**Методы String**

1. fun compareTo(other: String): Int

Сравнивает этот объект с указанным объектом для заказа. Возвращает ноль, если этот объект равен указанному другому объекту, отрицательное число, если оно меньше другого, или положительное число, если оно больше другого.

1. fun equals(other: Any?): Boolean

Указывает, равен ли какой-либо другой объект этому. Реализации должны соответствовать следующим требованиям:

* Рефлексивный: для любого ненулевого значения x x.equals (x) должен возвращать true.
* Симметричный: для любых ненулевых значений x и y x.equals (y) должен возвращать true тогда и только тогда, когда y.equals (x) возвращает true. Транзитивный: для любых ненулевых значений x, y и z, если x.equals (y) возвращает true, а y.equals (z) возвращает true, то x.equals (z) должен возвращать true.
* Последовательный: для любых ненулевых значений x и y множественные вызовы x.equals (y) последовательно возвращают истину или последовательно возвращают ложь, при условии, что никакая информация, используемая в равных сравнениях для объектов, не изменена. Никогда не равно нулю: для любого ненулевого значения x x.equals (null) должен возвращать false.

1. fun get(index: Int): Char

Возвращает символ этой строки по указанному индексу. Если индекс выходит за пределы этой строки, выдает исключение IndexOutOfBoundsException.

1. fun hashCode(): Int

Возвращает значение хэш-кода для объекта. Общий контракт hashCode: Каждый раз, когда он вызывается для одного и того же объекта более одного раза, метод hashCode должен последовательно возвращать одно и то же целое число, при условии, что никакая информация, используемая в равных сравнениях для объекта, не изменяется. Если два объекта равны согласно методу equals (), то вызов метода hashCode для каждого из двух объектов должен привести к одинаковому целочисленному результату.

1. operator fun plus(other: Any?): String

Возвращает строку, полученную путем объединения этой строки со строковым представлением данного другого объекта.

1. fun subSequence(startIndex: Int, endIndex: Int): [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html)

Возвращает новую последовательность символов, которая является подпоследовательностью этой последовательности символов, начиная с указанного startIndex и заканчивая прямо перед указанным endIndex. Параметры startIndex - начальный индекс (включительно). endIndex - конечный индекс (не включая).

1. fun toString(): String

Возвращает строковое представление объекта.

1. val length: Int

Возвращает длину этой последовательности символов.

**Методы Int**

1.

operator fun compareTo(other: Byte): Int

operator fun compareTo(other: Short): Int

operator fun compareTo(other: Int): Int

operator fun compareTo(other: Long): Int

operator fun compareTo(other: Float): Int

operator fun compareTo(other: Double): Int

Сравнивает это значение с указанным значением для заказа. Возвращает ноль, если это значение равно указанному другому значению, отрицательное число, если оно меньше другого, или положительное число, если оно больше другого.

2. infix fun and(other: Int): Int

Выполняет побитовую операцию И между двумя значениями.

3. operator fun dec(): Int

Уменьшает это значение.

4.

operator fun div(other: Byte): Int

operator fun div(other: Short): Int

operator fun div(other: Int): Int

operator fun div(other: Long): Long

Делит это значение на другое значение, усекая результат до целого числа, которое ближе к нулю.

operator fun div(other: Float): Float

operator fun div(other: Double): Double

Делит это значение на другое значение.

5.

fun equals(other: Int): Boolean

fun equals(other: Any?): Boolean

Указывает, равен ли какой-либо другой объект этому. Реализации должны соответствовать следующим требованиям:

Рефлексивный: для любого ненулевого значения x x.equals (x) должен возвращать true.

Симметричный: для любых ненулевых значений x и y x.equals (y) должен возвращать true тогда и только тогда, когда y.equals (x) возвращает true.

Транзитивный: для любых ненулевых значений x, y и z, если x.equals (y) возвращает true, а y.equals (z) возвращает true, то x.equals (z) должен возвращать true.

Последовательно: для любых ненулевых значений x и y множественные вызовы x.equals (y) последовательно возвращают истину или последовательно возвращают ложь, при условии, что никакая информация, используемая в равных сравнениях для объектов, не изменена.

Никогда не равно нулю: для любого ненулевого значения x x.equals (null) должен возвращать false.

6. fun hashCode(): Int

Возвращает значение хэш-кода для объекта.

Общий контракт hashCode:

Каждый раз, когда он вызывается для одного и того же объекта более одного раза, метод hashCode должен последовательно возвращать одно и то же целое число, при условии, что никакая информация, используемая в равных сравнениях для объекта, не изменяется.

Если два объекта равны согласно методу equals (), то вызов метода hashCode для каждого из двух объектов должен привести к одинаковому целочисленному результату.

7. operator fun inc(): Int

Увеличивает это значение.

1. infix fun or(other: Int): Int

Выполняет побитовую операцию ИЛИ между двумя значениями.

8.

operator fun minus(other: Byte): Int

operator fun minus(other: Short): Int

operator fun minus(other: Int): Int

operator fun minus(other: Long): Long

operator fun minus(other: Float): Float

operator fun minus(other: Double): Double

Вычитает другое значение из этого значения.

9.

operator fun plus(other: Byte): Int

operator fun plus(other: Short): Int

operator fun plus(other: Int): Int

operator fun plus(other: Long): Long

operator fun plus(other: Float): Float

operator fun plus(other: Double): Double

Добавляет другое значение к этому значению.

10.

operator fun rangeTo(other: Byte): [IntRange](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.ranges/-int-range/index.html)

operator fun rangeTo(other: Short): [IntRange](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.ranges/-int-range/index.html)

operator fun rangeTo(other: Int): [IntRange](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.ranges/-int-range/index.html)

operator fun rangeTo(other: Long): [LongRange](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.ranges/-long-range/index.html)

Создает диапазон от этого значения до указанного другого значения.

11.

operator fun rem(other: Byte): Int

operator fun rem(other: Short): Int

operator fun rem(other: Int): Int

operator fun rem(other: Long): Long

operator fun rem(other: Float): Float

operator fun rem(other: Double): Double

Вычисляет остаток от усечения от деления этого значения на другое значение. Результат либо равен нулю, либо имеет тот же знак, что и делимое, и имеет абсолютное значение меньше абсолютного значения делителя.