**Отчет**

**Наследники класса List**

* [MutableList](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-mutable-list/index.html)

interface MutableList<E> : List<E>, MutableCollection<E>

Универсальная колекция элементов с реализацией добавления и удаления элементов.

* [AbstractList](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-abstract-list/index.html)

abstract class AbstractList<out E> :   
    [AbstractCollection](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-abstract-collection/index.html)<E>,  
    [List](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-list/index.html)<E>

Скелетная реализация интерфейса списка только для чтения.

**Методы класса List, используемые в программе:**

* [plus](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/plus.html)

operator fun <T> [Iterable](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-iterable/index.html)<T>.plus(**element**: T): [List](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-list/index.html)<T>

operator fun <T> [Collection](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-collection/index.html)<T>.plus(**element**: T): [List](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-list/index.html)<T>

Возвращает список, содержащий все элементы исходной коллекции, а затем данный элемент.

* drop

Возвращает список, содержащий все элементы, кроме первых [n](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/drop.html#kotlin.collections$drop(kotlin.collections.Iterable((kotlin.collections.drop.T)),%20kotlin.Int)/n) элементов.

fun <T> [Iterable](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-iterable/index.html)<T>.drop(**n**: Int): [List](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-list/index.html)<T>

* dropLast

Возвращает список, содержащий все элементы, кроме последних [n](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/drop-last.html#kotlin.collections$dropLast(kotlin.collections.List((kotlin.collections.dropLast.T)),%20kotlin.Int)/n) элементов.

fun <T> [List](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-list/index.html)<T>.dropLast(**n**: Int): [List](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-list/index.html)<T>

* [indexOf](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-list/index-of.html)

Возвращает индекс первого вхождения указанного элемента в список или -1, если указанный элемент не содержится в списке.

abstract fun indexOf(**element**: E): Int

* contains

Возвращает true, если элемент найден в коллекции.

operator fun <T> [Iterable](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-iterable/index.html)<T>.contains(**element**: T): Boolean

* [binarySearch](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/binary-search.html)

Выполняет поиск в этом списке или его диапазоне предоставленного элемента с помощью алгоритма двоичного поиска. Предполагается, что список будет отсортирован по возрастанию в соответствии с сопоставимым естественным порядком его элементов, в противном случае результат не определен.

fun <T : [Comparable](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-comparable/index.html)<T>> [List](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-list/index.html)<T?>.binarySearch(  
    **element**: T?,  
    **fromIndex**: Int = 0,  
    **toIndex**: Int = size  
): Int

## Класс BufferedReader

Считывает текст из потока ввода символов, буферизуя символы таким образом, чтобы обеспечить эффективное чтение символов, массивов и строк.

Размер буфера может быть задан или использован размер по умолчанию. Значение по умолчанию достаточно велико для большинства целей.

[java.lang.Object](https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html)

[java.io.Reader](https://docs.oracle.com/javase/8/docs/api/java/io/Reader.html)

java.io.BufferedReader

* bufferedReader

fun [File](https://docs.oracle.com/javase/8/docs/api/java/io/File.html).bufferedReader(  
    **charset**: [Charset](https://docs.oracle.com/javase/8/docs/api/java/nio/charset/Charset.html) = Charsets.UTF\_8,  
    **bufferSize**: Int = DEFAULT\_BUFFER\_SIZE  
): [BufferedReader](https://docs.oracle.com/javase/8/docs/api/java/io/BufferedReader.html)

Возвращает новый BufferedReader для чтения содержимого этого файла.

### Параметры

bufferSize - необходимый размер буфера.

**Класс String**

class String : [Comparable](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-comparable/index.html)<String>, [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html)

Класс представляет символьные строки. Все строковые литералы в программах Kotlin, такие как "abc", реализуются как экземпляры этого класса.

* subSequence

Возвращает новую последовательность символов, которая является подпоследовательностью этой последовательности символов, начиная с указанной [startIndex](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/sub-sequence.html" \l "kotlin.CharSequence$subSequence(kotlin.Int,%20kotlin.Int)/startIndex) и заканчивается прямо перед указанным [endIndex](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/sub-sequence.html" \l "kotlin.CharSequence$subSequence(kotlin.Int,%20kotlin.Int)/endIndex).

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

**Интерфейс Iterator**

interface Iterator<out T>

Итератор над коллекцией или другой сущностью, которая может быть представлена в виде последовательности элементов. Позволяет последовательно обращаться к элементам.

* [hasNext](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-iterator/has-next.html)

Возвращает true значение, если итерация содержит больше элементов.

abstract operator fun hasNext(): Boolean