

Обработка строк

Класс String

Класс StringBuffer

```
String aboutCat = "Cat - it sounds good!";
```

Класс String

Класс является неизменяемым — созданный объект класса String не может быть изменен.

Методы могут только создавать и возвращать новые строки, в которых хранится результат операции.

Преимущества:

- использование строк в многопоточных средах (String является потокобезопасным)

- использование String Pool (это коллекция ссылок на String объекты, используется для оптимизации памяти)

- использование строк в качестве ключей в HashMap (ключ рекомендуется делать неизменяемым)

Конструкторы

```
String s = new String();
```

```
char[] chars = {'c', 'a', 't'};
```

```
String str = new String(chars);
```

```
char[] chars = {'c', 'a', 't', 'a', 'm', 'a', 'r', 'a', 'n'};
```

```
String str = new String(chars, 0, 3);
```

Информация о строке

Длина

```
String habr = "habrahabr";
```

```
int length = habr.length();
```

Отдельный символ

```
String testString = "kitten";
```

```
char myChar = testString.charAt(2);
```

Сравнение строк

```
String testString, s;  
if (testString.equals(s)) {  
    System.out.println("Stroki ravny");  
} else {  
    System.out.println("Stroki ne ravny");  
}
```

```
if (testString.compareTo(s) == 0) {  
    System.out.println("Stroki ravny");  
} else {  
    System.out.println("Stroki ne ravny");  
}
```

Слияние строк

Операторы "+" и "+="

```
String habr = "habra" + "habr"; // "habrahabr"
```

```
habr += ".ru"; // "habrahabr.ru"
```

Метод `String concat(String str)`

```
String javaHub =
```

```
    "habrhabr".concat(".ru").concat("/hub").concat("/java");
```

```
// получим "habrhabr.ru/hub/java"
```

Форматирование

```
String formatString = "We are printing double variable (%f),  
string ('%s') and integer variable (%d).";
```

```
System.out.println(String.format(formatString, 2.3, "habr",  
10));
```

```
// We are printing double variable (2.300000), string ('habr')  
and integer variable (10).
```

Методы

Поиск подстроки, регулярного выражения.

Замещение подстроки, регулярного выражения.

Выделение подстроки

Преобразование строки в верхний или нижний регистр

Регулярные выражения

Выражения, описывающие структуру текстовой строки с использованием обычных символов и метасимволов, определяющих свойства строки в целом или её отдельных частей.

Пример:

```
String regexp = "([01]?[0-9]|2[0-3]):[0-5][0-9]";
```

Подходящие строки: "02:59" "2:45" "13:08" "23:15"

Не подходящие строки: "56:45" "123:56" "9:65" "5 12"

Основные метасимволы (1)

- ^ — (крышка, циркумфлекс) начало проверяемой строки
- \$ — (доллар) конец проверяемой строки
- .
- (точка) представляет собой сокращенную форму записи для символьного класса, совпадающего с любым символом
- | — означает «или». Подвыражения, объединенные этим способом, называются альтернативами
- ? — (знак вопроса) означает, что предшествующий ему символ является необязательным
- + — обозначает «один или несколько экземпляров непосредственно предшествующего элемента
- * — любое количество экземпляров элемента (в том числе и нулевое)
- {n} — Ровно n раз
- {m,n} — От m до n включительно
- {m,} — Не менее m
- {,n} — Не более n

Основные метасимволы (2)

[] — один из символов, входящих в заданный набор

[^] — один из символов, не входящих в заданный набор

- — (дефис) интервал символов;

() — выделение группы

\\d — цифровой символ

\\D — не цифровой символ

\\s — пробельный символ

\\S — не пробельный символ

\\w — буквенный или цифровой символ или знак подчёркивания

\\W — любой символ, кроме буквенного или цифрового символа или знака подчёркивания

Примеры

ab — строка «ab»

a.b — строка из трех символов, первый a, второй — любой, третий b

a* — строка из любого количества букв a

[abc] — a, b, или c (один символ)

[^abc] — любой символ, кроме a, b, или c

[a-zA-Z]+ — непустая последовательность латинских букв в любом порядке

\\+ — символ +

Класс StringBuffer

Создание объекта:

```
StringBuffer sb = new StringBuffer();
```

```
StringBuffer strb = new StringBuffer("Text");
```

```
String s = "Example";
```

```
StringBuffer ex = new StringBuffer(s);
```

Методы

StringBuffer append(String str)

char charAt(int index)

void setCharAt(int index, char ch)

StringBuffer delete(int start, int end)

StringBuffer insert(int offset, String str)

StringBuffer replace(int start, int end, String str)

StringBuffer reverse()

String substring(int start, int end)

String toString()