Форматирование и анализ даты в Java

В Java имеется удобный класс SimpleDateFormat, который можно использовать для форматирования и анализа даты. В этой статье Вы найдете обзор парметров, принимаемых методами класса SimpleDateFormat, а также будут приведены результаты выполнения метода format с разнообразными параметрами.

Приведем простой пример форматирования даты с использованием класса SimpleDateFormat:

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.Locale;

*/\*\**

*\**

*\* @author seostella.com*

*\*/*

public class SimpleDateFormatExample {

public static void main(String[] args) {

SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy.MM.dd G 'at' HH:mm:ss z");

System.out.println("date: " + dateFormat.format( new Date() ) );

}

}

Если на компьютере локализацией по умолчанию является русская локализация, то в результате выполнения приведенного кода Вы увидите следующее:

date: 2012.02.07 н.э. at 15:13:08 EET

Рассмотрим некоторые методы класса SimpleDateFormat. Для начала, у класса есть 4 конструктора:

SimpleDateFormat()

*// Создает SimpleDateFormat, используя паттерн времени и формат символов по умолчанию для текущей локализации*

SimpleDateFormat(String pattern)

*// Создает SimpleDateFormat, используя заданный паттерн времени и формат символов по умолчанию для текущей локализации*

SimpleDateFormat(String pattern, DateFormatSymbols formatSymbols)

*// Создает SimpleDateFormat, используя заданные паттерн времени и формат символов*

SimpleDateFormat(String pattern, Locale locale)

*// Создает SimpleDateFormat, используя заданный паттерн времени и формат символов по умолчанию для заданной локализации*

Следующий пример демонстрирует работу всех четырех конструкторов:

package simpledateformatexample;

import java.text.DateFormatSymbols;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.Locale;

*/\*\**

*\**

*\* @author seostella.com*

*\*/*

public class SimpleDateFormatExample {

public static void main(String[] args) {

Date currentDate = new Date();

SimpleDateFormat dateFormat = null;

dateFormat = new SimpleDateFormat();

System.out.println("Constructor 1: " + dateFormat.format( currentDate ) );

dateFormat = new SimpleDateFormat("dd MMMM");

System.out.println("Constructor 2: " + dateFormat.format( currentDate ) );

dateFormat = new SimpleDateFormat("dd MMMM", myDateFormatSymbols );

System.out.println("Constructor 3: " + dateFormat.format( currentDate ) );

dateFormat = new SimpleDateFormat("dd MMMM", Locale.ENGLISH);

System.out.println("Constructor 4: " + dateFormat.format( currentDate ) );

}

private static DateFormatSymbols myDateFormatSymbols = new DateFormatSymbols(){

@Override

public String[] getMonths() {

return new String[]{"января", "февраля", "марта", "апреля", "мая", "июня",

"июля", "августа", "сентября", "октября", "ноября", "декабря"};

}

};

}

Результатом выполнения этого кода будут следующие четыре строки:

Constructor 1: 07.02.12 15:34

Constructor 2: 07 Февраль

Constructor 3: 07 февраля

Constructor 4: 07 February

**Конструктор по умолчанию** использует паттерн времени и формат символов по умолчанию для текущей локализации. То есть, для русской локализации стандартным паттерном времени является паттерн "dd.MM.yy HH:mm".

**Конструктор SimpleDateFormat(String pattern)** принимает паттерн даты, в котором будет отдавать результат метод format(). В примере мы использовали паттерн "dd MMMM", который требует от format() вывести дату в следующем формате: "две цифры дня месяца" + "пробел" + "название месяца". В данном случае используется название месяца по умолчанию для текущей локализации, т.е., "Февраль".

**Конструктор SimpleDateFormat("dd MMMM", myDateFormatSymbols )** аналогичен предыдущему за исключением того, что название месяца используется не по умолчанию, а те, которые возвращает переменная myDateFormatSymbols. В свою очередь, в переменной myDateFormatSymbols мы переопределили метод getMonths() чтобы он возвращал названия месяцев с прописной буквы и в родительном падеже.

**Конструктор SimpleDateFormat("dd MMMM", Locale.ENGLISH)** аналогичен конструктору SimpleDateFormat(String pattern), но использует заданную локализацию. В нашем случае это английская локализация Locale.ENGLISH.

Рассмотрим подробно параметры, принимаемые классом SimpleDateFormat в качестве паттерна даты.

|  |  |  |
| --- | --- | --- |
| **Символ** | **Что означает** | **Пример** |
| G | эра (в английской локализации - AD и BC) | н.э. |
| y | год (4-х значное число) | 2012 |
| yy | год (последние 2 цифры) | 12 |
| yyyy | год (4-х значное число) | 2012 |
| M | номер месяца без лидирующих нулей | 2 |
| MM | номер месяца (с лидирующими нулями если номер месяца < 10) | 02 |
| MMM | четырех буквенное сокращение месяца в русской локализации и трех буквенное - в английской (Feb) | фев |
| MMMM | полное название месяца (в английской локализации - February) | Февраль |
| w | неделя в году без лидирующих нулей | 7 |
| ww | неделя в году с лидирующими нулями | 07 |
| W | неделя в месяце без лидирующих нулей | 2 |
| WW | неделя в месяце с лидирующим нулем (если это необходимо) | 02 |
| D | день в году | 38 |
| d | день месяца без лидирующих нулей | 7 |
| dd | день месяца с лидирующими нулями | 07 |
| F | день недели в месяце без лидирующих нулей | 1 |
| FF | день недели в месяце с лидирующими нулями | 01 |
| E | день недели (сокращение) | Вт |
| EEEE | день недели (полностью) | вторник |
| a | AM/PM указатель | AM |
| H | часы в 24-часовом формате без лидирующих нулей | 6 |
| HH | часы в 24-часовом формате с лидирующим нулем | 06 |
| k | количество часов в 24-часовом формате | 18 |
| K | количество часов в 12-часовом формате | 6 |
| h | время в 12-часовом формате без лидирующих нулей | 6 |
| hh | время в 12-часовом формате с лидирующим нулем | 06 |
| m | минуты без лидирующих нулей | 32 |
| mm | минуты с лидирующим нулем | 32 |
| s | секунды без лидирующих нулей | 11 |
| ss | секунды с лидирующим нулем | 11 |
| S | миллисекунды | 109 |
| z | часовой пояс | EET |
| Z | часовой пояс в формате RFC 822 | +0200 |
| ' | символ экранирования для текста | 'Date=' |
| '' | кавычка | 'o''clock' |

Рассмотрим несколько примеров паттернов даты и времени, которые представлены в официальной документации. Русская локализация:

|  |  |
| --- | --- |
| **Паттерн даты и времени** | **Результат** |
| "e;yyyy.MM.dd G 'at' HH:mm:ss z"e; | 2012.02.07 н.э. at 16:51:35 EET |
| "e;EEE, MMM d, ''yy"e; | Вт, фев 7, '12 |
| "e;h:mm a"e; | 4:51 PM |
| "e;hh 'o''clock' a, zzzz"e; | 04 o'clock PM, Eastern European Time |
| "e;K:mm a, z"e; | 4:51 PM, EET |
| "e;yyyyy.MMMMM.dd GGG hh:mm aaa"e; | 02012.Февраль.07 н.э. 04:51 PM |
| "e;EEE, d MMM yyyy HH:mm:ss Z"e; | Вт, 7 фев 2012 16:51:35 +0200 |
| "e;yyMMddHHmmssZ"e; | 120207165135+0200 |

Английская локализация:

|  |  |
| --- | --- |
| **Паттерн даты и времени** | **Результат** |
| "e;yyyy.MM.dd G 'at' HH:mm:ss z"e; | 2012.02.07 AD at 16:55:57 EET |
| "e;EEE, MMM d, ''yy"e; | Tue, Feb 7, '12 |
| "e;h:mm a"e; | 4:55 PM |
| "e;hh 'o''clock' a, zzzz"e; | 04 o'clock PM, Eastern European Time |
| "e;K:mm a, z"e; | 4:55 PM, EET |
| "e;yyyyy.MMMMM.dd GGG hh:mm aaa"e; | 02012.February.07 AD 04:55 PM |
| "e;EEE, d MMM yyyy HH:mm:ss Z"e; | Tue, 7 Feb 2012 16:55:57 +0200 |
| "e;yyMMddHHmmssZ"e; | 120207165557+0200 |