

# Java advanced

Работа с текстом



## Работа с текстом

# Класс String

java.lang.String

`java.lang.String` – класс, используемый для работы со строками. Каждый экземпляр строки представляет собой набор символов. Строки являются неизменяемыми типами (immutable), однажды созданный объект не может быть изменён.

# Способы создания строк

## String create

1. Путем присвоения строкового литерала переменной типа String.
2. Путем непосредственного вызова конструктора String().
3. С помощью оператора сцепления строк (+) для создания одной строки из любой комбинации экземпляров класса String и строковых литералов.
4. Путем извлечения свойства или вызова метода, который возвращает строку.
5. Путем вызова метода форматирования для преобразования значения или объекта в строковое представление.

# Методы класса String

## String methods

<code>t.charAt(int index)</code>	Возвращает значение типа <code>char</code> по определённому индексу.
<code>t.concat(String s)</code>	Добавляет строку <code>s</code> в конец строки <code>t</code> .
<code>t.compareTo(String s)</code>	Лексикографически сравнивает строки <code>t</code> и <code>s</code> .
<code>t.equals(Object o)</code>	Сравнивает строку <code>t</code> с объектом <code>o</code> .
<code>t.format(String f, Object... args)</code>	Возвращает отформатированную заданный строкой <code>f</code> формат и аргументы <code>args</code> .
<code>t.isEmpty()</code>	Возвращает <code>true</code> если <code>t.length() == 0</code>
<code>t.length()</code>	Возвращает длину строки.
<code>t.matches(String reg)</code>	Проверяет соответствие строки <code>t</code> регулярному выражению <code>reg</code> .
<code>t.substring(int begin)</code>	Создаёт новую строку начиная с позиции <code>begin</code> строки <code>t</code> .

# Класс `StringBuilder`

## `StringBuilder`

Экземпляры класса `StringBuilder` являются изменяемы, что позволяет эффективнее работать с изменяемыми строками.

Метод `append()` добавляет символьную последовательность в конец строки.

Метод `insert()` добавляет символьную последовательность в указанное место строки.

Экземпляры `StringBuilder` не являются потокобезопасными. В мультипоточных приложениях рекомендуется использовать `StringBuffer`, методы которого синхронизированы.

# Пакет java.time

java.time

`java.time` – основной API для даты, времени, отрезков времени и моментов времени.

<code>Clock</code>	Обеспечивает доступ к текущей дате и времени используя часовой пояс
<code>LocalDateTime</code>	Дата и время без учета временной зоны. (2015-12-21T21:50:50)
<code>LocalDate</code>	Дата без учета временной зоны. (2015-12-21)
<code>LocalTime</code>	Время без учета временной зоны. (21:50:50)
<code>ZonedDateTime</code>	Дата и время с временной зоной. (2015-12-21T21:50:50+01:00 Europe/Paris)
<code>OffsetDateTime</code>	Дата и время со смещением по Гринвичу (2015-12-21T21:50:50+01:00)
<code>Duration</code>	Период времени (400 sec). В секундах или наносекундах.
<code>Period</code>	Отрезок времени (2 years 5 months). В годах, месяцах и днях.

# Регулярные выражения

## Regular Expressions

**Регулярные выражения** - это формальный язык поиска и осуществления манипуляций с подстроками в тексте, основанный на использовании метасимволов.

Классы для работы с регулярными выражениями находятся в пакете `java.util.regex`.

Экземпляр класса `Pattern` содержит в себе скомпилированное представление регулярного выражения.

Класс `Matcher` содержит механизм для операций сравнения входящей последовательности с шаблоном, хранящемся в экземпляре `Pattern`.



# Метасимволы

## Metacharacters

Символ	Описание
<b>\b</b>	Позиция, соответствующая границе слова
<b>\B</b>	Позиция, не соответствующая границе слова
<b>\n</b>	Соответствует символу новой строки
<b>\r</b>	Соответствует символу возврата каретки
<b>\t</b>	Соответствует символу табуляции
<b>\f</b>	Соответствует символу конца файла
<b>\d</b>	Соответствует любой десятичной цифре
<b>\D</b>	Соответствует любому символу, кроме десятичной цифры
<b>\w</b>	Соответствует любому алфавитно-цифровому символу и символу подчеркивания, т.е. символ, образующий "слово"
<b>\W</b>	Соответствует всем символам, которые не попадают под определение метасимвола \w
<b>\s</b>	Соответствует любому пробельному символу
<b>\S</b>	Соответствует любому не пробельному символу

# Квантификаторы

## Quantifiers

Квантификатор	Описание
<code>?</code>	Предшествующий символ либовходит в строку один раз, либо вообще в нее не входит
<code>*</code>	Предшествующий символ входит в строку любое число раз, в том числе и 0
<code>+</code>	Предшествующий символ входит в строку один и более число раз
<code>{n}</code>	Предшествующий символ входит в строку n раз
<code>{n, }</code>	Предшествующий символ входит в строку n и более количество раз
<code>{n, m}</code>	Предшествующий символ входит в строку от n до m раз

# Специальные символы

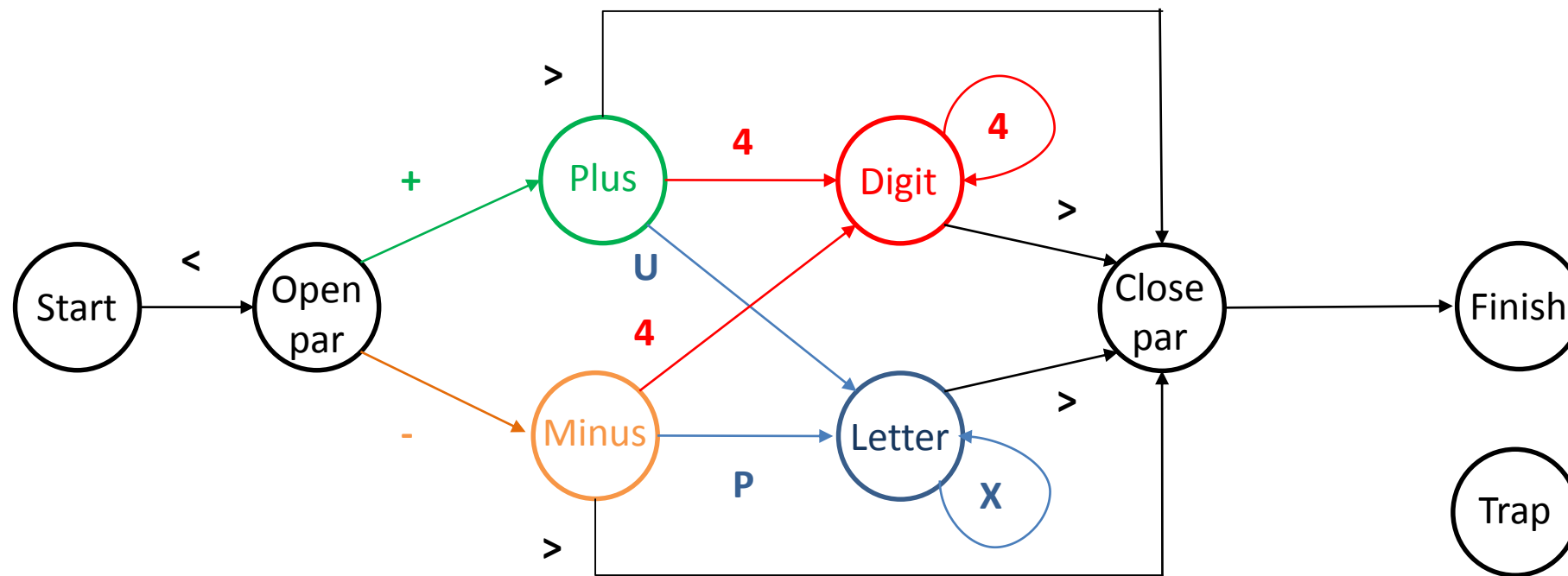
## Special characters

Метасимвол	Описание
.	Соответствует одному любому символу
[...]	Соответствует одному символу из тех, что перечислены в квадратных скобках
[^...]	Соответствует одному любому символу, не перечисленному в квадратных скобках
^	Позиция в начале строки
\$	Позиция в конце строки
	Любое из разделяемых выражений
(...)	Круглые скобки служат для логического объединения частей регулярного выражения

# Конечный автомат

## Finite State Machine

$\langle (\backslash + | -) (([0-5])^* | ([P-Z])^*) \rangle$



# Проверка знаний

TestProvider.com



TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на [TestProvider.com](https://testprovider.com)

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.



Q&A