

Лабораторная работа № 1

Изучение основных логических конструкций в Java, реализация алгоритмов сортировки чисел и строк, поиска чисел в отсортированном по возрастанию массиве

Подготовка к работе:

1. Освоить тему «Массивы в Java». Повторить синтаксис и основные правила создания массивов и работы с ними;
2. Повторить основные правила и синтаксис создания и использования конструкций циклов **for**, **foreach**, **while**, условного оператора **if**, операторов перехода **break** и **continue**;
3. Изучить алгоритмы сортировки массивов чисел «вставкой», «пузырьком», алгоритм бинарного поиска по отсортированному массиву.

Выполнение работы:

1. В среде IntelliJ IDEA создать проект Java, в котором реализовать ввод с консоли N произвольных целых чисел (тип **int**) и их запись в массив.

В Java для унификации взаимодействия с различными устройствами ввода/вывода существуют два абстрактных класса:

- **InputStream** (поток ввода).
- **OutputStream** (поток вывода).

При работе с консолью удобно использовать класс **BufferedReader**, который является наследником абстрактного класса **Reader**. В конструкторе **BufferedReader** получает ссылку на любого наследника **Reader** и сохраняет её себе в приватное поле (переменную). И затем в некоторых методах делегирует свои задачи данному объекту. Например:

```
BufferedReader reader =  
new BufferedReader(new InputStreamReader(System.in));
```

Здесь **InputStreamReader** это наследник **Reader** и получает в конструкторе поток ввода **InputStream**. В данном случае в классе **System** в поле **in** хранится ссылка типа **InputStream**. Такой **InputStream** уже открыт и готов к приёму входных данных. Обычно это данные, вводимые с клавиатуры. Например:

```
int [] arr = new int[20];  
for (int i = 0; i < arr.length; i++) {
```

```
    arr[i] = Integer.parseInt(reader.readLine());  
}
```

Здесь метод `.readLine()` считывает строку с клавиатуры (заканчивающуюся нажатием Enter), а метод `Integer.parseInt()` преобразовывает строку в целое число типа `int`.

2. Реализовать сортировку введённого массива целых чисел по возрастанию и вывод результирующего массива в консоль.

Для отображения содержимого элементов массива можно воспользоваться утилитарным классом `Arrays`, который позволяет корректно отображать различные типы содержимого массива. Например:

```
System.out.println(Arrays.toString(array));
```

3. Доработать проект, реализовав алгоритм сортировки массива чисел «вставкой» или «пузырьком» (в зависимости от номера бригады).
4. Реализовать алгоритм поиска в массиве произвольного числа, введённого с клавиатуры. Если такое число существует и найдено, то вывести индекс ячейки в массиве, где хранится это число.
5. Реализовать алгоритм бинарного поиска для нахождения в массиве произвольного числа, введённого с клавиатуры.
6. Доработать проект таким образом, чтобы он обеспечивал ввод с консоли `N` произвольных строк (слов или выражений), запись их в массив и сортировку в алфавитном порядке.

При решении данной задачи можно воспользоваться тем, что класс `String` уже реализует интерфейс `Comparable`, который используется для сравнения объектов. И в классе `String` метод сравнения `int compareTo(String s)` как раз реализован таким образом, чтобы сравнивать строки в алфавитном порядке.