

Работа с текстом

№ урока: 3 Курс: JAVA Advanced

Средства обучения: Компьютер с установленной IntelliJ IDEA.

Обзор, цель и назначение урока

На уроке рассматривается работа с текстом и регулярными выражениями:

- Класс String, его методы.
- Класс StringBuilder.
- Класс LocalDateTime.
- Класс DateTimeFormatter.
- Регулярные выражения.
- Принцип работы регулярных выражений на примере конечного автомата (Finite State Machine).

Изучив материал данного занятия, учащийся сможет:

- Использовать методы классов String и StringBuilder для работы со строками.
- Использовать классы пакета java.time для работы со временем и датой.
- Создавать и использовать регулярные выражения.
- Понимать механизм работы регулярных выражений на примере конечного автомата.

Содержание урока

1. Методы класса String. Объекты класса String.
2. Конкатенация и сравнение строк.
3. Отличия между String, StringBuffer и StringBuilder.
4. Форматирование строк. Локализация.
5. Работа с датой и временем, пакет java.time.
6. Форматирование даты и времени.
7. Регулярные выражения.
8. Конечный автомат.

Резюме

- Класс String представляет символьные строки. Все строчные литералы в Java-программах реализованы как объекты класса String.
- Объекты класса String являются immutable (неизменяемыми) и не могут быть изменены после создания. Для корректного сравнения объектов класса String необходимо использовать метод equals().
- Для конкатенации строк рекомендуется использовать классы StringBuilder и StringBuffer. Объекты классов StringBuilder и StringBuffer являются mutable и позволяют изменять своё состояние. Класс StringBuilder идентичен классу StringBuffer, за исключением того что он не является синхронизированным и работает быстрее.
- Форматирование строк с помощью метода String.format() аналогично использованию класса java.util.Formatter. Для локализации форматирования первым аргументом следует указать локаль, например Locale.ENGLISH.
- Пакет java.time представляет разработчику широкий функционал для работы с датой и временем.
- Класс LocalDateTime позволяет создавать объекты, хранящие в себе дату и время без указания временной зоны, например 2015-12-31T13:59:59.

- Объекты класса `LocalDate` хранят в себе только дату (2015-12-31), а объекты класса `LocalTime` – время (13:59:59).
- Объекты классов `LocalDateTime`, `LocalTime` и `LocalDate` являются `immutable`. Чтобы передать только время или дату из объекта `LocalDateTime` необходимо использовать методы `toLocalTime()` или `toLocalDate()` соответственно.
- Чтобы отформатировать вывод даты и времени по заданным условиям, необходимо использовать экземпляр `DateTimeFormatter`. Экземпляр `DateTimeFormatter` представляет собой шаблон, согласно которому будет отформатирован вывод экземпляра `LocalDateTime`.
- Регулярные выражения (англ. *regular expressions*) — формальный язык поиска и осуществления манипуляций с подстроками в тексте, основанный на использовании метасимволов. По сути это строка-образец (англ. *pattern*, по-русски её часто называют «шаблоном», «маской»), состоящая из символов и метасимволов и задающая правило поиска.
- Для использования регулярных выражений используется пакет `java.util.regex`, который содержит классы `Pattern` и `Matcher`.
- Регулярное выражение в виде строки должно быть скомпилировано в экземпляр класса `Pattern`. Полученный шаблон используется для сравнения последовательности символов классом `Matcher`.
- В классе `Matcher` содержится механизм, который производит операции соответствия на последовательности символов, интерпретируя шаблон `Pattern`.
- Конечный автомат (Finite State Machine) – абстрактный автомат, число возможных внутренних состояний которого конечно.
- Принято полагать, что конечный автомат начинает работу в состоянии q_0 , последовательно считывая по одному символу входного слова (цепочки входных символов). Считанный символ переводит автомат в новое состояние, в соответствии с функцией переходов.
- Читая входную цепочку символов x и делая переходы из состояния в состояние, автомат, после прочтения последнего символа входного слова окажется в некотором состоянии q' . Если это состояние является заключительным, то говорят, что автомат допустил слово x .

Закрепление материала

- Каким способом необходимо производить конкатенацию строк? Почему?
- Какими способами можно отформатировать вывод строк?
- Как локализовать вывод?
- Какой пакет рекомендуется использовать для работы с датой и временем?
- Как использовать регулярные выражения в Java?
- Что такое конечный автомат?
- Объясните механизм работы регулярных выражений на примере конечного автомата.

Дополнительное задание

Напишите консольное приложение, позволяющее пользователю зарегистрироваться под «Логин», состоящем только из символов латинского алфавита, и пароля, состоящего из цифр и символов.

Самостоятельная деятельность учащегося

Задание 1

Выучите основные конструкции и понятия, рассмотренные на уроке.

Задание 2

Напишите программу, которая бы позволила вам по указанному адресу web-страницы выбирать все ссылки на другие страницы, номера телефонов, почтовые адреса и сохраняла полученный результат в файл.

Задание 3

Напишите шуточную программу «Дешифратор», которая бы в текстовом файле могла бы заменить все предлоги на слово «^%#@!».

Задание 4

Создайте текстовый файл-чек по типу «Наименование товара – 0.00(цена)грн.» с определенным количеством наименований товаров и датой совершения покупки. Выведите на экран информацию из чека в формате текущей локали пользователя и в формате локали en-US.

Рекомендуемые ресурсы

Класс String

<http://docs.oracle.com/javase/7/docs/api/java/lang/String.html>

Класс StringBuilder

<http://docs.oracle.com/javase/7/docs/api/java/lang/StringBuilder.html>

Класс Formatter

<http://docs.oracle.com/javase/7/docs/api/java/util/Formatter.html>

Класс Locale

<http://docs.oracle.com/javase/7/docs/api/java/util/Locale.html>

Пакет java.time

<https://docs.oracle.com/javase/8/docs/api/java/time/package-summary.html>

Регулярные выражения Wiki

https://ru.wikipedia.org/wiki/Регулярные_выражения

Синтаксис регулярных выражений

http://regexstudio.com/ru/TRegExpr/Help/regex_syntax.html

Пакет java.util.regex

<http://docs.oracle.com/javase/7/docs/api/index.html?java/util/regex/Pattern.html>

Конечный автомат

https://ru.wikipedia.org/wiki/Конечный_автомат

Проверка регулярного выражения

<https://regex101.com/>