



# Язык программирования Java. Начальный уровень

## Мультимедийный курс

автор: Васильев А.Н.

[www.vasilev.kiev.ua](http://www.vasilev.kiev.ua)

Киев 2017



# Лекция 8. Дата и время



- Класс `Date`
- Класс `Calendar`
- Класс `SimpleDateFormat`
- Примеры использования даты и времени

*У нас в уезде писарь был. Год рождения в пачпорте одной циферкой записывал - чернила, шельма, экономил. Потом дело прояснилось - его в острог. А пачпорта уж переделывать не стали - документ всё-таки.*

*из к/ф "Формула любви"*

# Класс Date



Объект класса `Date` используется для представления определенного момента времени (с точностью до миллисекунд)

Для использования класса `Date` нужна инструкция `import java.util.Date`

## Основные конструкторы:

`Date ()`

Создается объект, который соответствует текущему (на момент создания объекта) моменту времени

`Date (long)`

Создается объект, который соответствует моменту времени (в миллисекундах), определяемому аргументом конструктора

В Java "отсчет времени" начинается с 1 января 1970 года (по Гринвичу)

Для получения текущего времени (в миллисекундах) может использоваться метод `System.currentTimeMillis ()`

# Методы класса Date



Метод	Описание
<code>boolean after(Date)</code>	Проверка на предмет того, что дата из объекта вызова следует после даты, указанной аргументом метода
<code>boolean before(Date)</code>	Проверка на предмет того, что дата из объекта вызова следует до, указанной аргументом метода
<code>int compareTo(Date)</code>	Сравнение двух дат (результат - целое число 0, 1 или -1)
<code>boolean equals(Object)</code>	Проверка на предмет равенства дат
<code>long getTime()</code>	Получение времени (в миллисекундах) для указанной даты (объект вызова)
<code>void setTime(long)</code>	Определение значения в миллисекундах для объекта даты
<code>String toString()</code>	Преобразование объекта в текстовый формат



# Программа: класс Date



```
import java.util.Date;
class DateMethDemo{
    public static void main(String args[]){
        // Текущая дата:
        Date start=new Date();
        System.out.println("Начало выполнения: "+start);
        long time;
        // Миллион секунд назад:
        time=start.getTime()-1000000000;
        Date t=new Date(time);
        System.out.println("Миллион секунд назад: " +t);
        // Время (почти) завершения программы:
        Date finish=new Date();
        System.out.println("Завершение выполнения: "+finish);
        // Длительность выполнения программы:
        System.out.println("Прошло "+(finish.getTime()-
start.getTime())+" миллисекунд");
    }
}
```

## Результат программы:

Начало выполнения: Thu Nov 24 10:58:10 EET 2016  
Миллион секунд назад: Sat Nov 12 21:11:30 EET 2016  
Завершение выполнения: Thu Nov 24 10:58:10 EET 2016  
Прошло 174 миллисекунд

© Васильев А.Н.

# Класс Calendar



Класс `Calendar` содержит методы для выполнения преобразований, связанных с формой представления даты и времени

При работе с классом `Calendar` используется инструкция `import java.util.Calendar`

Получение ссылки на объект:

```
Calendar cld=Calendar.getInstance();
```



Ссылка на объект, соответствующий текущему моменту времени

# Статические поля класса Calendar



Поле	Описание
DAY_OF_MONTH	День в месяце
DAY_OF_WEEK	День в неделе
DAY_OF_YEAR	День в году
HOUR_OF_DAY	Час дня
MILLISECOND	Миллисекунды в секунде
MINUTE	Минуты в часе
MONTH	Месяц
SECOND	Секунды в минуте
time	Время для данной даты в миллисекундах
WEEK_OF_MONTH	Неделя в месяце
WEEK_OF_YEAR	Неделя в году
YEAR	Год
ZONE_OFFSET	Временной сдвиг часовой зоны (в миллисекундах)

# Методы класса Calendar



Метод	Описание
<code>void add(int,int)</code>	Добавление значения к полю
<code>boolean after(Object)</code>	Проверка на предмет того, что дата из объекта вызова следует после даты, указанной аргументом метода
<code>boolean before(Object)</code>	Проверка на предмет того, что дата из объекта вызова следует до, указанной аргументом метода
<code>int compareTo(Calendar)</code>	Сравнение двух дат (результат - целое число 0, 1 или -1)
<code>boolean equals(Object)</code>	Проверка на предмет равенства дат
<code>int get(int)</code>	Получение значения поля, переданного аргументом методу
<code>Date getTime()</code>	Возвращается объект класса Date, соответствующий данной дате

# Методы класса Calendar



Метод	Описание
<code>long getTimeInMillis()</code>	Значение для времени (в миллисекундах) для данной даты
<code>int getWeeksInWeekYear()</code>	Количество недель в году
<code>int set(аргументы)</code>	Присваивание значений полям. Аргументы передаются разными способами: поле и значение; год, месяц, день; год, месяц, день, час, секунды; год, месяц, день, час, секунды и миллисекунды
<code>void setTime(Date)</code>	На основе объекта класса <code>Date</code> задается дата
<code>void setTimeInMillis(long)</code>	На основе значения в миллисекундах задается дата
<code>String toString()</code>	Преобразование объекта к текстовому формату

# Программа: класс Calendar



```
import java.util.Calendar;
class CalendarDemo{
    public static void main(String args[]){
        // Текущая дата:
        Calendar start=Calendar.getInstance();
        System.out.println("Начало выполнения: "+start.getTime());
        long time;
        // Миллион секунд назад:
        time=start.getTimeInMillis()-1000000000;
        Calendar t=Calendar.getInstance();
        t.setTimeInMillis(time);
        System.out.println("Миллион секунд назад: " +t.getTime());
        // Время (почти) завершения программы:
        Calendar finish=Calendar.getInstance();
        System.out.println("Завершение выполнения: "+finish.getTime());
        // Длительность выполнения программы:
        System.out.println("Прошло "+(finish.getTimeInMillis()-
start.getTimeInMillis())+" миллисекунд");
    }
}
```

## Результат программы:

```
Начало выполнения: Thu Nov 24 11:08:55 EET 2016
Миллион секунд назад: Sat Nov 12 21:22:15 EET 2016
Завершение выполнения: Thu Nov 24 11:08:55 EET 2016
Прошло 75 миллисекунд
```

# Программа: класс Calendar



```
import java.util.Calendar;
class UsingCalendarDemo{
    public static void main(String args[]){
        // Текущая дата:
        Calendar today=Calendar.getInstance();
        System.out.println("Сегодня "+today.getTime());
        System.out.println("День недели:\t"+today.get(Calendar.DAY_OF_WEEK));
        System.out.println("Дата:\t\t"+today.get(Calendar.DAY_OF_MONTH));
        System.out.println("Месяц:\t\t"+today.get(Calendar.MONTH));
        System.out.println("Год:\t\t"+today.get(Calendar.YEAR));
        System.out.println("Часы:\t\t"+today.get(Calendar.HOUR));
        System.out.println("Минуты:\t\t"+today.get(Calendar.MINUTE));
        System.out.println("Секунды:\t\t"+today.get(Calendar.SECOND));
    }
}
```

## Результат программы:

```
Сегодня Thu Nov 24 11:25:05 EET 2016
День недели:      5
Дата:             24
Месяц:            10
Год:              2016
Часы:             11
Минуты:           25
Секунды:          5
```



# Класс SimpleDateFormat



**Класс SimpleDateFormat предназначен для определения формы текстового представления даты и времени**

**Для использования класса SimpleDateFormat нужна инструкция**  
`import java.text.SimpleDateFormat`

## Основные конструкторы:

`SimpleDateFormat()`

**Создание объекта на основе используемого по умолчанию шаблона**

`SimpleDateFormat(String)`

**Создание объекта на основе шаблона который определяется строкой форматирования (аргумент конструктора)**

**Для формирования текстовой строки с датой/временем используется метод `format()`**



# Класс SimpleDateFormat: основные символы строки форматирования



Символ	Назначение
Y	Год
M	Месяц в году
w	Неделя в году
W	Неделя в месяце
D	День в году
d	День в месяце
F	День недели в году
E	Название дня в неделе
u	Номер дня в неделе
H	Час в сутках (0-23)
k	Час в дне (1-24)
K	Час в формате am/pm (0-11)
h	Час в формате am/pm (1-12)
m	Минуты в часе
s	Секунды в минуте
S	Миллисекунды

# Программа: класс Calendar



```
import java.text.SimpleDateFormat;
import java.util.Date;
class SimpleDateFormatDemo{
    public static void main(String[] args){
        long time=System.currentTimeMillis();
        Date today=new Date(time);
        SimpleDateFormat sdf=new SimpleDateFormat("ddd - MM - YYYY");
        System.out.println(sdf.format(today));
        sdf=new SimpleDateFormat("Год:\tYYYY\nМесяц:\tMM\nДень:\tdd");
        System.out.println(sdf.format(today));
    }
}
```

## Результат программы:

024 - 11 - 2016

Год: 2016

Месяц: 11

День: 24

# Домашнее задание



- Напишите программу, в которой определяется время и дата через 1 000 000 секунд после текущей даты и времени.
- Напишите программу, в которой на заданном интервале времени случайно выбирается дата и время, и для этой даты отображается день недели.
- Напишите программу, в которой дата выводится в формате, в котором сначала следует год, затем месяц, после этого - день месяца.