Java Beans

Java Collections
6 уровень, 2 лекция

ОТКРЫТА

— Вот ты где.

Я тут подумал и решил устроить тебе еще одну маленькую лекцию, которая будет тебе очень полезна. Пока ты не работал программистом, ты, скорее всего, не сталкивался со специальной терминологией, и я хочу тебя сейчас познакомить с несколькими распространёнными понятиями.

Лет 10 назад массовое распространение получила концепция **EJB** – **E**nterprise **J**ava **B**eans.

— А что значит Java Beans?

— Bean по-английски боб. А Java Beans – это, стало быть, кофейные бобы (Java – сорт кофе). Такой айтишный юмор.

Бизнес-логику программы представляли в виде набора высокоуровневых объектов – бинов, которые умели обмениваться сообщениями, сохранять себя, находить друг друга по имени, и еще кучу всего. Обычно это достигалось за счет специального супер-навороченного родительского класса, хотя были и другие подходы. Поведение таких объектов очень регламентировалось.

Три самых известных вида EJB-бинов:

Entity Bean – бин, цель которого — хранить некоторые данные. В логику такого бина встроен механизм сохранения себя и своих полей в базу данных. Такой объект может быть уничтожен, а потом воссоздан из базы заново. Но кроме хранения данных у него нет никакой логики.

Session Beans делятся на две категории.

Stateless Session Bean – это бин, который не хранит во внутренних переменных важных данных, нужных для его работы. Такой бин можно уничтожить, а затем заново создать, и он будет выполнять свою функцию, как и раньше.

Statefull Session Bean – это бин, который хранит у себя внутри данные, которые использует при работе. Если мы вызываем методы этого бина, то в каждом следующем вызове он может использовать часть данных, переданных ему в предыдущих. И все равно этот бин – это не то же самое, что обычный объект.

Но в использовании бинов тоже было не все так радужно, поэтому скоро маятник качнулся в обратную сторону. И разработчики стали все чаще использовать обычные объекты. Им даже придумали специальное название.

POJO (**P**lain **O**ld **J**ava **O**bject) – старый обычный Java-объект. Такие объекты не обладали какими-то суперфункциями и не наследовались от суперобъектов. Самые обычные Java-объекты.

Когда ты познакомишься с EJB на практике, ты поймешь, в чем разница. Грубо говоря, POJO – это нож, а EJB – это швейцарский нож, по которому можно еще и звонить.

— Интересное сравнение.

— Да, вот еще что.

Со временем в назначении объектов/классов возникла специализация. Как результат – выделились некоторые роли, объекты которых получили новые названия.

DTO — Data Transfer Object – объект, который создается с целью быть использованным при транспортировке данных. Обычно к таким объектам два требования: а) уметь хранить данные, б) уметь сериализоваться. Т.е. их используют только для пересылки данных.

Создал объект, записал в него нужные данные из бизнес-логики, сериализовал в JSON/XML и отправил куда-надо. Или наоборот – пришло сообщение – десериализовал его в DTO-объект и вытягивай из него данные.

Entity – это объект, который хранится в базе данных. Но они не содержат никакой бизнес-логики. Можно сказать, что это – данные бизнес-модели.

Есть еще **DAO** – Data Access Object. Задача DAO — сохранять объекты в базу и доставать их из нее.

Пример:

Взаимодействие DAO и Entity

```
UserEntity user = UserDAO.getUserById("1535");
if (user.getAge()>18)
{
 user.setMobilization(true);
 UserDAO.save(user);
}
```

Комментарии

```
UserEntity – это класс, который хранит данные о пользователе (User-Entity)
UserDAO – это класс, который достает данные (объекты UserEntity) из базы и сохра
```

На этом все.

Пусть это и небольшая ознакомительная лекция, но больше ты сейчас все равно не поймешь. Каждую из этих тем можно днями рассказывать, а EJB – годами.

Но я хочу, чтобы ты хотя бы представлял, о чем речь, если столкнёшься с такими вещами в разговоре, переписке, на форуме или на собеседовании.

— Гм. Спасибо, Билаабо. Да, думаю, технических терминов мне не хватает. Спасибо большое тебе еще раз.

‹ Предыдущая

JavaCoder

Введите текст комментария

Anonymous #3113984 Уровень 51, Ukraine — 24 августа, 13:24

[Разница между микросервисами и EJB]

[+ еще немного инфы]

[+ Сергей Немчинский об EJB]

POJO-объекты

[Разница между POJO и Java Bean]

Ответить +1

Regina C QA Auto Engineer в - — 30 мая 2021, 16:44

про DTO и POJO понравилось вот здесь - [Наглядный пример различия DTO, POCO (POJO) и Value Object]

Ответить +8

E K Уровень 41, Краснодар, Россия — 25 июня 2021, 15:54

Спасибо! Хорошая статейка

Ответить 0

Булат Уровень 37, Москва — 13 мая 2021, 12:42

>

```
1 if (user.getAge()>18)
2 {
3  user.setMobilization(true);
```

> товарищ майор добрался до Java, черт

Ответить +13

hidden #2322530 Уровень 41 — 9 сентября 2020, 15:45

все в духе JavaRush, вначале на 35 вы решите тестовое задание на стажировку, а в следующих лекциях мы только вам расскажем что к чему там вообще

Ответить +3

Хорс Уровень 41, Харьков — 2 сентября 2020, 18:20

Ответить

+7

Валентин Кудинов Уровень 41, Самара, Россия

11 апреля 2020, 17:17

После гугления отметил для себя что Бин это просто класс написанный по определённым правилам. Например должен быть конструктор по умолчанию и геттеры и сеттеры. Бины использует спринг. Сначала при настройке спринга мы указываем ему где лежат наши объекты бины. Потом когда нам понадобится объект мы говорим спрингу дай такой то объект он сам его создаст и предоставит ссылку.

Ответить

+40

Vorlock Уровень 31, Днепр, Украина

29 января 2020, 20:45

в блоке "Комментарии" неплохо бы использовать перенос строк или явно подсвечивать ползунок

Ответить

+7

skrskr Уровень 39, Санкт-Петербург, Россия

20 декабря 2019, 22:50

Если кому интересно зачем это всё и почему, то советую погуглить Spring Framework и EJB (аналоги, хотя спринг более мейнстримовый). По факту, это то чем вы будете заниматься на работе, как Java девелопер.

Ответить

0

Даниил Salesforce Developer в **Viseven** MASTER

10 сентября 2019, 20:46

Интересно к чему это всё было... Даже толком без примеров, наверное что-то достаточно "абстрактное"...

Ответить

0

Vitaly Khan Java Developer в **Onollo** MASTER

22 декабря 2019, 06:23

темы слишком обширные, чтобы на каждую давать примеры. для введения пойдет.
на стажировке или на работе скорее всего сразу будете иметь дело с многими из перечисленных.

Ответить

0

Nail Уровень 23, Кельн

23 августа 2019, 20:45

"У каждого Session Bean есть своя функция. Один делает одно, другой другое."
-А где это применяется?
-То там, то сям

Ответить

+35

Pig Man Главная свинья в **Свинарнике**

12 февраля 2021, 23:43

"*Но я хочу, чтобы ты хотя бы представлял, о чем речь, если столкнёшься с такими вещами в разговоре, переписке, на форуме или на собеседовании*" от наших познаний все будут в восторге

Ответить

+1

ОБУЧЕНИЕ

Курсы программирования

Курс Java

Помощь по задачам

Подписки

Задачи-игры

СООБЩЕСТВО

Пользователи

Статьи

Форум

Чат

Истории успеха

Активности

КОМПАНИЯ

О нас

Контакты

Отзывы

FAQ

Поддержка

RUSH

JavaRush — это интерактивный онлайн-курс по изучению Java-программирования с нуля. Он содержит 1200 практических задач с проверкой решения в один клик, необходимый минимум теории по основам Java и мотивирующие фишки, которые помогут пройти курс до конца: игры, опросы, интересные проекты и статьи об эффективном обучении и карьере Java-девелопера.

НАЧАТЬ ОБУЧЕНИЕ

ЯЗЫК ИНТЕРФЕЙСА

Русский

СКАЧИВАЙТЕ НАШИ ПРИЛОЖЕНИЯ