## День 8

Темы:

[Урок 23: StringBuilder](https://www.youtube.com/watch?v=Vw2GPl3APD4&list=PLAma_mKffTOSUkXp26rgdnC0PicnmnDak&index=23&ab_channel=alishev)

[Урок 23 (продолжение): Форматирование строк в Java](https://www.youtube.com/watch?v=WABydTyTlFs&list=PLAma_mKffTOSUkXp26rgdnC0PicnmnDak&index=24&ab_channel=alishev)

[Урок 24: Класс Object и метод toString()](https://www.youtube.com/watch?v=KEQ043yT3F4&list=PLAma_mKffTOSUkXp26rgdnC0PicnmnDak&index=25&ab_channel=alishev)

Доп.материалы - Как измерить время работы программы:

long startTime = System.currentTimeMillis(); //сохраняем время перед запуском блока кода

… // ваш код, время которого необходимо измерить

long stopTime = System.currentTimeMillis(); //сохраняем время после того, как код отработал

System.out.println(“Длительность работы, в мс.: “ + stopTime - startTime);

**Задачи:**

1. Создать строку, состоящую из чисел от 0 до 20000. Важно понимать, что это одна строка, полученная конкатенацией (“склеиванием”) чисел из диапазона через пробел (0 + “ “ + 1 + “ “ + 2 + … + 20000).

После создания такой строки, вызов System.out.println() на этой строке должен вывести в консоль сразу все числа из диапазона:

0 1 2 3 4 5 6 7 8 9 10 11 12 … 19995 19996 19997 19998 19999 20000

Для того, чтобы почувствовать разницу в производительности между конкатенацией обычных строк (класс String) и использовании StringBuilder, реализуйте описанную задачу этими двумя способами, замеряя время работы программы.

2. Скопируйте класс Самолет из задания дня 6.  
Внесите изменения в класс таким образом, чтобы следующий код выводил информацию о самолете, аналогично вызову метода info().

Airplane airplane = **new** Airplane ("Boeing", **2000**, **150**, **10000**);

System.out.println(airplane);