**Методы класса Arrays**

Ну и на закуску рассмотрим методы класса Arrays, которые упрощают работу с массивами: они помогут вам копировать, сортировать массивы и так далее. Особо сейчас в них не вчитывайтесь, методы надо применять, а не читать! Сохраните себе их куда-нибудь недалеко, а лучше распечатайте и повесьте на стену, чтобы они были на виду.

Вот список наиболее используемых:

|  |  |  |
| --- | --- | --- |
| **Метод** | **Что делает** | **Пример** |
| *copyOf*(*array*, *length*) | Предназначен для копирования массива. В качестве аргументов принимает массив, который нужно скопировать, и длину нового массива. | //Создаем массив  **int**[] array = {**22**, **4**, **5**, **6**};  //Создаем копию  **int**[] copy = Arrays.copyOf(array, array.length); |
| *copyOfRange*(*array*, *from*, *to*) | Копирует часть массива, начиная с индекса from и заканчивая на индексе to. | //Создаем массив  **int**[] array = {**22**, **4**, **5**, **6**};  //Создаем копию, в отведенных рамках  **int**[] copy = Arrays.copyOfRange(array, **1**, **3**);  //copy будет равен {4, 5}. |
| *toString*(*array*) | Позволяет получить все элементы массива в виде одной строки. | //Создаем массив  **int**[] array = {**22**, **4**, **5**, **6**};  //Выводим массив в консоль  System.out.println(Arrays.toString(array));  //Будет напечатана строка: [22, 4, 5, 6] |
| *sort*(*array*) | Сортирует массив методом *quick* *sort.* | //Создаем массив  **int**[] array = {**22**, **4**, **5**, **6**};  //Сортируем его  Arrays.sort(array);  //Будет: [4, 5, 6, 22] |
| *binarySearch*(*array*, *element*) | Ищет элемент методом бинарного поиска и возвращает его индекс.  Подробнее о том, что такое [бинарный поиск](https://neerc.ifmo.ru/wiki/index.php?title=%D0%A6%D0%B5%D0%BB%D0%BE%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9_%D0%B4%D0%B2%D0%BE%D0%B8%D1%87%D0%BD%D1%8B%D0%B9_%D0%BF%D0%BE%D0%B8%D1%81%D0%BA). | //Создаем массив  **int**[] array = {**4**, **5**, **6**, **22**};  //Ищем элемент и печатаем его индекс в консоль  System.**out**.println(Arrays.binarySearch(array, **4**));  //Будет напечатано 0. |
| *fill*(*array*, *value*) | Заполняет массив переданным значением (удобно использовать, если нам необходимо значение по умолчанию для массива). | //Создаем массив  **int**[] array = **new** **int**[**10**];  //Заполняем все ячейки массива цифрой 15  Arrays.fill(array, **15**); |
| *equals*() | Проверяет, равны ли массивы, и возвращает *true* или *false*. | //Создаем массив 1  **int**[] array1 = {**1**, **2**, **4**, **56**};  //Создаем массив 2  **int**[] array2 = {**1**, **762**, **44**, **5**};  //Сравниваем массивы  Arrays.equals(array1, array2);  //Массивы не равны, метод вернёт false |

**Обратите внимание,** что указанные выше методы работают со всеми видами массивов.