

## Лабораторная работа №2.

### Java SE, ООП, шаблоны

#### Работа с базами данных, использование API JDBC

**15. Прокат автомобилей.** Клиент оставляет Заявку на Автомобиль конкретной модели, указывая паспортные данные, срок аренды. Администратор оформляет Заказ, выбрав Автомобиль, соответствующий Заявке из списка доступных, после чего Клиенту выставляется Счет. По факту возврата Администратором фиксируется текущее состояние Автомобиля.

**Часть 1** Создать набор объектов Автомобили, провести сортировку и поиск объекта в наборе.

#### Часть 2

Вывести список заявок на автомобиль заданной модели.

Вывести информацию о количестве доступных автомобилей заданной модели.

Вывести текущее состояние заданного автомобиля.

Оформить заказ, оплатить счет.

#### Часть 1. Java SE, ООП, шаблоны

Создать консольное приложение на языке Java, удовлетворяющее требованиям:

На основе сущностей предметной области создать классы их описывающие. При создании объектов и их коллекций использовать порождающие шаблоны проектирования (Design Patterns).

Реализовать алгоритмы сортировки и поиска объекта (применяя Comparator). При реализации алгоритмов бизнес-логики использовать по возможности структурные и поведенческие шаблоны.

#### Часть 2 JDBC API

1. Спроектировать базу данных для своего варианта задания. Таблицы БД должны быть во 2 нормальной форме. Для работы с БД рекомендуется использовать СУБД MySQL.

2. Организовать соединение с базой данных: создать Java-класс, реализующий работу с connection (JdbcConnector).

3. При организации доступа к БД использовать шаблон Data Access Object. Создать DAO классы для выполнения запросов на извлечение информации из БД (использовать PreparedStatement) и добавление информации. Подготовленные SQL запросы хранить в виде констант в соответствующем DAO классе.

4. Использовать файл ресурсов database.properties, в котором хранить параметры доступа к БД.

Взаимодействие пользователя с приложением организовать в консольном виде.

#### Часть 3 Пул соединений

1. Создать класс (ConnectionPool), реализующий пул соединений, хранящихся в коллекции и предоставляемых повторно. Обеспечить потокобезопасность пула. Заменить класс JdbcConnector на ConnectionPool.

2. Продумать обработку исключений, выбрасываемых в пуле и классах DAO, логгер

	Срок выдачи задания:	Срок сдачи задания:
12 группа	25.09.23	1 часть 5.10.23 2 часть 12.10.23 3 часть 19.10.23 пул соединений 26.10.23 исключения, логгер
13 группа	25.09.23	1 часть 2.10.23

		2 часть 9.10.23 3 часть 16.10.23 пул соединений 23.10.23 исключения, логгер
14 группа	25.09.23	1 часть 2.10.23 2 часть 9.10.23 3 часть 16.10.23 пул соединений 23.10.23 исключения, логгер

Номер варианта согласовывается с преподавателем

Максимальное количество баллов за работу – 60.

Номер окончательной версии:

1 версия – коэффициент 1, 2 версия – коэффициент 0,6, 3 версия – коэффициент 0,4.

Дата сдачи

До указанной даты – коэффициент 1, опоздание до 1 недели – коэффициент 0,6, позже – коэффициент 0,4.

Варианты:

**1. Факультатив.** Существует перечень **Курсов**, за каждым из которых закреплён один **Преподаватель**. **Студент** записывается на один или несколько **Курсов**, данные о регистрации сохраняются. По окончании обучения **Преподаватель** выставляет **Студенту** оценку, которая сохраняется в **Архиве**.

**Часть 1** Создать набор объектов **Курсы**, провести сортировку и поиск объекта в наборе.

**Часть 2**

Вывести информацию о количестве студентов записавшихся на заданный курс.

Вывести фамилии преподавателей, ведущих более одного курса.

Вывести информацию об оценках заданного студента.

Открыть новый курс, назначить дату начала занятий курса.

**2. Платежи.** **Администратор** отвечает за ведение каталога **Счетов** в банке. **Клиент** имеет одну или несколько **Кредитных Карт (КК)**, каждая из которых соответствует некоторому **Счету**. **Клиент** может при помощи **КК** сделать **Платеж** на другой **Счет**, аннулировать **Счет**.

**Часть 1** Создать набор объектов **Счета**, провести сортировку и поиск объекта в наборе.

**Часть 2**

Вывести информацию о счетах заданного клиента.

Вывести сумму платежей заданного клиента за указанный период.

Вывести сумму на счете заданного клиента.

Сделать платеж на другой счет; аннулировать счет.

**3. Больница.** **Врач** регистрирует поступившего **Пациента**, определяет предварительный диагноз. **Врач** может сделать назначение **Пациенту** (процедуры,

лекарства, операции), регистрируя его в **Листе назначений**. Пациент может просмотреть список своих назначений. Пациент может быть выписан из **Больницы**.  
**Часть 1** Создать набор объектов **Назначения**, провести сортировку и поиск объекта в наборе.

#### **Часть 2**

Вывести информацию обо всех пациентах заданного врача.  
Вывести информацию о диагнозе заданного пациента.  
Вывести информацию обо всех назначениях заданного пациента.  
Сделать назначение пациенту, выписать пациента.

**4. Вступительные экзамены.** Абитуриент регистрируется на один из **Факультетов** с фиксированным планом набора, сдает экзамены по соответствующим **Предметам**. Результаты экзаменов **Администратором** регистрируются в **Ведомости**. Система подсчитывает средний балл и определяет **Абитуриентов**, зачисленных в учебное заведение.

**Часть 1** Создать набор объектов **Ведомость**, провести сортировку и поиск объекта в наборе.

#### **Часть 2**

Вывести информацию обо всех абитуриентах, зарегистрировавшихся на факультет.  
Вывести информацию об абитуриентах, со средним баллом выше среднего по заданному факультету.  
Вывести информацию о среднем балле по заданному факультету.  
Зарегистрировать нового абитуриента, зафиксировать результаты сдачи экзамена по предмету.

**5. Библиотека.** Читатель имеет возможность осуществлять поиск **Книг** в **Каталоге**. Книга может присутствовать в **Библиотеке** в одном или нескольких экземплярах. Библиотекарь выдает Читателю Книгу, регистрируя это в **Формуляре**.

**Часть 1** Создать набор объектов **Книги**, провести сортировку и поиск объекта в наборе.

#### **Часть 2**

Вывести информацию о наличии свободных экземпляров заданной книги.  
Вывести информацию о читателях, которые имеют задолженность более 1 месяца.  
Вывести информацию о книгах заданного автора.  
Выдать книгу читателю, списать экземпляр книги.

**6. Конструкторское бюро.** Заказчик представляет **Техническое Задание (ТЗ)**, в котором перечислен перечень **Работ** с указанием квалификации и количества требуемых специалистов. **Администратор** рассматривает **ТЗ** и оформляет **Проект**, назначая на него незанятых **Работников** требуемой квалификации, после чего рассчитывается стоимость **Проекта** и **Заказчику** выставляется **Счет** из расчета заработной платы назначенных специалистов.

**Часть 1** Создать набор объектов **Проекты**, провести сортировку и поиск объекта в наборе.

#### **Часть 2**

Вывести информацию обо всех заданиях указанного заказчика.  
Вывести информацию о требуемых специалистах указанного задания.  
Назначить в проект работников.  
Вывести информацию о неоплаченных проектах.

**7. Телефонная станция.** Администратор осуществляет подключение **Абонентов**. Абонент может выбрать одну или несколько из предоставляемых **Услуг**. Абонент

оплачивает **Счет** за разговоры и **Услуги**. **Администратор** может просмотреть список неоплаченных **Счетов** и заблокировать **Абонента**.

**Часть 1** Создать набор объектов **Сервисы**, провести сортировку и поиск объекта в наборе.

**Часть 2**

Вывести информацию об услугах, используемых заданным абонентом.

Вывести счет заданного абонента.

Вывести список доступных услуг.

Оплатить счет, заблокировать абонента.

**8. Автобаза.** **Диспетчер** распределяет **Заявки** на **Рейсы** между **Водителями**, за каждым из которых закреплен свой **Автомобиль**. На **Рейс** может быть назначен **Автомобиль**, находящийся в исправном состоянии и характеристики которого соответствуют **Заявке**. **Водитель** делает отметку о выполнении **Рейса** и состоянии **Автомобиля**.

**Часть 1** Создать набор объектов **Заявки**, провести сортировку и поиск объекта в наборе.

**Часть 2**

Вывести список рейсов.

Вывести информацию обо всех рейсах заданного водителя.

Вывести список автомобилей, находящихся в ремонте.

Назначить водителя на рейс, поставить автомобиль на ремонт.

**9. Интернет-магазин.** **Администратор** осуществляет ведение каталога **Товаров**. **Клиент** формирует **Заказ** на произвольный перечень **Товаров**. **Администратор** просматривает список сделанных **Заказов**, определяет время **Доставки Товара**, регистрирует оплату по факту **Доставки**.

**Часть 1** Создать набор объектов **Товары**, провести сортировку и поиск объекта в наборе.

**Часть 2**

Вывести информацию обо всех заказах заданного клиента.

Вывести информацию о заказах, которые должны быть выполнены к заданному сроку.

Вывести список заказов, доставка которых задержана.

Сделать заказ, зарегистрировать оплату заказа.

**10. Система Ресторан.** **Клиент** осуществляет заказ из **Меню**. **Администратор** подтверждает **Заказ** и отправляет его на кухню для исполнения. **Администратор** выставляет **Счет**. **Клиент** производит его оплату.

**Часть 1** Создать набор объектов **Меню**, провести сортировку и поиск объекта в наборе.

**Часть 2**

Вывести информацию обо всех заказах клиента.

Вывести информацию о меню.

Вывести информацию о подтвержденных заказах.

Сделать заказ, оплатить заказ.

**11. Авиакомпания..** **Администратор** осуществляет назначение **Самолета** на **Рейс** с учетом вместимости и дальности полета и назначает на **Рейс** одну из летных **Бригад**. **Бригада** формируется **Командиром** и состоит из пилотов, штурмана, радиста, стюардесс. **Администратор** может отменить **Рейс** из-за погодных условий, либо изменить в полете **Аэропорт** назначения из-за технических неисправностей.

**Часть 1** Создать набор объектов **Рейсы**, провести сортировку и поиск объекта в наборе.

**Часть 2**

Вывести информацию о бригаде заданного рейса.

Вывести список всех рейсов.

Изменить аэропорт назначения рейса, отменить рейс.

Вывести информацию о задержанных рейсах.

**12. Периодические издания.** Администратор осуществляет ведение каталога периодических **Изданий**. Читатель может оформить **Подписку**, предварительно выбрав периодические **Издания** из списка. Система подсчитывает сумму для оплаты и регистрирует **Платеж**

**Часть 1** Создать набор объектов **Издания**, провести сортировку и поиск объекта в наборе.

**Часть 2**

Вывести список всех изданий.

Вывести список изданий заданного читателя.

Вывести список неоплаченных подписок.

Провести платеж, зарегистрировать издание.

**13. Заказ гостиницы.** Клиент заполняет **Заявку**, указывая количество мест в номере, класс апартаментов и время пребывания. Администратор просматривает поступившую **Заявку**, выделяет наиболее подходящий из доступных **Номеров**, после чего система выставляет **Счет Клиенту**. Оплата **Счета Клиентом** регистрируется системой, также Администратор может просмотреть неоплаченные **Счета**.

**Часть 1** Создать набор объектов **Номера**, провести сортировку и поиск объекта в наборе.

**Часть 2**

Вывести список свободных номеров заданного класса.

Вывести список неоплаченных счетов.

Забронировать номер, выставить счет клиенту.

Вывести список заявок.

**14. Жилищно-коммунальные услуги.** Квартиросъемщик отправляет **Заявку**, в которой указывает род работ, масштаб, и желаемое время выполнения. Диспетчер рассматривает **Заявку** и назначает **Рабочего** соответствующего профиля, регистрируя назначение в **Плане работ**. Диспетчер может отклонить **Заявку**, в случае занятости всех **Рабочих**.

**Часть 1** Создать набор объектов **Заявки**, провести сортировку и поиск объекта в наборе.

**Часть 2**

Вывести список заявок.

Вывести список рабочих, выполняющих заданную работу.

Вывести список просроченных заявок.

Назначить рабочего на выполнение работы, отклонить заявку.

**15. Прокат автомобилей.** Клиент оставляет **Заявку** на **Автомобиль** конкретной модели, указывая паспортные данные, срок аренды. Администратор оформляет **Заказ**, выбрав **Автомобиль**, соответствующий **Заявке** из списка доступных, после чего **Клиенту** выставляется **Счет**. По факту возврата Администратором фиксируется текущее состояние **Автомобиля**.

**Часть 1** Создать набор объектов **Автомобили**, провести сортировку и поиск объекта в наборе.

**Часть 2**

Вывести список заявок на автомобиль заданной модели.

Вывести информацию о количестве доступных автомобилей заданной модели.

Вывести текущее состояние заданного автомобиля.

Оформить заказ, оплатить счет.

**16. Скачки.** Клиент может делать **Ставки** на разных **Лошадей Забега**. Лошадь не может участвовать в нескольких **Забег**ах в один день. **Администратор** фиксирует список, состав и результаты **Забегов** на день.

**Часть 1** Создать набор объектов **Забег**и, провести сортировку и поиск объекта в наборе.

**Часть 2**

Вывести список лошадей заданного забега.

Вывести список забегов на указанную дату.

Вывести список выигравших клиентов забега.

Зафиксировать состав и результаты забега.

**17. Тестирование.** Тьютор создает **Тест** из нескольких **Вопросов** закрытого типа (выбор одного или более вариантов из N предложенных) по определенному **Предмету**. Тьютор назначает Тест Студенту. Студент просматривает список доступных **Тестов**, отвечает на **Вопросы**. Система подсчитывает результаты, заносит их в **Ведомость**.

**Часть 1** Создать набор объектов **Тесты**, провести сортировку и поиск объекта в наборе.

**Часть 2**

Вывести список доступных тестов.

Вывести список вопросов теста.

Вывести результаты теста заданного студента.

Назначить тест студенту, пройти тест, записать результаты теста (просуммировать результат ответа на каждый вопрос теста).

**18. Кофе-машина.** Пользователь обладает **Счетом**. Кофе-машина содержит набор **Напитков**, с заданным числом порций и дополнительных **Ингредиентов**. Пользователь может купить один или несколько **Напитков**. **Администратор Кофе-машины** осуществляет ее наполнение.

**Часть 1** Создать набор объектов **Напитки**, провести сортировку и поиск объекта в наборе.

**Часть 2**

Вывести список доступных напитков.

Вывести список доступных ингредиентов.

Вывести сумму счета заданного пользователя.

Наполнить кофе-машину, купить напиток (напитки).

**19. Парк.** Владелец парка дает указания **Леснику** о высадке (лечении, художественной обработке, уничтожении) **Растений**. Лесник отчитывается о выполнении. Владелец просматривает результаты и подтверждает исполнение.

**Часть 1** Создать набор объектов **Растения**, провести сортировку и поиск объекта в наборе.

**Часть 2**

Вывести список имеющихся в парке растений.

Вывести список возможных действий с растениями..

Вывести список заданий для лесника.

Назначить задание леснику, записать результаты выполнения задания.

**20. Турагентство.** Клиент выбирает и оплачивает **Тур** (отдых, экскурсия, шоппинг). Турагент имеет возможность определить тур как горящий, размер скидки постоянным клиентам.

**Часть 1** Создать набор объектов **Туры**, провести сортировку и поиск объекта в наборе.

**Часть 2**

Вывести информацию обо всех турах агентства.

Вывести информацию обо всех заказах заданного клиента.

Вывести информацию о горящих турах заданного типа.

Сделать заказ тура, установить скидку клиенту, имеющему более N оплаченных заказов.

**21. Система Команда разработчиков.** Заказчик представляет техническое задание (ТЗ), в котором перечислен перечень **Работ** с указанием квалификации и количества требуемых специалистов. Менеджер рассматривает ТЗ и оформляет **Проект**, назначая на него незанятых **Разработчиков** требуемой квалификации, после чего рассчитывается стоимость **Проекта** и **Заказчику** выставляется **Счет**. **Разработчик** имеет возможность отметить количество часов, затраченных на работу над проектом.

**Часть 1** Создать набор объектов **Разработчики**, провести сортировку и поиск объекта в наборе.

**Часть 2**

Вывести информацию обо всех проектах для заданного заказчика.

Вывести информацию обо всех разработчиках команды.

Вывести информацию о разработчиках по заданному проекту.

Назначить разработчика на проект, оплатить счет.

**22. Videотека.** В БД хранится информация о видеотеке: фильмы, актеры, режиссеры. Для **фильмов** необходимо хранить: название, имена актеров, дату выхода; страну, в которой выпущен фильм, рейтинг. Для **актеров** и **режиссеров** необходимо хранить: ФИО; дату рождения. Администратор обновляет информацию о рейтинге фильмов

**Часть 1** Создать набор объектов **Фильмы**, провести сортировку и поиск объекта в наборе.

**Часть 2**

Вывести информацию об актерах, снимавшихся в заданном фильме, информацию о фильмах с наивысшим рейтингом.

Вывести информацию об актерах, снимавшихся как минимум в N фильмах.

Вывести информацию об актерах, которые были режиссерами хотя бы одного из фильмов.

Добавить новый фильм, удалить все фильмы, дата выхода которых была более заданного числа лет назад.

**23. Расписание занятий.** В БД хранится информация о **преподавателях**, проводимых ими **занятиях** по **дисциплинам** и студенческих **группах**, посещающих занятия. Для **занятий** необходимо хранить: название; время проведения (день недели); аудитории, в которых проводятся занятия. Для **преподавателей** необходимо хранить: ФИО; **дисциплины**, которые он ведет. Для **группы** необходимо хранить количество студентов и дисциплины, которые группа посещает. Для дисциплины – название и количество часов.

**Часть 1** Создать набор объектов **Дисциплины**, провести сортировку и поиск объекта в наборе.

**Часть 2**

Вывести информацию о преподавателях, работающих в заданный день недели в заданной аудитории.

Вывести список дисциплин заданной группы.

Вывести дни недели, в которых занято заданное количество аудиторий.

Назначить преподавателя на дисциплину, перенести первые занятия заданных дней недели на последнее место.

**24. Погода.** В БД хранится информация о **погоде** в различных **регионах**. Для **погоды** необходимо хранить: регион; дату; температуру; осадки. Для **регионов** необходимо хранить: название; площадь; **тип** жителей. Для **типов** жителей необходимо хранить: название; язык общения, количество.

**Часть 1** Создать набор объектов **Регионы**, провести сортировку и поиск объекта в наборе.

**Часть 2**

Вывести сведения о погоде в заданном регионе.

Вывести даты, когда в заданном регионе шел снег и температура была ниже заданной отрицательной.

Вывести информацию о погоде за прошедшую неделю в регионах, жители которых общаются на заданном языке.

Обновить информацию о погоде для заданного региона, добавить новый регион.

**25. Система LowCost-Авиакомпания.** Клиент заказывает и оплачивает **Билет** на **Рейс** с учетом наличия\отсутствия багажа и права первоочередной регистрации и посадки (Цена Билета может быть ниже стоимости провоза багажа). С приближением даты **Рейса** или наполнением самолета, цена на **Билет** может повышаться.

**Часть 1** Создать набор объектов **Рейсы**, провести сортировку и поиск объекта в наборе.

**Часть 2**

Вывести количество билетов на заданный рейс.

Вывести допустимый размер багажа и стоимость провоза багажа на заданный рейс.

Вывести процент увеличения стоимости билета на заданный рейс перед датой вылета по сравнению с первоначальной стоимостью. Добавить рейс в систему, обновить информацию о количестве свободных мест на рейс.

**26. Система Зоопарк.**

В БД хранится информация о **животных**. У животного существует определенный **Уход** (условия содержания) и **Рацион** питания.

**Директор** зоопарка дает указания **Смотрителю** об уходе (лечении, питании, содержании) за **Животными**. **Смотритель** отчитывается о выполнении. **Директор** просматривает результаты и подтверждает исполнение.

**Часть 1** Создать набор объектов **Животные**, провести сортировку и поиск объекта в наборе.

**Часть 2**

Вывести список имеющихся в зоопарке животных.

Вывести список возможных действий с животными.

Вывести список заданий для смотрителя.

Назначить задание смотрителю, записать результаты выполнения задания.