

Домашнее задание №3. Spring

Данное задание является завершающим в курсе и включает несколько целей:

1. Отработка навыка использования фреймворка внедрения зависимостей Spring
2. Отработка навыка разработки приложений на слабосвязных компонентах
3. Отработка навыка работы с базами данных из языка Java.

Задание состоит из двух частей. В первой части необходимо разработать приложение на базе фреймворка Spring, которое будет состоять из набора компонент, каждая из которых отвечает за свою часть логики системы, а их совокупность представляет собой все приложение. Обратите внимание, что все компоненты разработанной системы должны быть связаны, то есть они не могут знать о классах друг друга (но могут знать об именах или других настройках).

Первая часть задания предполагает чтение данных из файлов и запись данных в БД. Для работы с файлами могут быть использованы классы **File** и **Scanner**. Работа с базой данных выполняется стандартными средствами Spring JDBC, при этом конкретная выбранная СУБД не оказывает какого-либо влияния на характер выбранной работы.

За первую часть задания можно получить четыре балла. В случае, если работа выполнена не до конца (не выполнены какие-либо требования) но имеет принципиально правильный подход, по решению проверяющего балл может быть снижен.

Во второй части задания, необходимо разработать аннотацию и определить способ обработки. Вторая часть задания может быть решена даже без решения первой. За это задание начисляется два балла.

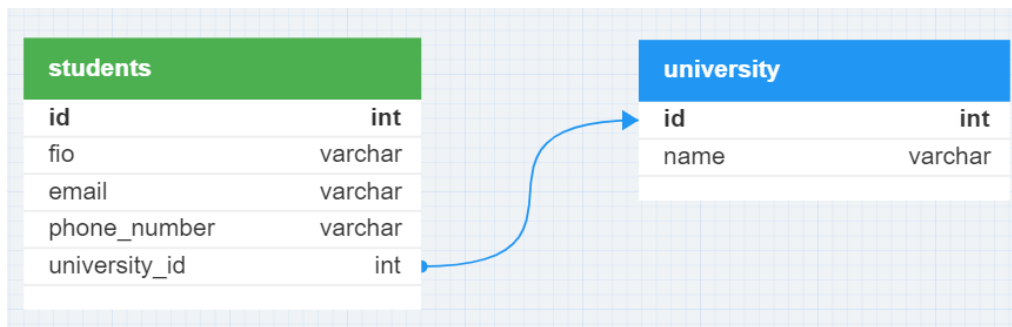
Итого, за работу можно получить шесть баллов

Часть 1. Ядро системы (4 балла)

Дан набор текстовых документов, каждый из которых содержит одну или несколько записей о студентах следующего вида:

Фамилия Имя Отчество номер_телефона email наименование_вуза

Необходимо разработать приложение, которое будет получать на вход наименование папки, в которой расположены текстовые файлы, получать из файлов все значения и записывать их в базу данных со следующей схемой таблиц:



Приложение должно состоять из набора слабосвязных компонент, каждый из которых отвечает за собственную часть логики переноса данных, в частности должны быть разработаны следующие компоненты:

- 1) Компонент чтения данных из файлов
- 2) Компонент записи данных в БД
- 3) Промежуточные компоненты

В настоящий момент времени число и состав промежуточных компонент фиксировано, однако в дальнейшем может потребоваться создание дополнительных компонент. Система должна предоставлять такую архитектуру, которая позволит внедрить новые компоненты без необходимости внесения изменений в уже существующие, а также гибко настраивать порядок следования компонент.

Базовая реализация компонент включает следующее:

- 1) Компонента чтения данных получает адрес папки файловой системы, сканирует имеющиеся там файлы и возвращает прочитанные строки
- 2) Компонента записи данных получает строку подключения к базе, включая адрес, имя пользователя и пароль, а затем записывает получаемых ей студентов в указанную базу данных.
- 3) Промежуточная компонента проверки данных исправляет ФИО так, чтобы каждый его компонент начинался с большой буквы, а почта и телефон соответствовали стандартному шаблону. Если кто-то из них не корректен, то информацию об этом необходимо залогировать, но не записывать в базу.
- 4) Промежуточная компонента проверки вуза проверяет, правильность записи вуза. Один и тот же вуз может быть записан как большими, так и маленькими буквами, а также иметь какие-либо дополнение или нет. Например: МФТИ НИУ и мфти – это один и тот же вуз. Предложите несколько правил подобной проверки в этой компоненте, и приведите вуз к одному выбранному названию.

Приложите к выполненному заданию следующее:

- 1) Набор модульных тестов, со стопроцентным покрытием каждой компоненты
- 2) Набор интеграционных тестов, проверяющих интеграцию всех компонент
- 3) Документацию для всех нетривиальных, на ваш взгляд, частей системы.
- 4) Скрипт разворачивания и запуска приложения, а также тестовые данные, позволяющие продемонстрировать работу приложения
- 5) В случае, если СУБД не может быть представлена библиотекой в составе приложения (как H2 например), следует приложить докер образ развернутой системы.

Часть 2. Логирование (2 балла)

Разработайте аннотацию **@LogTransformation**, которая может быть использована для аннотирования промежуточных компонент системы. Если промежуточный компонент имеет такую аннотацию, то после завершения выполнения операции в лог записывается дата-время начала операции, название класса компоненты, данные полученные на вход операции, данные возвращаемые в результате операции. Название лог файла может быть указано в качестве параметра аннотации.