

Java. Дата и время

Для удобной работы с датой и временем в Java используются классы `Calendar` и `Date`. Оба класса находятся в библиотеке `java.util`. Начнем знакомство с датой. Класс `Date` хранит время в миллисекундах начиная с 1 января 1970 года. Данный класс имеет конструктор по умолчанию, который возвращает текущее время. Кроме этого можно создать объект `Date` используя конструктор, который принимает количество миллисекунд начиная с 1 января 1970 года. Для получения этого внутреннего времени используется метод `getTime()`. Кроме этого уже после создания можно изменить время с помощью `setTime(long date)`. Для того, чтобы отображать дату и время в удобном для вас формате используется `SimpleDateFormat`:

```
1 package ru.cybern;
2
3 import java.text.SimpleDateFormat;
4 import java.util.Date;
5 public class Test {
6     public void test()
7     {
8         Date d = new Date();
9         SimpleDateFormat format1 = new SimpleDateFormat("dd.MM.yyyy hh:mm");
10        SimpleDateFormat format2 = new SimpleDateFormat("День dd Месяц MM Год уууу
        Время hh:mm");
11        System.out.println(format1.format(d)); //25.02.2013 09:03
12        System.out.println(format2.format(d)); //День 25 Месяц 02 Год 2013 Время 09:03
13    }
14}
```

Очевидно, что при создании шаблона для отображения даты `dd` — означает день, `MM` — месяц, `уууу` — год, `hh` — часы и `mm` — минуты. В шаблоне могут присутствовать не все единицы, кроме того как вы увидели выше в качестве разделитель можно использовать любой текст.

Рассмотрим абстрактный класс `Calendar`. Он позволяет работать с датой в рамках календаря, т.е он умеет прибавлять день, при этом учитывать високосные годы и прочее. Единственной реализацией его является класс `GregorianCalendar`, также как и у даты конструктор по умолчанию возвращает календарь на текущий день, но вы можете задать его явно указав все параметры:

```
1 import java.text.SimpleDateFormat;
2 import java.util.Calendar;
3 import java.util.Date;
4 import java.util.GregorianCalendar;
5
6 public class Test {
7     public void test()
8     {
9         Calendar c = new GregorianCalendar();//календарь на текущую дату
10        Calendar c2 = new GregorianCalendar(2013, 11, 25);//календарь на 25.11.2013
11        c2.add(Calendar.DAY_OF_YEAR, 1); //увеличиваем дату на 1 день
12        System.out.println(c2.getTime());// 26.11.2013
13        c2.add(Calendar.DAY_OF_YEAR, -1); //уменьшаем дату на 1 день
14    }
15}
```

Календарь достаточно мощный класс, который позволяет получать названия месяцев и дней недели, увеличивать или уменьшать различные параметры текущей даты, а также получать их. Для удобства работы с ними вам нужно просто разобраться с типами данных с которыми он работает:

- DAY_OF_YEAR — день года (0- 365)
- DAY_OF_MONTH — день месяца (какой по счету день в месяце 0 — 31)
- WEEK_OF_MONTH — неделя месяца
- WEEK_OF_YEAR — неделя в году
- MONTH — номер месяца
- Year — номер года
- Calendar.ERA — эра

Т.е. большинство методов принимает на вход int field, где в качестве одного из вариантов вы можете выбрать перечисленные выше значения.

```
1 package ru.cybern;
2
3 import java.text.SimpleDateFormat;
4 import java.util.Calendar;
5 import java.util.Date;
6 import java.util.GregorianCalendar;
7
8 public class Test {
9     public void test()
10    {
11        Calendar c = new GregorianCalendar(2013, 11, 25);//календарь на 25.11.2013
12        System.out.println(c.get(Calendar.MONTH));//11
13        System.out.println(c.get(Calendar.YEAR));//2013
14        System.out.println(c.get(Calendar.DAY_OF_WEEK_IN_MONTH));//4
15        System.out.println(c.get(Calendar.DAY_OF_WEEK));//4
16        System.out.println(c.get(Calendar.DAY_OF_YEAR));//359
17        System.out.println(c.get(Calendar.DAY_OF_MONTH));//25
18    }
19}
```