 минут и охлаждением со скоростью $0,1^\circ\text{C}/\text{с}$). Расчёт времени технологического процесса, состоящего из нескольких технологических операций, реализовать с использованием шаблона проектирования **«Composite»**.
10. Разработать приложение, вычисляющее значение матричного выражения, записанного в Польской инверсной записи. Выражение состоит из операндов (обозначающих квадратные матрицы) и операций, операнды и операции отделяются друг от друга пробелом. В Польской инверсной записи сначала следуют операнды, а затем знак операции, что позволяет избежать использование скобок в выражении. Например, выражение $(a + b) * (c + d) - e / f$ в Польской инверсной записи примет вид: $a \ b \ + \ c \ d \ + \ * \ e \ f \ / \ -$. После ввода пользователем выражения программа должна запрашивать у него значение матриц, обозначенных операндами (при этом один и тот же операнд не должен запрашиваться дважды). При вводе значений матриц сначала запрашивается размер матриц (общий для всех операндов - квадратных матриц), затем тип матрицы (обычная, симметричная, верхнетреугольная, нижнетреугольная, диагональная, диагональная с постоянным элементом на главной диагонали), затем вводятся элементы матрицы. После чего вычисляется и выводится значение введённого выражения, но промежуточные результаты в памяти храниться не

должны, вычисление всех операций (сложение, вычитание, умножение, деление: умножение на обратную матрицу) должно быть реализовано через шаблон проектирования «Composite».

Указания:

Для шаблона проектирования «Decorator» реализовать сначала базовую функциональность приложения, и только после этого, не меняя классы, реализующие эту функциональность, добавить класс (классы), реализующие с помощью декоратора расширенную функциональность. Аналогично для шаблона проектирования «Composite» сначала необходимо реализовать обработку базовых объектов, и только после этого с помощью компоновщика объединять их в составные объекты (в наборы объектов).

Задание 3:

В индивидуальном проекте при получении ссылки на экземпляр класса-сущности через ссылку на имеющийся экземпляр другого класса-сущности, хранящий получаемую ссылку как своё поле, реализовать шаблон проектирования «Lazy Load», для чего использовать шаблон «Proху».

Указание: для классов-сущностей необходимо описать интерфейсы, содержащие необходимые get-теры и set-теры. Имеющиеся классы-сущности должны реализовать эти интерфейсы. Также для классов-сущностей, имеющих ссылки на другие классы-сущности, описать классы-обёртки, которые при обращении к соответствующему get-теру, обращаются к методам других классов, выполняющих запросы к базе данных и подгружающих необходимые классы-сущности по мере необходимости. В такой ситуации неявное обращение к классу-оболочке, а значит и к классу, взаимодействующему с базой данной, будет происходить на уровне jsp-страниц, что ставит вопрос о корректном закрытии подключения к базе данных (или возврате подключения в пул). Решить этот вопрос можно, используя обработку события уничтожения запроса. Обработчик этого события должен получить доступ к подключению, или классу, его содержащему, и закрыть его. При этом необходимый класс может автоматически размещаться в атрибуте запроса при его создании (в обработке события создания запроса).

Задание 4*:

В индивидуальном проекте вынести бизнес-логику приложения в отдельный слой, отделив её от визуализации (сервлетов) и от способа доступа к данным (классов, выполняющих SQL-запросы). Используя шаблон проектирования «Proху» и возможности механизма рефлексии в Java (класс `java.lang.reflect.Proxy` и интерфейс `java.lang.reflect.InvocationHandler`) автоматизировать выполнение транзакции в методах классов бизнес-логики.

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

◀ Лабораторная работа 3. Порождающие шаблоны проектирования

Перейти на...



Лабораторная работа 5. Шаблоны проектирования поведения ►

Вы зашли под именем [Беляев Максим](#) ([Выход](#))
[ПИ\(ПОКС\) 6. ШП](#)

[Русский \(ru\)](#)

[Русский \(ru\)](#)

[English \(en\)](#)

[Скачать мобильное приложение](#)