

Date e orari

Le date sono degli oggetti molto complessi da gestire: assumono forme diverse a seconda del luogo geografico in cui ci troviamo. La manipolazione delle date e delle ore è una delle attività ricorrenti di un programmatore.

Inoltre sono disponibili le più moderne `LocalDate` e `LocalDateTime`. Oltre a queste classi, è molto probabile che serva utilizzarne altre quali: `DateFormat` e `SimpleDateFormat`, che permettono la trasformazione da stringa a data e viceversa.

In Java 8: `LocalDate`, `LocalTime`, `LocalDateTime`

```
LocalDate oggi = LocalDate.now(); // Data di oggi

System.out.println("oggi: " + oggi);

//settare una data precisa (mese 1-based)
LocalDate mauroBirthday = LocalDate.of(1969, 7, 28);

//Per il mese possiamo usare le costanti di Month
mauroBirthday = LocalDate.of(1969, Month.JULY, 28);

System.out.println("mauroBirthday: " + mauroBirthday);
```

In Java 8: `LocalDate`, `LocalTime`, `LocalDateTime` II

```
LocalDate inizioCorsi = LocalDate.of(2017, Month.OCTOBER, 12);
LocalDate natale = LocalDate.of(2017, Month.DECEMBER, 25);

System.out.println("Fino a natale: " + inizioCorsi.until(natale));
System.out.println("Fino a natale: " + inizioCorsi.until(natale, ChronoUnit.DAYS));

LocalDate festaLiberazione = LocalDate.of(2017, Month.APRIL, 25);
LocalDate natale = LocalDate.of(2017, Month.DECEMBER, 25);

System.out.println("Fino a natale: " + festaLiberazione.until(natale));
```

```
System.out.println("Fino a natale: " + festaLiberazione.until(natale, ChronoUnit.DAYS));
```

```
System.out.println(oggi.plusMonths(1));
```

```
System.out.println(oggi.minusMonths(1));
```

In Java 8: LocalDate, LocalTime, LocalDateTime III

```
DayOfWeek inizioMillennio = LocalDate.of(2000, 1, 1).getDayOfWeek();
```

```
System.out.println("inizioMillennio: " + inizioMillennio);
```

```
System.out.println(inizioMillennio.getValue());
```

```
System.out.println(DayOfWeek.SATURDAY.plus(3));
```

```
LocalDateTime ldt = LocalDateTime.now();
```

```
System.out.println(ldt);
```

```
LocalDate ld = LocalDate.of(2009, 1, 28);
```

```
System.out.println(ld);
```

```
DateTimeFormatter dtf = DateTimeFormatter.ofPattern("M/d/yyyy");
```

```
System.out.println(dtf.format(ld));
```

Convertire LocalDate a java.sql.Date

```
import java.sql.Date;
```

```
//...
```

```
LocalDate locald = LocalDate.of(1969, 07, 28);
```

```
Date date = Date.valueOf(locald);
```

```
r.setDateOfBirth(date);
```

l'operazione contraria è

```
Date date = r.getDate();
```

```
LocalDate localD = date.toLocalDate();
```

r è il record e `.getDate()` il metodo per farsi ritornare la data. Se avessi un campo `dataNascita` il metodo dovrebbe chiamarsi `getDateNascita()`.

Usa le classi del package `java.time` invece di `java.util.Date` e `java.sql.Date` con **** JDBC 4.2 o superiore.****

Esempio con `PreparedStatement`

```
myPreparedStatement.setObject(
```

```
    ... ,                                // qui passa il numero ordinale dell'argomento.
```

```
    myJavaUtilDate.toInstant()           // Converti da `java.util.Date` nel più moderno  
    `java.time.Instant` (UTC).
```

```
        .atZone( ZoneId.of( "Europe/Rome" ) ) // Sette un time zone particolare, per determinare la  
data. Istanziando un `ZonedDateTime`.
```

```
        .toLocalDate()                   // Estrai la data di tipo `java.time.LocalDate` dall'oggetto.
```

```
)
```

esempi

```
LocalDate todayLocalDate = LocalDate.now( ZoneId.of( "Europe/Paris" ) );
```

```
// Usare "continent/region" come region name; non usare quelli codificati da 3 lettere.
```

```
LocalDate localDate = ResultSet.getObject( 1 , LocalDate.class );
```

```
//la questione è irrelevante in JDBC 4.2 o successive.
```

Converti a `java.sql.Date`

```
java.sql.Date sqlDate = java.sql.Date.valueOf( todayLocalDate );
```

```
//viceversa
```

```
LocalDate localDate = sqlDate.toLocalDate();
```

Nuovi metodi di java.util.Date

- `java.util.Date.from(Instant)`
- `java.util.Date::toInstant.`

```
Instant instant = myUtilDate.toInstant();
```

```
ZonedDateTime zoneld = ZonedDateTime.of ( "America/Montreal" );
```

```
ZonedDateTime zdt = ZonedDateTime.ofInstant ( instant , zoneld );
```

```
LocalDate localDate = zdt.toLocalDate();
```

```
public class MainClass {
```

```
    public static void main(String[] args) {
```

```
        java.util.Date utilDate = new java.util.Date();
```

```
        java.sql.Date sqlDate = new java.sql.Date(utilDate.getTime());
```

```
        System.out.println("utilDate:" + utilDate);
```

```
        System.out.println("sqlDate:" + sqlDate);
```

```
    }
```

```
}
```

```
//http://www.java2s.com/Tutorial/Java/0040__Data-  
Type/ConvertfromajavautilDateObjecttoajavasqlDateObject.htm
```