

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 3. Порождающие шаблоны проекти...](#)

Лабораторная работа 3. Порождающие шаблоны проектирования

Лабораторная работа №3

Порождающие 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.

Задание 1:

Дан [исходный код приложения](#), которое считывает из XML-документа данные о днях рождения некоторых людей и выводит их в виде таблицы на форме, позволяя отсортировать по различным критериям. Необходимо выполнить оптимизацию структуры классов приложения в соответствии с шаблоном проектирования «Factory Method», с помощью которого необходимо создавать объекты классов, реализующих интерфейс `Comparator`. При этом создание таких объектов нужно реализовать через объект класса `JComboBox`, который может быть параметризован любым классом. При отображении строк в выпадающем списке, используется метод `toString()` класса, которым параметризован компонент `JComboBox`. Также можно получить выбранный пользователем объект методом `getSelectedItem()` класса `JComboBox`, выполнив явное приведение типов к классу, которым параметризован `JComboBox`.

После оптимизации структуры классов необходимо добавить в таблицу ещё один строковый столбец и способ сортировки по этому столбцу в соответствии с вариантом.

Варианты:

1. почтовый адрес (строка, содержащая улицу, номер дома, номер квартиры, почтовый индекс) с сортировкой по индексу;
2. номер телефона (код телефона в круглых скобках и номер телефона после скобок) с сортировкой по части номера телефона после скобок;
3. адрес электронной почты с сортировкой по доменному имени почтового сервера;
4. номер паспорта (серия и собственно номер, разделённые пробелом) с сортировкой по собственно номеру;
5. номер кредитной карты (четыре группы цифр, разделённые пробелом) с сортировкой по последней группе цифр;
6. почтовый адрес (строка, содержащая улицу, номер дома, номер квартиры, почтовый индекс) с сортировкой количеству символов в названии улицы;
7. номер телефона (код телефона в круглых скобках и номер телефона после скобок) с сортировкой по количеству цифр в коде телефона;
8. адрес электронной почты с сортировкой по длине имени пользователя;
9. доменное имя с сортировкой по уровню домена;
10. номер кредитной карты (четыре группы цифр, разделённые пробелом) с сортировкой количеству цифр в первой группе;

Задание 2:

Даны списки некоторых объектов, хранящихся в некотором внешнем хранилище (см. задание своего варианта). Необходимо написать приложение, считывающее из аргументов командной строки тип и имена файлов, в которых хранятся объекты. Первый аргумент определяет тип используемых файлов (csv или xml формат). Далее приложение выполняет требуемую обработку данных и выводит их на экран.

Указания:

Создание списков объектов на основе файлов различных типов реализовать через шаблон проектирования «Builder». Анализ расширения файла и выбор нужного «строителя» реализовать через шаблон проектирования «Factory Method». Сам класс, реализующий «Factory Method», создавать через шаблон проектирования «Singleton».

Варианты:

1. Даны: список грузовых автомобилей (гос. номер, марка, водитель) и список рейсов (дата, масса груза, длина маршрута, гос. номер автомобиля). Определить суммарный пробег каждого автомобиля и среднюю массу перевозимого им груза за последний месяц.
2. Даны: список счетов банка (номер счёта, владелец) и список платежей (дата, счёт плательщика, счёт получателя, дата платежа, сумма платежа). Вывести для каждого счёта баланс по дням (в порядке возрастания даты).
3. Даны: список квартир (номер квартиры, подъезда, этаж, имя владельца) и список заявок на ремонт (квартира, дата выполнения работ, описание работы, планируемое время работы в часах). Вывести для каждой квартиры суммарное время выполнения всех работ (в порядке убывания этого времени).
4. Даны: список организаций (название, адрес) и список направлений на распределение (организация, фамилия и инициалы студента, специальность, дата выхода на работу). Вывести для каждой организации список всех направлений на распределение с сортировкой по дате выхода на работу, сам список организаций отсортировать по убыванию количества заявок.
5. Даны: список адвокатов (фамилия, имя, отчество, специализация) и список судебных процессов (адвокат, начало и окончание процесса, описание, результативность: выиграл или проиграл). Вывести для каждого адвоката список его дел (в порядке возрастания даты начала процесса) с подсчётом результативности адвоката (отношение количества часов, затраченных на успешные дела, к общему количеству часов по всем делам). Сам список адвокатов отсортировать по специализации с сортировкой по убыванию результативности адвоката по каждой специализации.
6. Даны: список абонентов телефонного оператора (номер телефона, баланс счёта в рублях, имя владельца) и исходящих звонков (абонент, дата и время начала и окончания звонка, стоимость минуты звонка в рублях). Вывести список абонентов, отсортированных по имени, и для каждого абонента вывести список звонков по возрастанию времени начала звонка. Для каждого абонента вывести актуальный баланс (за вычетом стоимости каждого звонка).
7. Даны: список планет (название, радиус, масса) и список спутников (родительская планета, название, масса, дата открытия). Вывести для каждой планеты список её спутников в порядке их открытия, сам список планет отсортировать по радиусу планеты, для каждой планетарной системы подсчитать общую массу.
8. Даны: список станков (название, производительность: количество деталей в час, прибыль с одной детали в рублях) и список ремонтных работ за месяц (станок, дата ремонта, описание, длительность в часах, стоимость в рублях). Вывести для каждого станка список всех ремонтных работ (с сортированием по дате) и доходность каждого станка (прибыль от всех произведенных за месяц деталей, за вычетом времени, проведённого на ремонте, и стоимости самого ремонта, при этом считать, что станок в штатном режиме работает 10 часов в день, 25 дней в месяц).
9. Даны: список вопросов теста (содержимое, максимальная оценка в баллах) и список вариантов ответов (вопрос, содержание ответа, правильный ли это вариант). Вывести для каждого вопроса список вариантов ответов в случайном порядке, подсчитать вероятность угадывания ответа для каждого вопроса. Сам список вопросов вывести в порядке убывания максимальной оценки.
10. Даны: список экскурсий туристического оператора (описание экскурсии, стоимость экскурсии) и список туристов (экскурсия, на которую записан турист; имя; возраст; является ли студентом). Вывести для каждой экскурсии список её участников. Список участников отсортировать по имени, список экскурсий - по убыванию выручки за экскурсию (стоимость экскурсии суммируется по всем участникам, при этом студенты оплачивают лишь половину стоимости, подростки до 15 лет - треть стоимости, дети до 7 лет участвуют в экскурсии бесплатно).

Задание 3:

Для индивидуального проекта (см. отдельный блок в нижней части курса), отделить логику обработки данных от их представления (отделить классы-сервлеты от класса **Storage** или его аналогов) с помощью принципа инверсии зависимостей. Для создания экземпляров классов, реализующих требуемые интерфейсы, использовать шаблон проектирования «Factory Method» или «Abstract Factory». Для создания экземпляров классов-сущностей использовать шаблон проектирования «Builder».

Вместо указанных выше шаблонов можно реализовать свой собственный IoC-контейнер (IoC — Inversion of Control)

Для создания экземпляров класса, реализующего интерфейс **java.sql.Connection**, использовать шаблоны проектирования «Object Pool» (его модификацию «Connection Pool») и «Singleton».

В случае использования собственного IoC-контейнера необходимо реализовать в нём механизм внедрения зависимостей (dependency injection) через конструкторы или *set-теры*

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

◀ 6. Шаблоны проектирования поведения

Перейти на...



[Лабораторная работа 4. Структурные шаблоны проектирования](#) ▶

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

[Русский \(ru\)](#)
[Русский \(ru\)](#)
[English \(en\)](#)

[Скачать мобильное приложение](#)