



# Optional in Java 8 und Java 9

	Сигнатура	Описание (Description)	Семантика типов
Создаём	<b>static &lt;T&gt; Optional&lt;T&gt; of(T value)</b>	Returns an Optional describing the given non-null value.	$(x, \emptyset) \Rightarrow (O(x), \forall)$
	<b>static &lt;T&gt; Optional&lt;T&gt; ofNullable(T value)</b>	Returns an Optional describing the given value, if non-null, otherwise returns an empty Optional	$(x, \emptyset) \Rightarrow (O(x), O(\emptyset))$
	<b>static &lt;T&gt; Optional&lt;T&gt; empty()</b>	Returns an empty Optional instance.	$(x, \emptyset) \Rightarrow (O(\emptyset), O(\emptyset))$
Забираем	<b>Boolean isPresent()</b>	If a value is present, returns true, otherwise false.	$(O(x), O(\emptyset)) \Rightarrow (true, false)$
	<b>T get()</b>	If a value is present, returns the value, otherwise throws NoSuchElementException.	$(O(x), O(\emptyset)) \Rightarrow (x, \forall)$
	<b>T orElse(T other)</b>	If a value is present, returns the value, otherwise returns other.	$(O(x), O(\emptyset)) \Rightarrow (x, t)$
	<b>T orElseGet(Supplier&lt;? extends T&gt; supplier)</b>	If a value is present, returns the value, otherwise returns the result produced by the supplying function.	$(O(x), O(\emptyset)) \Rightarrow (x, s())$
	<b>&lt;X extends Throwable&gt; T orElseThrow(Supplier&lt;? extends X&gt; exceptionSupplier)</b>	If a value is present, returns the value, otherwise throws an exception produced by the exception supplying function.	$(O(x), O(\emptyset)) \Rightarrow (x, \forall)$
Используем	<b>void ifPresentOrElse(Consumer&lt;? super T&gt; action, Runnable emptyAction)</b>	If a value is present, performs the given action with the value, otherwise performs the given empty-based action	$(O(x), O(\emptyset)) \Rightarrow (c(x), a())$
	<b>Void ifPresent(Consumer&lt;? super T&gt; action)</b>	If a value is present, performs the given action with the value, otherwise does nothing.	$(O(x), O(\emptyset)) \Rightarrow (c(x), \emptyset())$
	<b>Optional&lt;T&gt; filter(Predicate&lt;? super T&gt; predicate)</b>	If a value is present, and the value matches the given predicate, returns an Optional describing the value, otherwise returns an empty Optional.	$(O(x), O(\emptyset)) \Rightarrow (?x, O(\emptyset))$
	<b>Optional&lt;T&gt; or(Supplier&lt;? extends Optional&lt;? extends T&gt;&gt; supplier)</b>	If a value is present, returns an Optional describing the value, otherwise returns an Optional produced by the supplying function.	$(O(x), O(\emptyset)) \Rightarrow (O(x), s)$

<b>&lt;U&gt; Optional&lt;U&gt; map(Function&lt;? super T, ? extends U&gt; mapper)</b>	If a value is present, returns an Optional describing (as if by ofNullable(T)) the result of applying the given mapping function to the value, otherwise returns an empty Optional.	$(O(x), O(\emptyset)) \Rightarrow (ofNullable(f(x)), O(\emptyset))$
<b>&lt;U&gt; Optional&lt;U&gt; flatMap(Function&lt;? super T, ? extends Optional&lt;? extends U&gt;&gt; mapper)</b>	If a value is present, returns the result of applying the given Optional-bearing mapping function to the value, otherwise returns an empty Optional.	$(O(x), O(\emptyset)) \Rightarrow (O(f(x)), O(\emptyset))$
<b>Stream&lt;T&gt; stream()</b>	If a value is present, returns a sequential Stream containing only that value, otherwise returns an empty Stream.	$(O(x), O(\emptyset)) \Rightarrow (Stream.of(x), Stream.empty())$

## Где использовать Optional в классе?

