

# Строки





TEL-RAN  
by Starta Institute

1

# ПОВТОРЕНИЕ ИЗУЧЕННОГО

# Повторение

- Поиск
  - Линейный поиск
- Сортировка
  - Сортировка пузырьком



2

# ВОПРОСЫ ПО ПОВТОРЕНИЮ

# Введение

- Строки
- StringBuilder
- StringBuffer

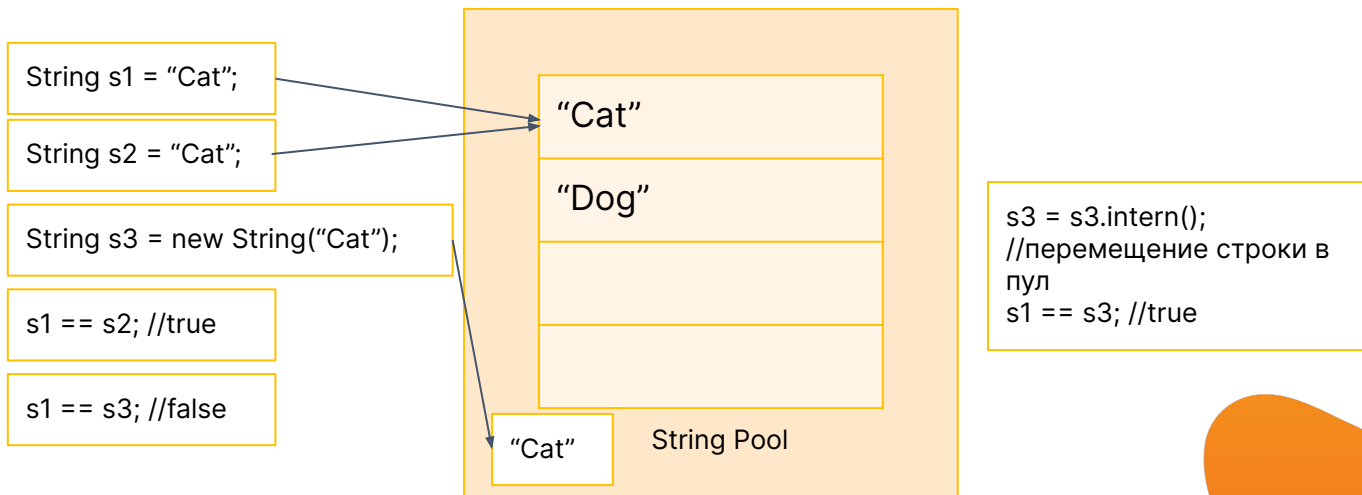


3

# ОСНОВНОЙ БЛОК

# Строки

- Java использует концепцию строкового литерала.
  - более эффективно с точки зрения памяти
  - новые объекты не создаются, если они уже существуют в пуле строковых констант.
- `String s = new String("Hello!");`
  - новый строковый объект в обычной памяти "куче" (не в пуле).
- Пул строк (String Pool) - множество строк в куче (Java Heap Memory)



# Строки

- String - это последовательность символов. (array of chars)
- В Java объекты String являются неизменяемыми, что означает константу, которая не может быть изменена после создания.







TEL-RAN  
by Starta Institute

# 4

## ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

## ЗАДАНИЕ

1. Создайте строку как литерал: `str1 = "Java"`
2. Создайте строку как литерал: `str2 = "Java"`
3. Создайте строку как Объект: `str 3 = new ("Java")`
4. Создайте строку как Объект: `str 4 = new ("Java")`
5. Сравните строки 1 и 2 используя `==`
6. Сравните строки 1 и 2 используя `equals`
7. Сравните строки 3 и 4 используя `==`
8. Сравните строки 3 и 4 используя `equals`
9. Сравните строки 1 и 3 используя `==`
10. Сравните строки 1 и 3 используя `equals`
11. Проанализируйте результат

# Класс StringBuilder

- StringBuilder в Java представляет изменяемую последовательность символов.
- StringBuilder не предоставляет никаких гарантий синхронизации.
- Экземпляры StringBuilder небезопасны для использования несколькими потоками.



# Класс `StringBuilder`

Конструкторы:

<b><code>StringBuilder()</code></b>	Создает экземпляр без символов и начальной емкостью 16 символов.
<b><code>StringBuilder(int capacity)</code></b>	Создает экземпляр без символов и начальной емкости, заданной аргументом <code>capacity</code> .
<b><code>StringBuilder(CharSequence seq)</code></b>	Создает экземпляр, содержащий те же символы, что и указанный <code>CharSequence</code> .
<b><code>StringBuilder(String str)</code></b>	Создает экземпляр, инициализированный содержимым указанной строки.



# StringBuilder - методы

- **append()** - метод добавляет строковое представление к последовательности.
- **int capacity()** - метод возвращает текущую емкость.
- **char charAt(int index)** - метод возвращает значение char в этой последовательности по указанному индексу.
- **delete(int start, int end)** - метод удаляет символы в подстроке этой последовательности.
- **deleteCharAt(int index)** - метод удаляет символ в указанной позиции в этой последовательности.
- **int length()** - метод возвращает длину (количество символов).
- **replace(int start, int end, String str)** - метод заменяет символы в подстроке этой последовательности, символами в указанной строке String str.
- **reverse()** - метод приводит к замене этой последовательности символов обратной последовательностью.
- **substring()** - метод возвращает новый объект String.
- **toString()** - метод возвращает строку, представляющую данные в этой последовательности.



TEL-RAN  
by Starta Institute

5

# ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

## ЗАДАНИЕ

1. Создайте StringBuilder объекты используя все предоставленные конструкторы
2. Используя метод append добавьте к объекту, который создан при помощи конструктора без параметров, строку "StartaUniversity"
3. Распечатайте все объекты на консоле
4. Примените метод reverse() к любому созданному объекту
5. Распечатайте объект на консоле

Самостоятельно:

1. Создайте объект `StringBuilder`
2. Добавьте в объект строки
  - a. `Hello`
  - b. `" "` (пробел)
  - c. `world!`
3. Распечатайте результат
4. Вставьте строку `"beautiful "` между слов `Hello` и `world`
5. Распечатайте результат
6. Распечатайте полную емкость (capacity) объекта
7. Распечатайте размер объекта
8. Проанализируйте результат
9. Создайте массив `char` размера `StringBuilder`
10. Сохраните объект `StringBuilder` в массив, используйте метод `getChars()`
11. Распечатайте результат



# Класс StringBuffer

- StringBuffer в Java представляет изменяемую последовательность символов.
- StringBuffer может содержать символы и подстроки, вставленные в середину или добавленные в конец.
- StringBuffer **синхронизирован**.
- Экземпляры StringBuffer безопасны для использования несколькими потоками.



# Класс StringBuffer

Конструкторы:

<b>StringBuffer()</b>	Создает экземпляр без символов и начальной емкостью 16 символов.
<b>StringBuffer(int capacity)</b>	Создает экземпляр без символов и начальной емкости, заданной аргументом capacity.
<b>StringBuiler(String str)</b>	Создает экземпляр, инициализированный содержимым указанной строки.



# StringBuffer - методы

- **append()** - метод добавляет строковое представление к последовательности.
- **insert()** - метод вставляет строковое представление в эту последовательность.
- **replace(int start, int end, String str)** - метод заменяет символы в подстроке этой последовательности символами в указанной строке String.
- **int capacity()** - метод возвращает текущую емкость.
- **delete(int start, int end)** - метод удаляет символы в подстроке этой последовательности.
- **int indexOf()** - метод возвращает индекс в этой строке первого вхождения указанной подстроки.
- **int length()** - метод возвращает длину (количество символов).
- **reverse()** - метод приводит к замене этой последовательности символов обратной последовательностью.
- **substring()** - метод возвращает новый объект String, содержащий подпоследовательность символов, содержащихся в данный момент в этой последовательности символов.
- **toString()** - метод возвращает строку, представляющую данные в этой последовательности.



TEL-RAN  
by Starta Institute

# 6

## ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

## ЗАДАНИЕ

1. Создайте объект StringBuffer используя конструктор String
  - а. "It is String Builder Object"
2. Распечатайте объект
3. Замените слово Builder на слово Buffer
  - а. Используйте метод replace()
4. Распечатайте результат

# StringBuilder VS StringBuffer

- StringBuilder и StringBuffer - отличные возможности Java по обработке строк.
- Классы идентичны, за исключением одного важного отличия: StringBuilder не синхронизирован, что означает, что он не является потокобезопасным.
- Преимуществом StringBuilder является более высокая производительность.
- В случаях, когда используется многопоточность, необходимо использовать StringBuffer, а не StringBuilder.



# Экспресс-опрос

- **Вопрос 1.**

Вы создали строку `String srt = new String("Java");` Можете ли вы изменить эту строку в дальнейшем коде?

- **Вопрос 2.**

Верно ли утверждение, что строка в памяти храниться как массив символов?



7

# ВОПРОСЫ ПО ОСНОВНОМУ БЛОКУ





TEL-RAN  
by Starta Institute

# 8

# ПРАКТИЧЕСКАЯ РАБОТА

# Практическое задание 1

1. Напишите метод, который принимает две строки в аргументах, соединяет их и определяет реверсивный порядок, выведите результат на консоль.
  - a. `concatReversStr(String str1, String str2)`
2. Дан массив символов, объедините его и замените символ '?' на правильный символ, выведите результат на консоль.
  - a. `char[] arr = {'l', ' ', 'l', 'i', 'k', '?', ' ', 'j', 'a', 'v', 'a'};`



# Реализация задания 1

```
public static void main(String[] args) {  
    char[] arr = {'I', ' ', 'l', 'i', 'k', '?', ' ', 'J', 'a', 'v', 'a'};  
    StringBuilder sb = new StringBuilder();  
    sb.append(arr, offset: 0, arr.length);  
    System.out.println(sb);  
    sb.replace( start: 5, end: 6, str: "e");  
    System.out.println(sb);  
  
    System.out.println(concatReversStr("mama", "papa"));  
}
```

# Практическое задание 2

1. Изменить порядок слов в данной строке на обратный
  - a. предположим, что строка не содержит знаки препинания, но содержит пробелы
  - b. Используйте алгоритм Линейный поиск для разделения слов
  - c. Используйте StringBuilder в решении

Например: "мама папа" → "папа мама"



# Реализация задания 2

```
for (int j = 0; j <= str.length(); j++) {  
    if (j == str.length()) {  
        countWorld = countWorld - 1;  
        worlds[countWorld] = stringBuilder.toString();  
        stringBuilder = new StringBuilder();  
    } else if (str.charAt(j) != ' ') {  
        stringBuilder.append(str.charAt(j));  
    } else {  
        countWorld = countWorld - 1;  
        worlds[countWorld] = stringBuilder.toString();  
        stringBuilder = new StringBuilder();  
    }  
}
```



TEL-RAN  
by Starta Institute

9

# ОСТАВШИЕСЯ ВОПРОСЫ

# Домашнее задание

№1

Дан массив размера  $n-1$ , содержащий только различные целые числа в диапазоне от 1 до  $n$ . Найдите недостающий элемент.

input:  
arr[] = {1,2,3,5}  
out: 4

arr[] = {6,1,2,8,3,4,7,10,5}  
out: 9

№2

Сформировать SQL запрос: Вернуть все записи из cars, где параметры равны заданным, используя StringBuilder.

Если значение null, то параметр не должен попадать в запрос.

{"model", "V-60", "country", "Germany", "city", "Berlin", "year", null, "active", true}

Пример: {"id", 1, "country", null, "city", "Helsinki", "year", null}

Результат: SELECT \* FROM users WHERE id = '1' AND city = 'Helsinki';

# Полезные ссылки

- [StringBuilder \(Java Platform SE 8 \) \(oracle.com\)](#)
- [StringBuilder \(Java SE 17 & JDK 17\) \(oracle.com\)](#)
- [String \(computer science\) - Wikipedia](#)



# Дополнительная практика

Создать программу, которая переставит числа в переменных таким образом, чтобы при выводе на экран последовательность а, b и с оказалась строго возрастающей.

Например:

Числа в переменных а, b и с: 7 0 -5

Возрастающая последовательность -5 0 7