

Дженерики в Java



Наставник: Теплинская Мария Георгиевна

Группа: java-167

Дата: 25.03.2023

Определение дженериков

Дженерики (Обобщения, Generic Types) - это параметризованные типы.

Обобщенными могут быть классы, интерфейсы, методы, где тип данных, с которым работает соответствующая сущность, передается в виде параметра.

Дженерики позволяют реализовывать алгоритмы, которые логически идентичны для разных типов данных, и, что важно, обеспечивают **типовую безопасность** и предохраняет от ошибок времени выполнения (ClassCastException).



Пример дженериков

Since: 1.8

Type parameters: <T> – the type of value

@jdk.internal.ValueBased

```
public final class Optional<T> {
```

Common instance for empty().

2 usages

```
private static final Optional<?> EMPTY = new Optional<> ( value: null);
```

If non-null, the value; if null, indicates no value is present

26 usages

```
private final T value;
```

Returns an empty Optional instance. No value is present for this Optional.

Returns: an empty Optional

API Note: Though it may be tempting to do so, avoid testing if an object is empty by comparing with == or != against instances returned by Optional.empty(). There is no guarantee that it is a singleton. Instead, use isEmpty() or isPresent().

```
public static<T> Optional<T> empty() {  
    /unchecked/  
    Optional<T> t = (Optional<T>) EMPTY;  
    return t;  
}
```



Ограничения дженериков

Нельзя получить экземпляр класс по параметру:

```
T ob = new T(); // ✗
```

Нельзя создавать параметризованные статические переменные:

```
public static T ob; // ✗
```

Нельзя создавать обобщенные массивы:

```
T[] arr = new T[10]; // ✗
```

```
Box<Integer>[] iBoxes = new Box<Integer>[]; // ✗
```

Обобщенный класс не может расширять класс Throwable. Это означает, что *создавать обобщенные классы исключений нельзя (!!!)*.



Объявление обобщенного класса

```
Box<Integer> iBox = new Box<Integer>( value: 12);
```

```
Box<Integer> iBox2 = new Box<>( value: 13);
```

```
var iBox3 = new Box<>( value: 14);
```

```
var iBox4 = new Box<>( value: null);
```



Ограниченные типы

Допустим нам надо ограничить тип, которым мы параметризуем другой тип. Для этого используется следующая нотация:

`<T extends SuperClass>` (верхняя граница включительно)

Это значит, что `T` может быть заменен суперклассом или любым его подклассом.

Также можно ограничивать интерфейсами:

`<T extends SuperClass & Interface1 & Interface2 ...>`



Метасимволы (?)

Метасимвол ? - неизвестный тип, совпадает с любым достоверным объектом класса.

Это значит, что на месте ? может оказаться какой-то из типов, определенных соответствующим параметризованным типом.



Ограниченные метасимволы

Мы можем ограничить метасимвол ? сверху или снизу:

`<? extends SuperClass>`

значит среди возможных типов может быть любой из потомков SuperClass.

`<? super SubClass>`

значит среди возможных типов может быть любой из предков SubClass.

PECS - producer extends consumer super.



Обобщенные методы

Обобщенные методы - это методы, написанные с использованием параметризованного типа. При этом они могут быть как членами обобщенного класса, так и необобщенного.

```
public static <T> void findValue(T value, T[] values) { ... }
```

Примеры вызова:

```
findValue("bad", reviews);  
Utils.<String>findValue("sport", news);
```

Также обобщенными могут быть конструкторы и интерфейсы.



Базовые типы (Raw Types)

Базовый тип - обобщенный класс без аргументов.

Существует для совместимости старых версий Java кода с более новыми.

Не рекомендуется использовать!!!

Нарушает типовую безопасность.



Стирание

При компиляции прикладного кода Java все сведения об обобщенных типах удаляются (стираются). Это означает, что параметры типа сначала заменяются их ограничивающим типом, которым является тип `Object`, если никакого явного ограничения не указано. Затем выполняется требуемое приведение типов, определяемое аргументами типа, для обеспечения совместимости с типами, указанными в этих аргументах. Компилятор также обеспечивает эту совместимость типов.



Полезные ссылки

- Java. Полное руководство - Герберт Шилдт (Глава "Обобщения")
- Официальный Tutorial от Oracle:
<https://docs.oracle.com/javase/tutorial/java/generics/index.html>
- Лекция Тагира Валеева:
<https://www.youtube.com/live/usiKCn7Swxl?feature=share>
- Более глубокая статья на Хабре:
<https://habr.com/ru/company/sberbank/blog/416413/>
- PECS: <https://habr.com/ru/post/559268/>



Спасибо за внимание!

