

8.9

Метка времени (timestamp)

00:00–00:51

Введение

Привет!

Часто вместо даты и времени используются так называемые метки времени — числа, которые удобно хранить и с которыми часто удобнее работать. В этой теме вы познакомитесь с этим понятием (по-английски — `timestamp`) и с тем, как эту метку преобразовать в дату и время и обратно.

Метка времени в классическом понимании — это количество секунд, прошедших с полуночи 1 января 1970 года. Эта дата принята универсальной во всём компьютерном мире для отсчёта времени. То есть, по сути, любое время и любую дату можно выразить в виде такого числа. На самом деле, в большинстве объектов, которые работают с датами и со временем, время и дата как раз и хранятся в виде таких чисел.

00:51–02:06

Метка времени (timestamp) в Java

В Java метку времени принято использовать не в секундах, а в миллисекундах для большей точности.

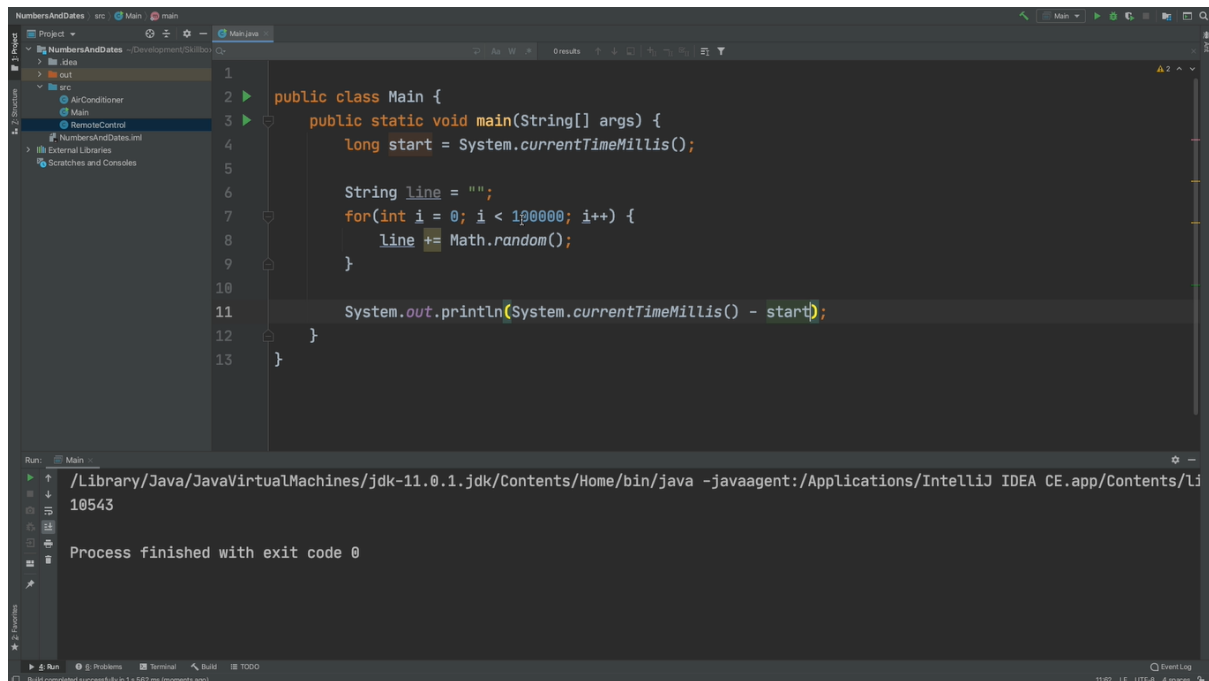
Получить текущую метку времени очень просто: достаточно использовать статический метод `currentTimeMillis` класса `System` — это будет время в миллисекундах.

Такую метку времени очень удобно использовать, например, для измерения длительности выполнения какого-нибудь программного кода.

Например, есть некий код — строка, к которой мы будем прибавлять 100 000 раз случайное число с помощью метода `math.random`.

Длительностью выполнения этой операции у нас здесь будет метка времени — в конце мы найдём текущую метку времени и вычтем из неё значение конечной метки времени.

Запустим этот код и увидим, что код выполнялся 10,5 секунд:



```
1 public class Main {
2     public static void main(String[] args) {
3         long start = System.currentTimeMillis();
4
5         String line = "";
6         for(int i = 0; i < 100000; i++) {
7             line += Math.random();
8         }
9
10        System.out.println(System.currentTimeMillis() - start);
11    }
12 }
13 }
```

Run: Main

/Library/Java/JavaVirtualMachines/jdk-11.0.1.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/Li
10543

Process finished with exit code 0

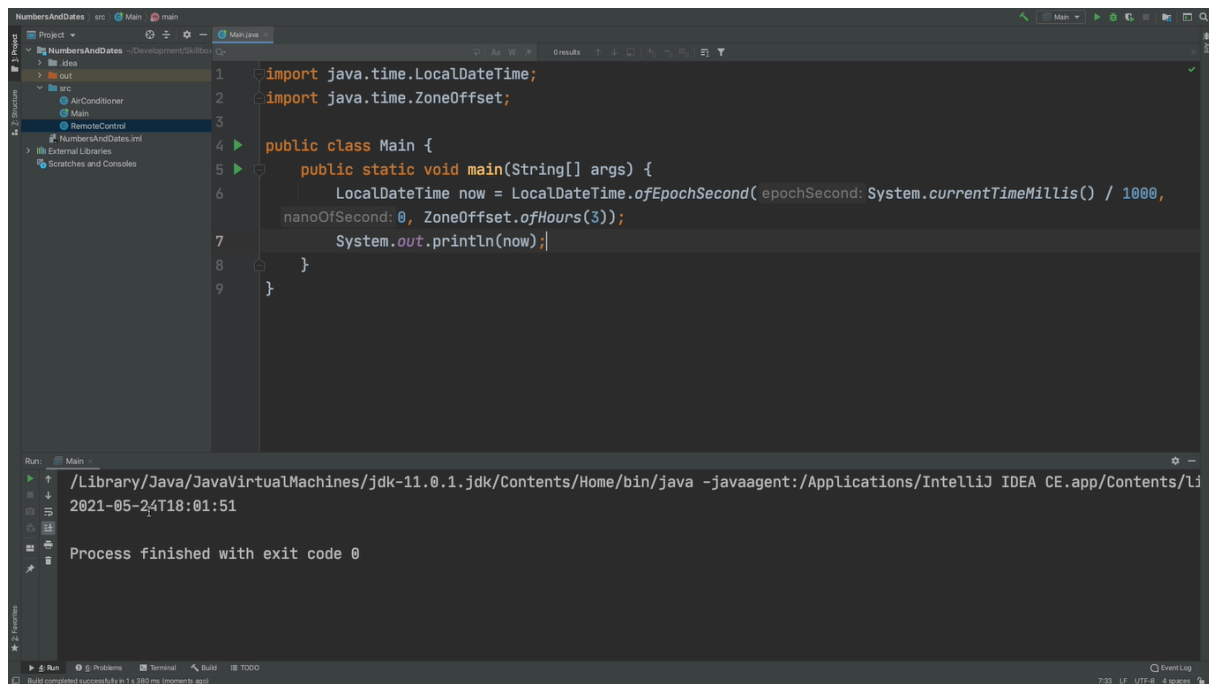
02:06–02:59

Преобразование метки времени в дату

Посмотрим, как можно преобразовать метку времени в дату, поскольку в ряде систем используют хранение времени именно в этом формате.

Для такого преобразования можно воспользоваться специальным статическим методом класса `LocalDateTime` — `ofEpochSecond`, куда передаётся три параметра:

- первый параметр — время в секундах;
- второй параметр — время в наносекундах;
- третий параметр — это часовой пояс. Мы будем отмерять его в часах, ориентируясь на московский часовой пояс, прибавляя число 3.



```
1 import java.time.LocalDateTime;
2 import java.time.ZoneOffset;
3
4 public class Main {
5     public static void main(String[] args) {
6         LocalDateTime now = LocalDateTime.ofEpochSecond( epochSecond: System.currentTimeMillis() / 1000,
7             nanoOfSecond: 0, ZoneOffset.ofHours(3));
8         System.out.println(now);
9     }
10 }
```

Run: Main

/Library/Java/JavaVirtualMachines/jdk-11.0.1.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/Li
2021-05-24T18:01:51

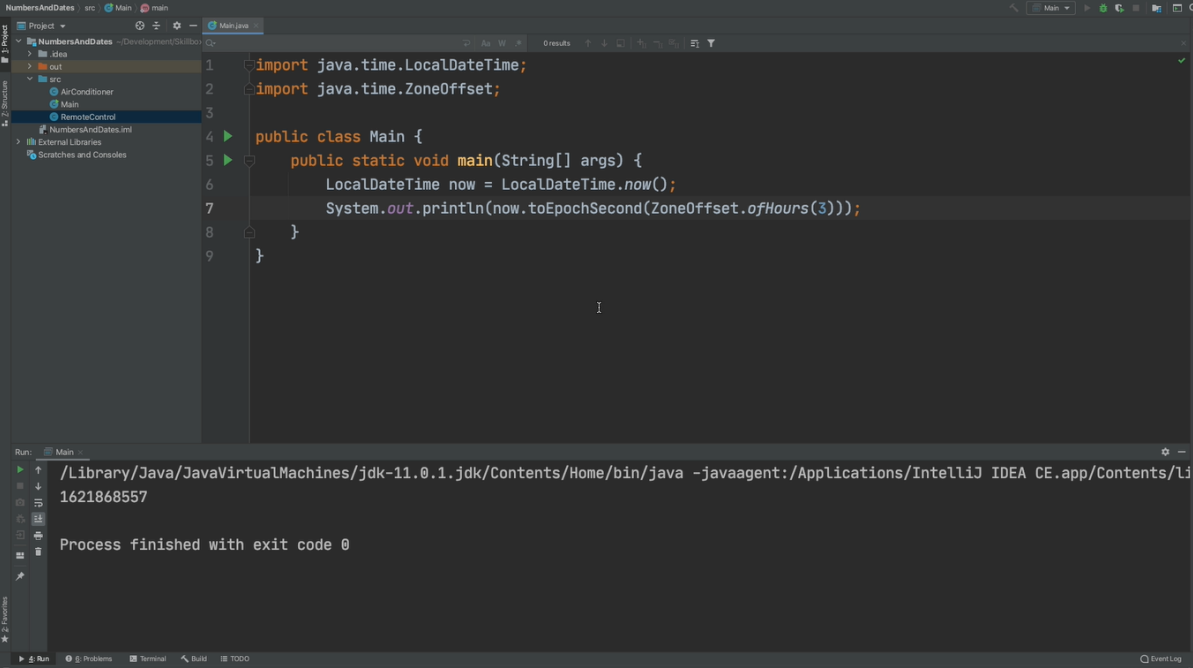
Process finished with exit code 0

Видим, что всё работает правильно.

02:59–03:29

Получение метки времени из даты

Рассмотрим, как получить метку времени из даты. Для этого достаточно снова воспользоваться методом `ofEpochSecond`. Например, если мы возьмём `LocalTime` и попробуем вывести метку времени (нужно будет указать часовой пояс, пусть это опять будет часовой пояс для Москвы), то получим текущую метку времени в секундах:



The screenshot shows the IntelliJ IDEA IDE interface. The main editor displays a Java file named `Main.java` with the following code:

```
1 import java.time.LocalDateTime;
2 import java.time.ZoneOffset;
3
4 public class Main {
5     public static void main(String[] args) {
6         LocalDateTime now = LocalDateTime.now();
7         System.out.println(now.toEpochSecond(ZoneOffset.ofHours(3)));
8     }
9 }
```

The left sidebar shows the project structure for `NumbersAndDates`, including `src` and `out` directories. The bottom panel shows the Run console output:

```
/Library/Java/JavaVirtualMachines/jdk-11.0.1.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/Li
1621868557
Process finished with exit code 0
```

03:29–03:47

Выводы

Итак, в этой теме вы познакомились с понятием метки времени, то есть `timestamp`. Вы узнали, как её можно получить вообще и как её можно получить из объекта с датой и временем. Также узнали, как её преобразовать в объект с датой и временем.