

Java Professional

General and repetition

Generics

Mentor: Rustam Khakov

Generics, Object.equals, Exceptions.

- Generics in Java
- Generic classes
- Generic methods
- Advantages
- Restrictions
- PECS
- Practice

Generics in Java

- Generics - параметризованные типы.
- Можно создавать классы, которые работают с различными типами данных.
- Универсальная сущность.

А зачем? Ведь все это Объекты

- Object является суперклассом всех остальных классов, и ссылка на объект может ссылаться на любой объект.
- Generics добавляют тип безопасности.

Generic classes

Generic classes – универсальный класс

- Точно так же, как не универсальный класс.
 - Отличие - содержит раздел параметров типа.
 - Допустимо несколько типов параметров, разделенных запятой.
 - Классы, которые generic называются параметризованными классами.
-
- `class Test<T> { /// }` – один тип параметра
 - `class Test<T, U>{ /// }` – два типа параметра

Generic Methods

- Работают только со ссылочными типами
- Массивы примитивных типов могут быть переданы в параметр
 - Данные о дженериках стираются во время компиляции
 - `List<Number> number = new ArrayList<Integer>();` - не компилируется

Синтаксис:

- `<T> void getElement(T element){///}`

Wildcards

Можем применить ограничение (Например, мы хотим, чтобы параметризированный класс принимал только Number.)

1. ? extends Number - ограничение сверху
2. ? super Number - ограничение снизу
3. ? - без ограничений
4. <> - diamond operation - magic)))

Advantages

1. Повторное использование кода
2. Безопасность типов: проблемы в коде во время компиляции, а не приводят к ошибкам во время выполнения

Disadvantages

1. Не зная как устроены Generics можно выстрелить себе в ногу
2. Иногда ожидание не соответствует действительности (приведение типов)

Generics Wildcards or PECS (Producer Extends Consumer Super)

принцип гласит:

Если метод имеет аргументы с параметризованным типом (например, `Collection<T>` или `Predicate<T>`), то в случае:

- если аргумент — производитель (producer), нужно использовать ? extends T,
- если аргумент — потребитель (consumer), нужно использовать ? super T.

Производитель(producer) и потребитель(consumer):

- если метод читает данные из аргумента, то этот аргумент — производитель
- если метод передает данные в аргумент, то аргумент является потребителем.

Полезные ссылки

[Lesson: Generics \(Updated\) \(The Java™ Tutorials > Learning the Java Language\) \(oracle.com\)](#)

[Александр Маторин — Неочевидные Дженерики – YouTube](#)

[Java Generics and Collections \[Book\] \(oreilly.com\)](#)