

Порождающие паттерны, часть 2

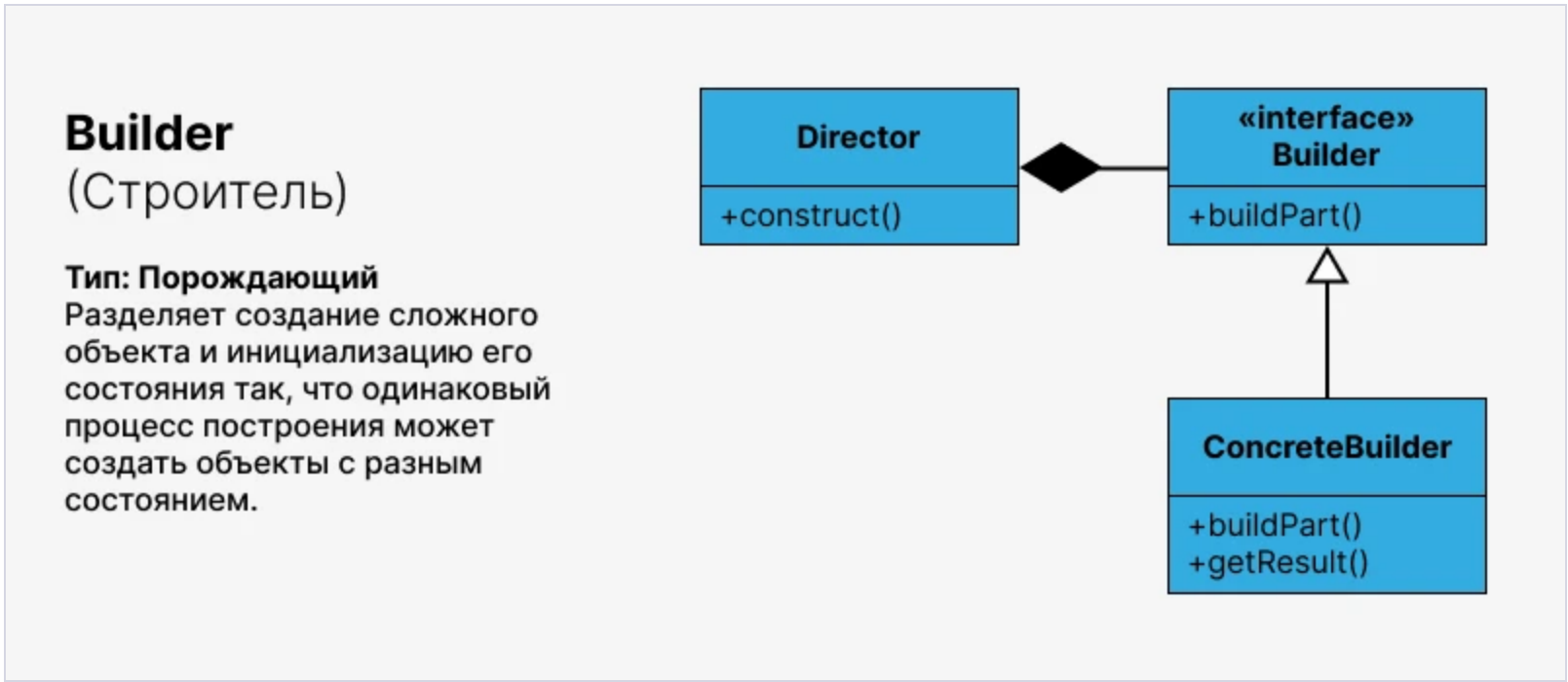
JSP & Servlets
16 уровень, 3 лекция

ОТКРЫТА

4.1 Builder

Строитель (Builder) — порождающий шаблон проектирования, который предоставляет способ создания составного объекта.

Отделяет конструирование сложного объекта от его представления так, что в результате одного и того же процесса конструирования могут получаться разные представления.



Сильные стороны:

- позволяет изменять внутреннее представление продукта;
- изолирует код, реализующий конструирование и представление;
- дает более тонкий контроль над процессом конструирования.

Слабые стороны:

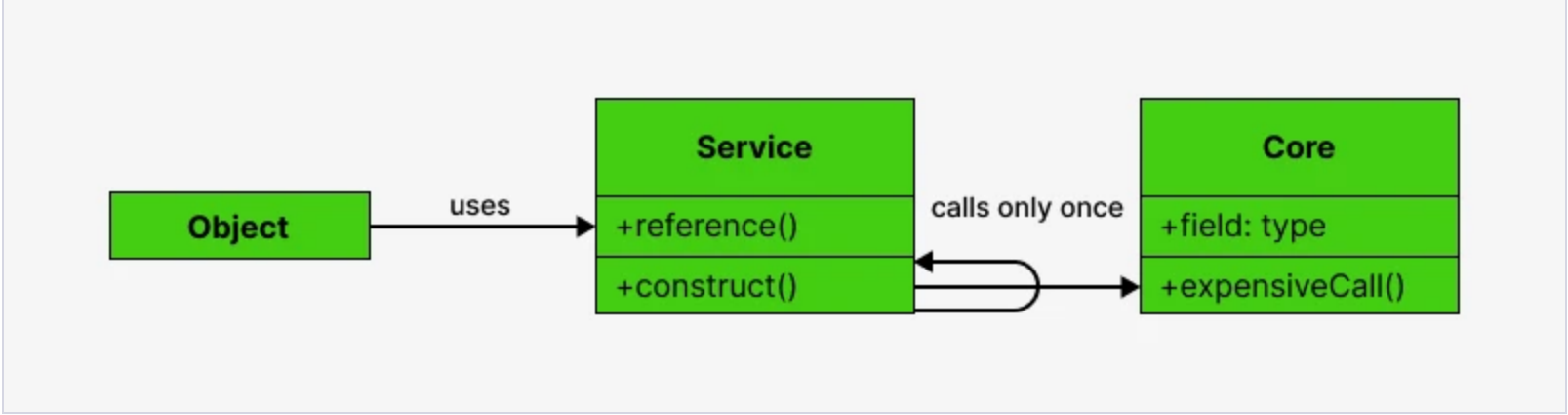
- алгоритм создания сложного объекта не должен зависеть от того, из каких частей состоит объект и как они стыкуются между собой;
- процесс конструирования должен обеспечивать различные представления конструируемого объекта.

Хороший пример – это класс `HttpRequest`, у него есть подкласс `HttpRequest.Builder`, с помощью которого можно создавать экземпляры класса `HttpRequest` и гарантировать их валидность.

4.2 Lazy Initialization

Отложенная (ленивая) инициализация (Lazy initialization) — приём в программировании, когда некоторая ресурсоёмкая операция (создание объекта, вычисление значения) выполняется непосредственно перед тем, как будет использован её результат.

Таким образом инициализация выполняется “по требованию”, а не заблаговременно. Аналогичная идея находит применение в самых разных областях: например, компиляция “на лету” и логистическая концепция “Точно в срок”.



Частный случай ленивой инициализации — создание объекта в момент обращения к нему — является одним из порождающих шаблонов проектирования. Как правило он используется в сочетании с такими шаблонами, как Фабричный метод, Одиночка и Заместитель.

Сильные стороны:

- Инициализация выполняется только в тех случаях, когда она действительно необходима;
- Ускоряется начальная инициализация приложения: все, что можно отложить, откладываем.

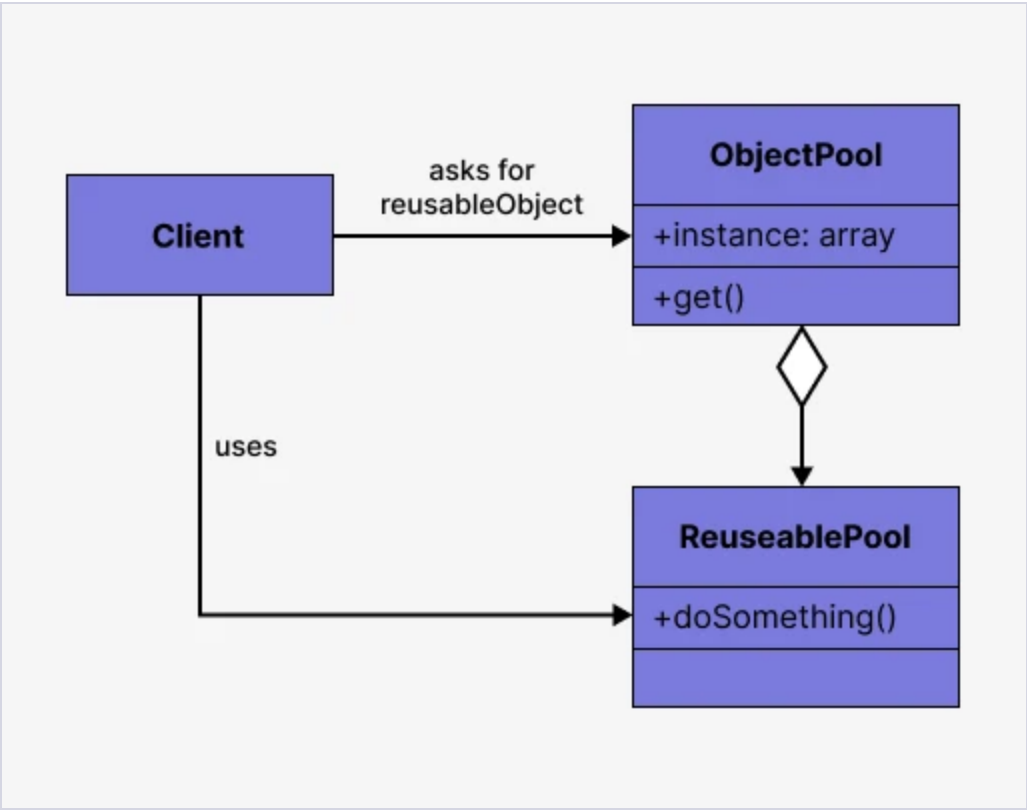
Слабые стороны:

- Невозможно явным образом задать порядок инициализации объектов;
- Возникает задержка при первом обращении к объекту, что может оказаться критичным при параллельном выполнении другой ресурсоёмкой операции. Из-за этого требуется тщательно просчитывать целесообразность использования “ленивой” инициализации в многопоточных программных системах.

Помнишь, как при написании web.xml там можно было указать порядок старта сервлетов? Это как раз и есть следствие ленивой загрузки. Tomcat создаст объекты сервлетов при первом к ним обращении.

4.3 Object pool

Объектный пул (object pool) — порождающий шаблон проектирования, набор инициализированных и готовых к использованию объектов. Когда системе требуется объект, он не создаётся, а берётся из пула. Когда объект больше не нужен, он не уничтожается, а возвращается в пул.



Объектный пул применяется для повышения производительности, когда создание объекта в начале работы и уничтожение его в конце приводит к большим затратам. Особенно заметно повышение производительности, когда объекты часто создаются-уничтожаются, но одновременно существует лишь небольшое их число.

Объектный пул удобен, если объект владеет другими ресурсами, кроме памяти — например, сетевыми сокетами. Либо если коллекция объектов занимает значительную часть памяти компьютера и “мусора” создаётся действительно много.

Как ты помнишь, Tomcat выполняет каждый запрос в отдельном потоке. Но потоки не создаются каждый раз заново, а хранятся в пуле потоков. Это позволяет быстрее выполнять запросы: когда поток нужен, он просто берется из пула. Кстати, вопрос: а как бы ты поместил запущенные поток в пул и взял его из пула?

−

+13

+

Комментарии (4)

популярные

новые

старые

JavaCoder

Введите текст комментария

Александр

Уровень 85

EXPERT

31 октября 2022, 17:37

⋮

Последняя задача hard. Даже забыл про метод offer у Queue или даже не знал про него вовсе. Помню: add peek, poll

Ответить

−

0

+

Александр Ц.

System Engineer

EXPERT

26 октября 2022, 11:58

⋮

Задача "Ленивый зам": в условиях не указано изменить id на 10001.

Ответить

−

0

+

Сергей

Уровень 79

14 ноября 2022, 19:41

⋮

Так и не пофиксили, может кому пригодится:
Вместо
new LazyInitializer(100001);
нужно писать
new LazyInitializer(10001);

Ответить

−

0

+

Oleg Khilko

Уровень 51

15 августа 2022, 19:25

⋮

Lazy Initialization == @Lazy в Спринге, очень классная аннотация для получения ракетницы Шварца right in time)

Ответить

−

+2

+

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

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.

ПОДПИСЫВАЙТЕСЬ

ЯЗЫК ИНТЕРФЕЙСА

Русский 

СКАЧИВАЙТЕ НАШИ ПРИЛОЖЕНИЯ



"Программистами не рождаются" © 2023 JavaRush