

# ЗАНЯТИЕ 14

---



# STRINGBUFFER И STRINGBUILDER

---

- Объекты `String` являются неизменяемыми, поэтому все операции, которые изменяют строки, фактически приводят к созданию новой строки, что сказывается на производительности приложения. Для решения этой проблемы, чтобы работа со строками проходила с меньшими издержками в Java были добавлены классы **`StringBuffer`** и **`StringBuilder`**. По сути они напоминают расширяемую строку, которую можно изменять без ущерба для производительности.

# STRINGBUFFER И STRINGBUILDER

---

- Эти классы похожи, практически двойники, они имеют одинаковые конструкторы, одни и те же методы, которые одинаково используются. Единственное их различие состоит в том, что класс `StringBuffer` синхронизированный и потокобезопасный. То есть класс `StringBuffer` удобнее использовать в многопоточных приложениях, где объект данного класса может меняться в различных потоках. Если же речь о многопоточных приложениях не идет, то лучше использовать класс `StringBuilder`, который **не потокобезопасный**, но при этом работает быстрее, чем `StringBuffer` в однопоточных приложениях.

# STRINGBUFFER И STRINGBUILDER

---

- <https://metanit.com/java/tutorial/7.3.php>
- StringBBMain.java

# ПРАКТИКА (ВЕЗДЕ ИСПОЛЬЗОВАТЬ МЕТОДЫ)

---

- 1.1) пользователь вводит текст. посчитать количество гласных,
- 1.2) согласных
- 2) Посчитать количество слов в тексте
- 3) Есть массив строк, вывести на экран все строки, длина которых больше N символов
- 4) Задание на изучение нового материала: вывести текущую дату, день недели и время. Подсказка кроется в Java 8 :)
- 5) Сравнить строки лексикографически
- 6) Есть массив слов. Нужно отфильтровать его так, чтобы в выходном массиве оказались слова, которые начинаются с буквы “W”/”w”. Было: {"width", "ellipse", "Wrong", "hyperbola», стало: [width, Wrong]
- 7) Есть текст, пользователь вводит несколько слов. Надо проверить, есть ли они в тексте и в каком количестве
- 8) Есть строка “Green. red. blue. yellow”. Необходимо заменить все точки на запяты