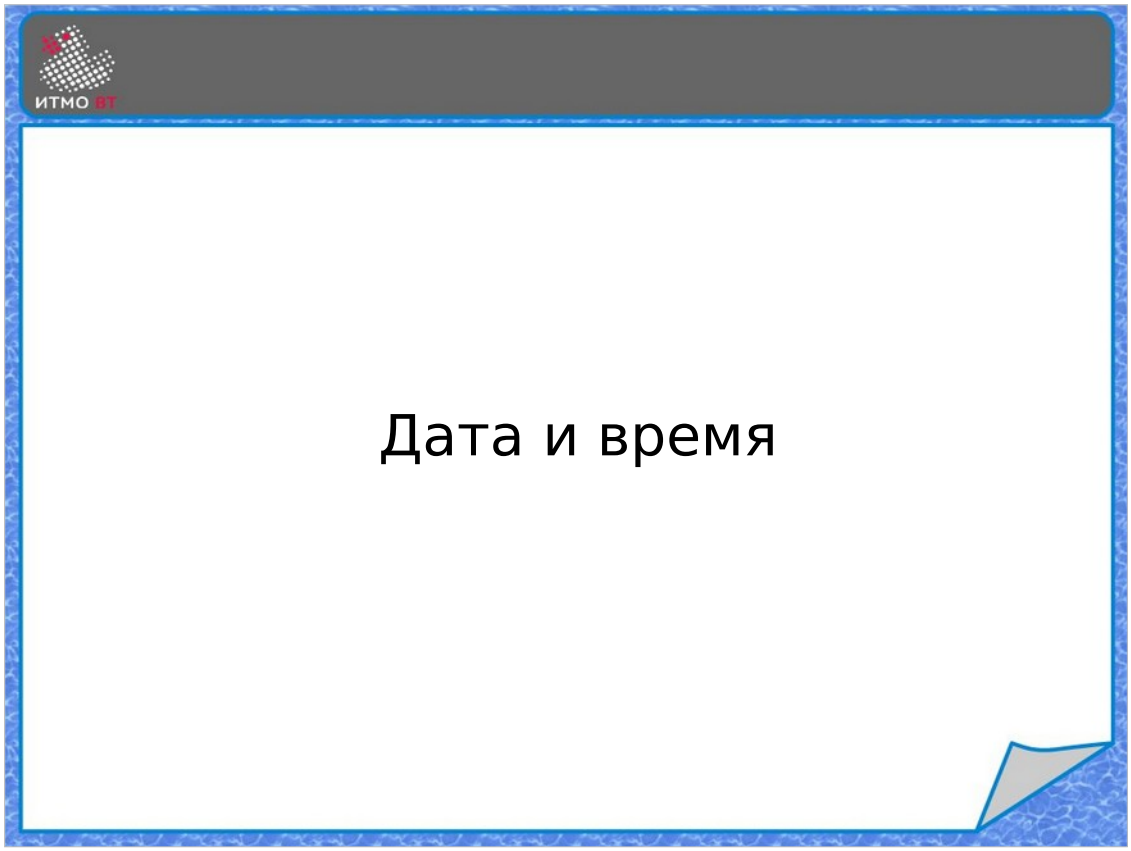
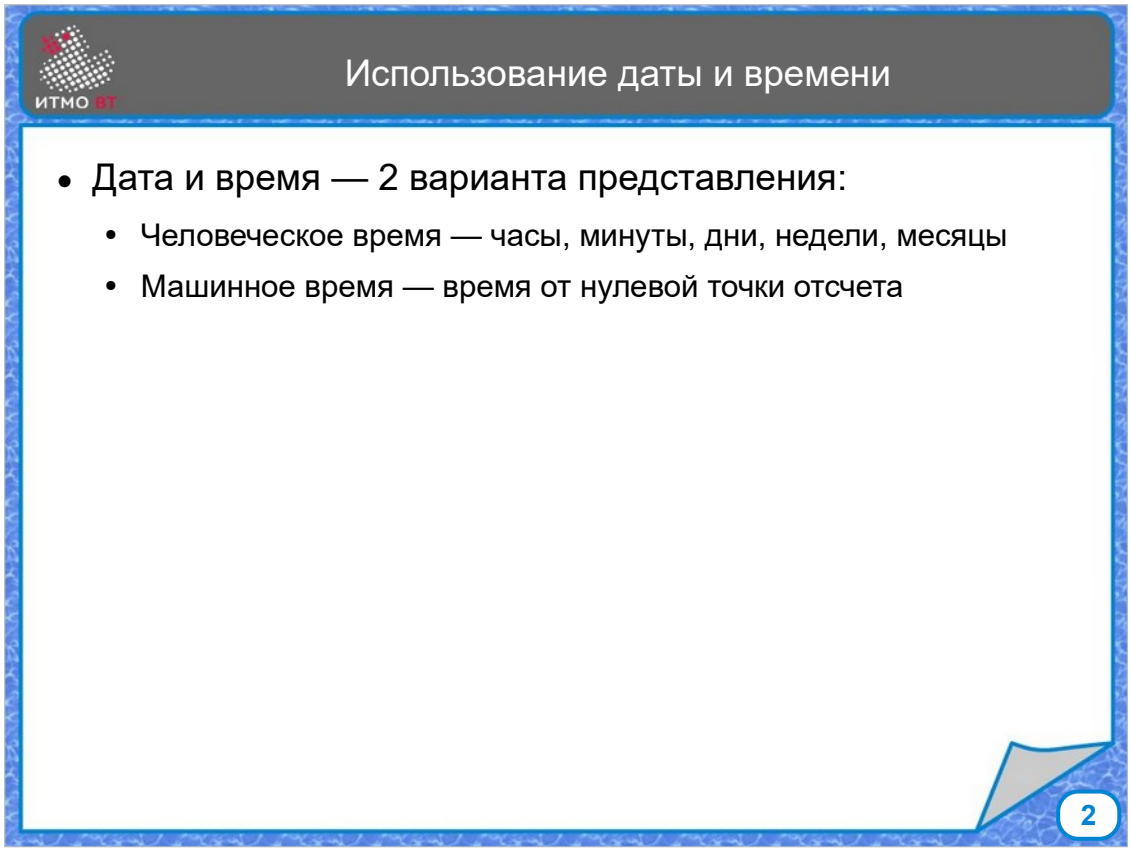
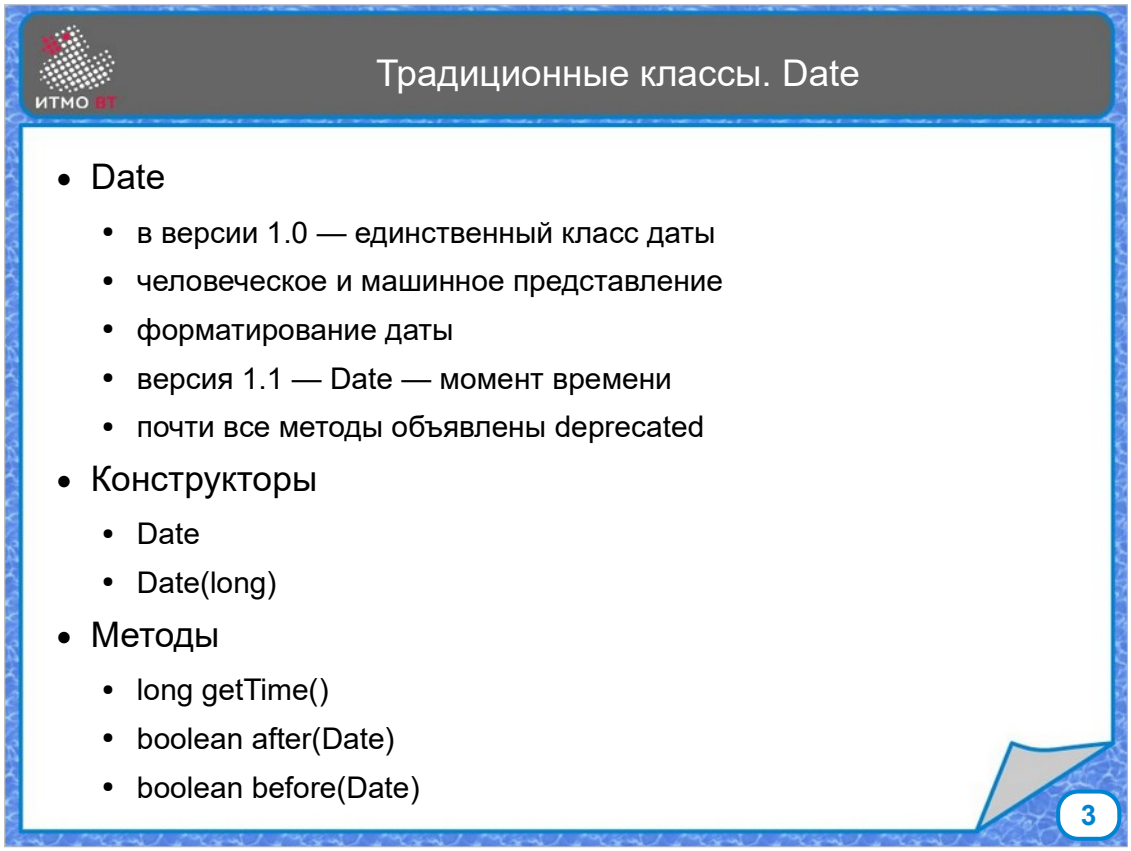
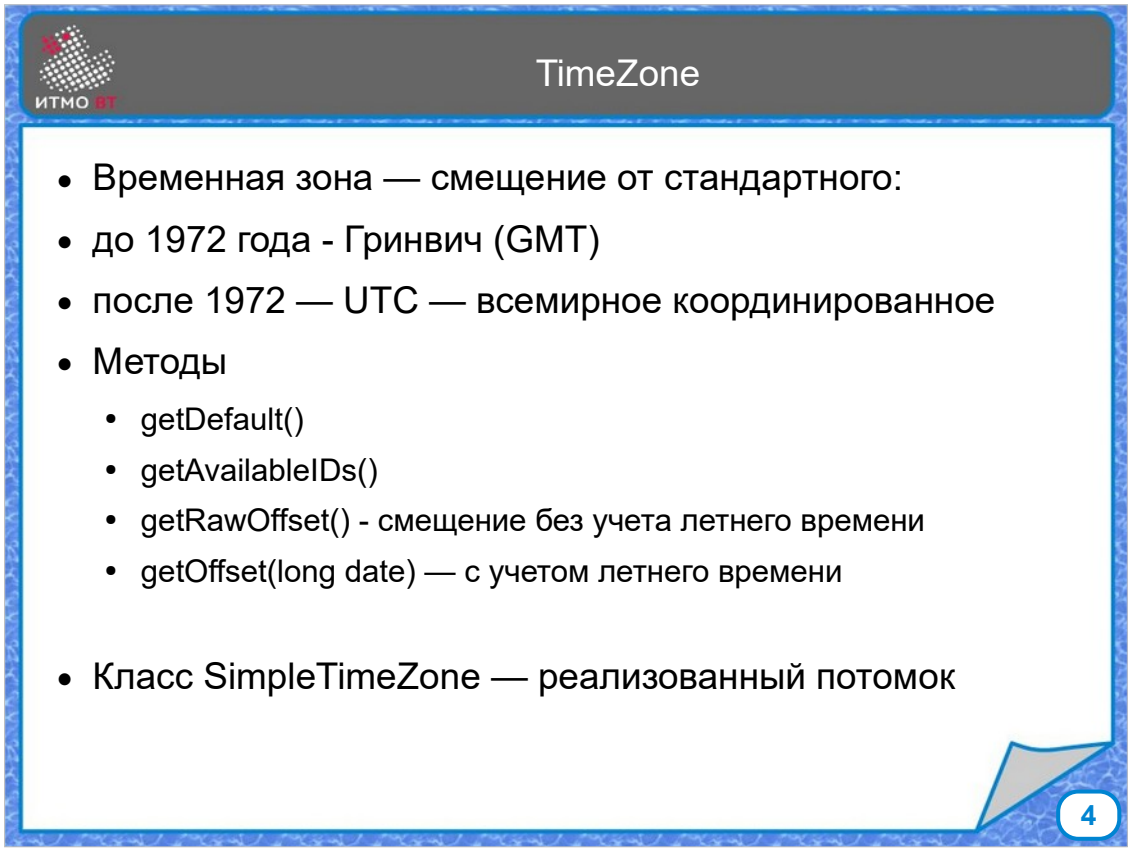

ИТМО ВТ
Дата и время



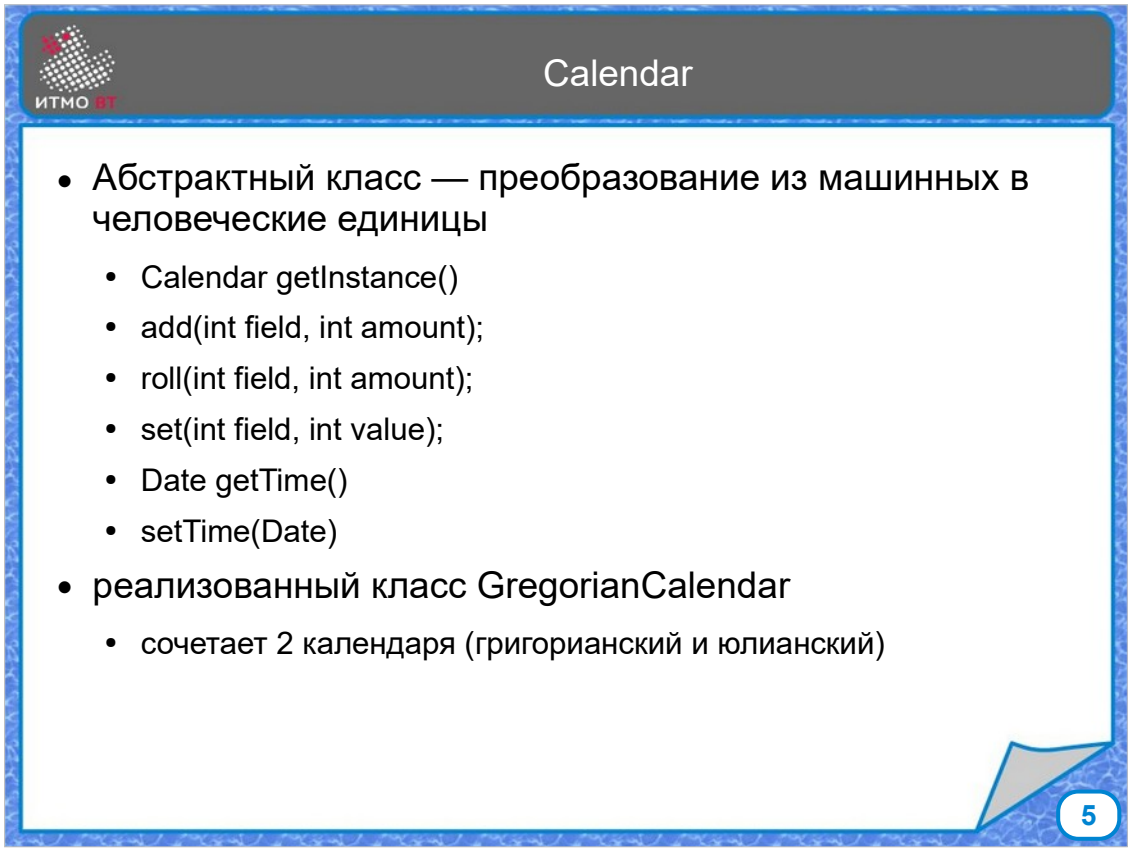
Для представления даты и времени обычно используется 2 основных формата - человеческое представление, состоящее из часов, минут, секунд, дней, недель, месяцев и т. д., и машинное представление, которым удобно оперировать при хранении дат и вычислениях с датой, временем и временными интервалами. Машинное представление обычно использует какой-то момент времени в качестве начального (epoch), и представляет собой количество единиц времени (секунд, миллисекунд или других), прошедших от этого начального момента. В разных языках программирования, приложениях, ОС приняты разные начальные моменты времени. Для Windows API - начальный момент 1 января 1601 года. Во многих языках программирования, в том числе C, C++, Java, Javascript, Python, ОС UNIX, Linux, MacOS за начальный принят момент полуночи 1 января 1970 года (UNIX epoch). Если дата представляется в 32-битном формате как количество секунд с 1 января 1970 года, то 19 января 2038 года в 3:14:08 произойдет переполнение.



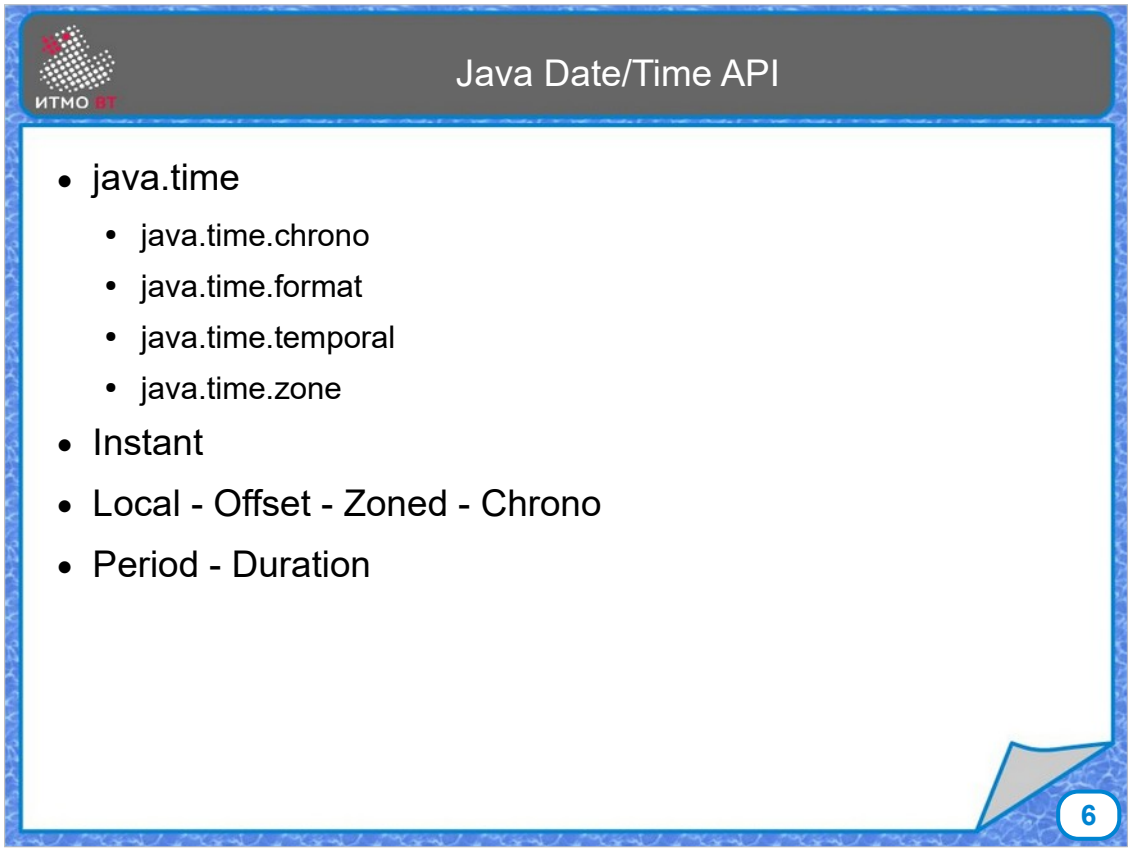
Изначально для хранения даты был предназначен один класс - Date, который совмещал в себе оба формата, и имел большое количество методов для операций преобразования между форматами, а также для форматирования. Начиная с версии 1.1 класс Date рекомендуется использовать только для представления момента времени, и почти все его методы объявлены устаревшими. Однако, во многих библиотеках этот класс до сих пор используется, при этом не устаревшими являются только 2 конструктора - без параметров (создающий объект Date с текущим временем, и конструктор с параметром типа long, позволяющий задать момент в миллисекундах от UNIX epoch. Также можно использовать методы getTime, equals, before и after, позволяющие сравнивать даты.



Класс TimeZone представляет смещение от стандартного времени для какой-то конкретной временной зоны. В качестве стандартного в настоящее время используется всемирное координированное время (UTC). Смещение может быть как базовым (raw), которое учитывает только часовой пояс, так и для конкретной зоны, которое учитывает дополнительные изменения времени, например, переход на летнее и на зимнее время. При этом моменты введения летнего времени в разных временных зонах хранятся и учитываются.

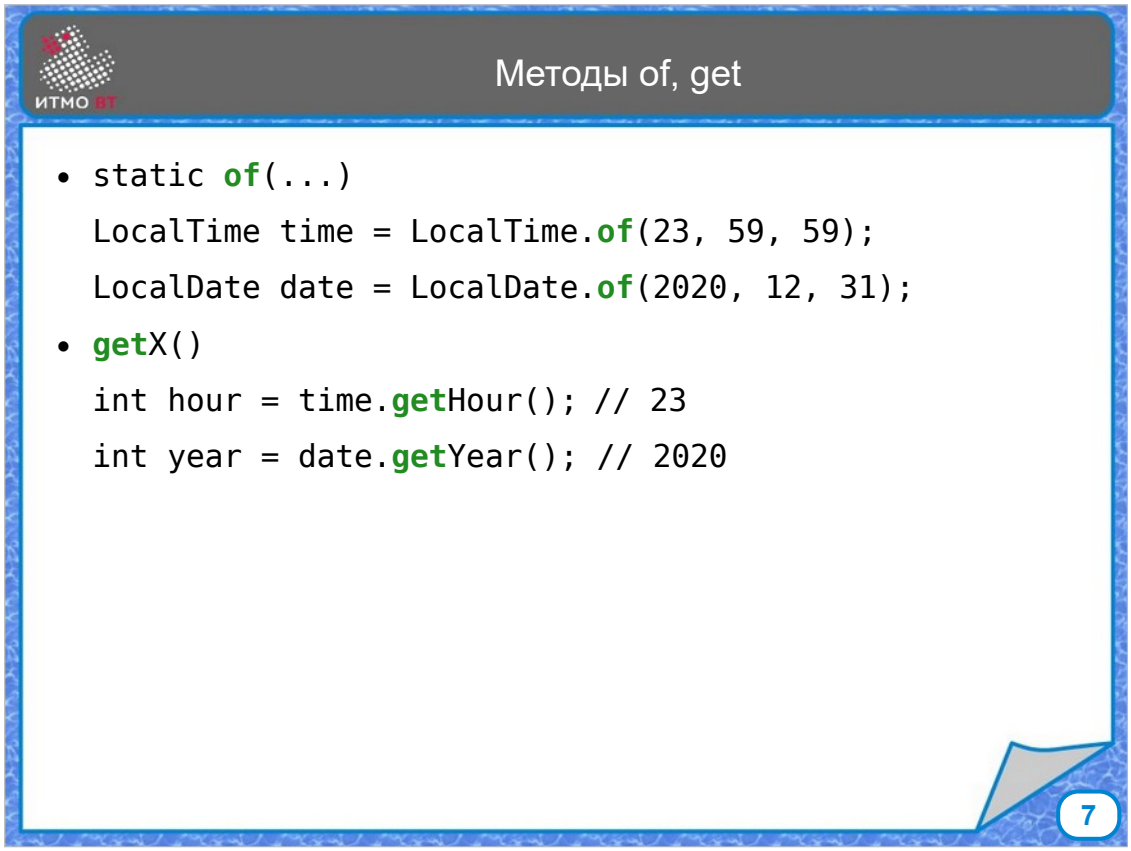


Класс Calendar - абстрактный класс, основным назначением которого является преобразование даты и времени между машинным и человеческим представлением. Для реального преобразования используется класс GregorianCalendar, сочетающий представление григорианского и юлианского календарей.



В 8 версии Java появилась новая библиотека для работы с датами и временем. Основные классы, позволяющие использовать стандартные временные величины и григорианский календарь в соответствии с ISO-8601, находятся в пакете java.time. В остальных пакетах располагаются классы для работы с другими календарями (chrono), форматирования (format), доступа к полям даты и времени (temporal) и временных зон (zone).

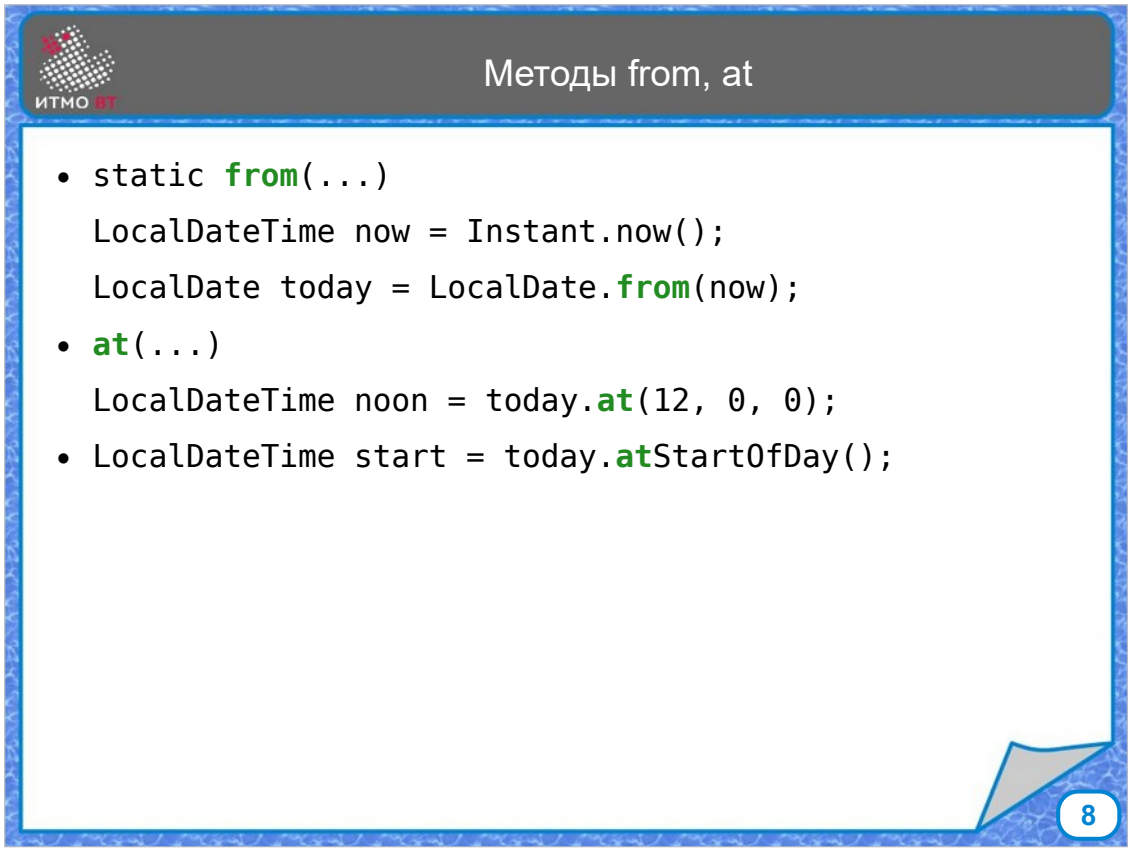
Основные классы библиотеки - Instant (момент времени), классы для локальных даты/времени, которые могут использоваться для большинства задач, поясного времени (offset) для работы с сетевыми протоколами и базами данных, конкретными временными зонами (zoned) с учетом местных особенностей, и нестандартными календарями (chrono). Также представлены классы для промежутков времени: коротких (Duration) и длинных (Period).



Классы, представляющие дату и время, имеют набор стандартных методов.

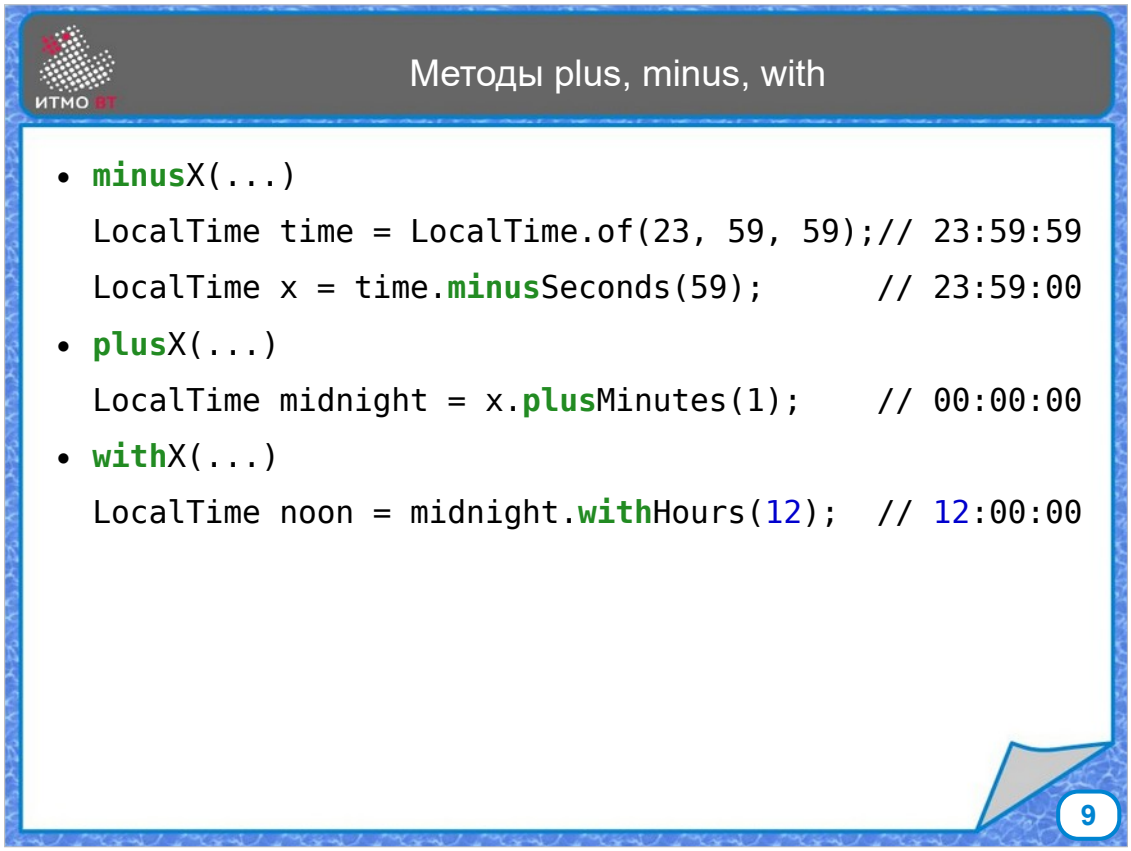
Статический фабричный метод of позволяет создать объект по заданным в параметрах полям.

Метод get позволяет получить значение любого поля.

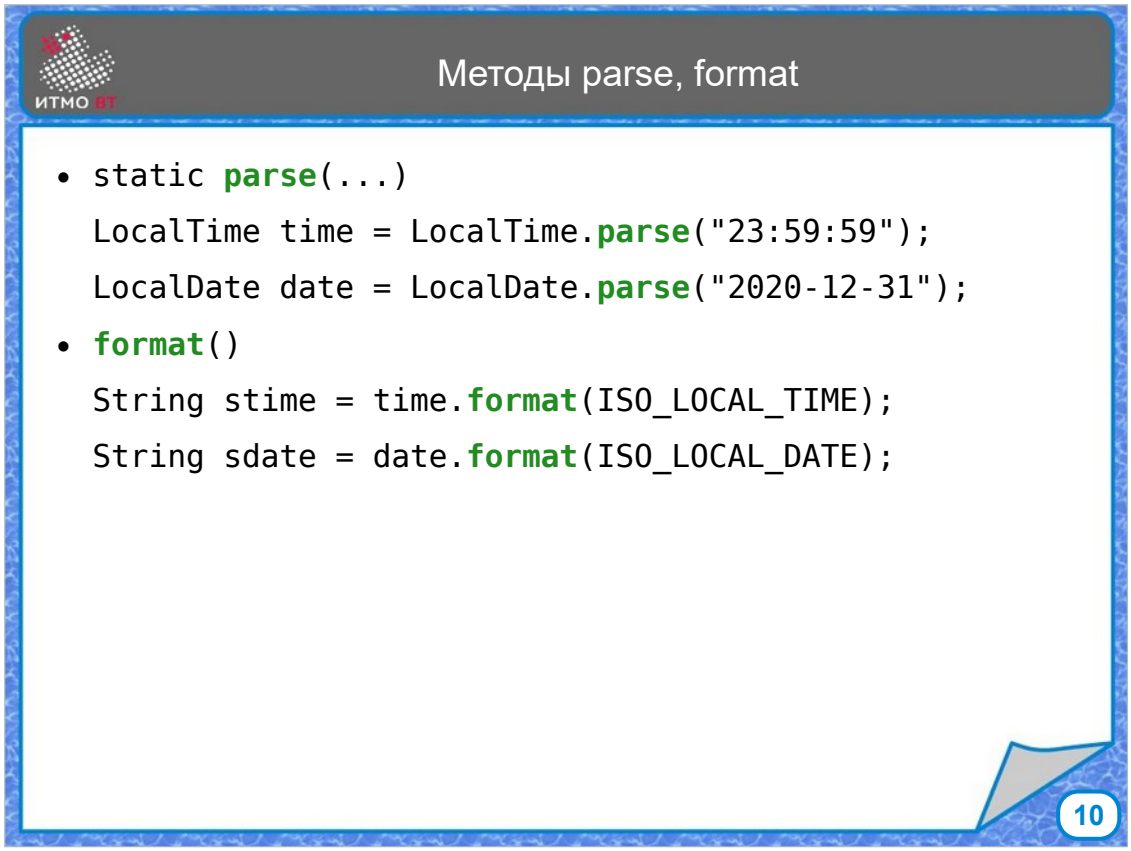


Статический метод from позволяет получить объект одного типа из объекта другого типа.

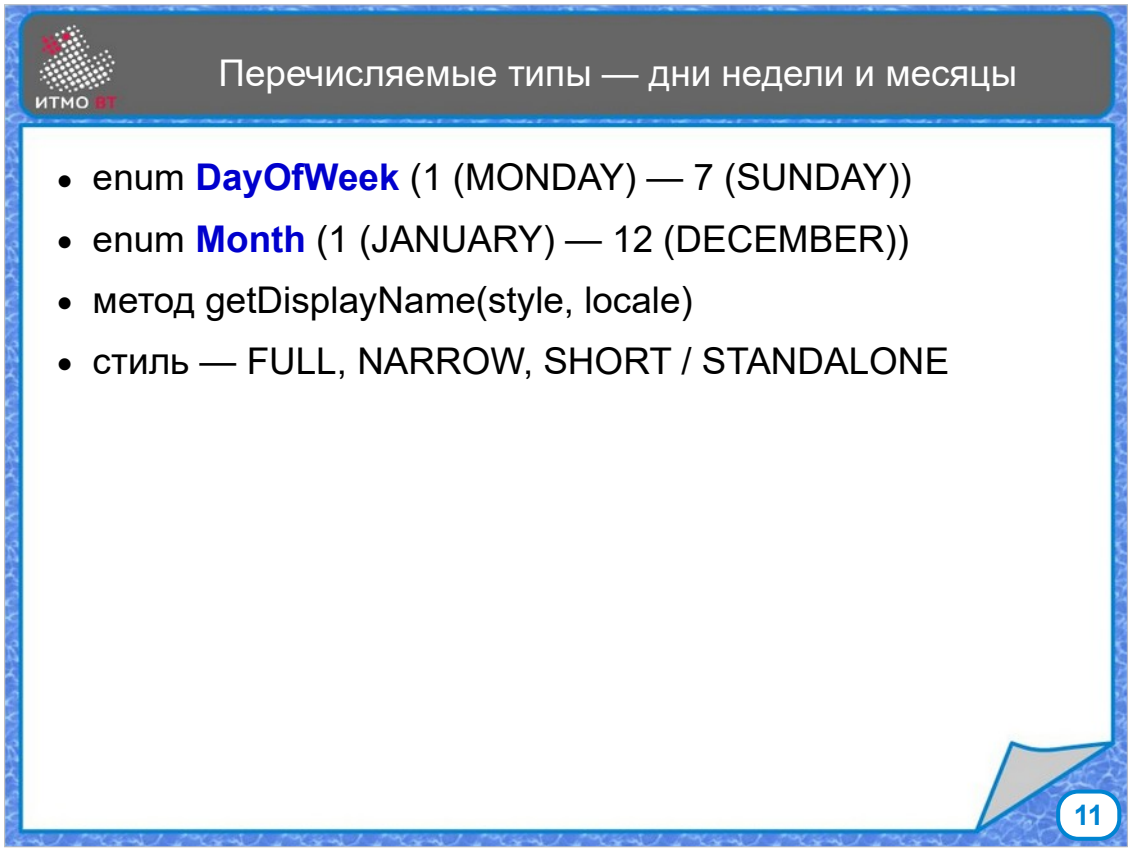
Метод at позволяет получить объект из другого объекта, дополнив или уточнив значения других полей.



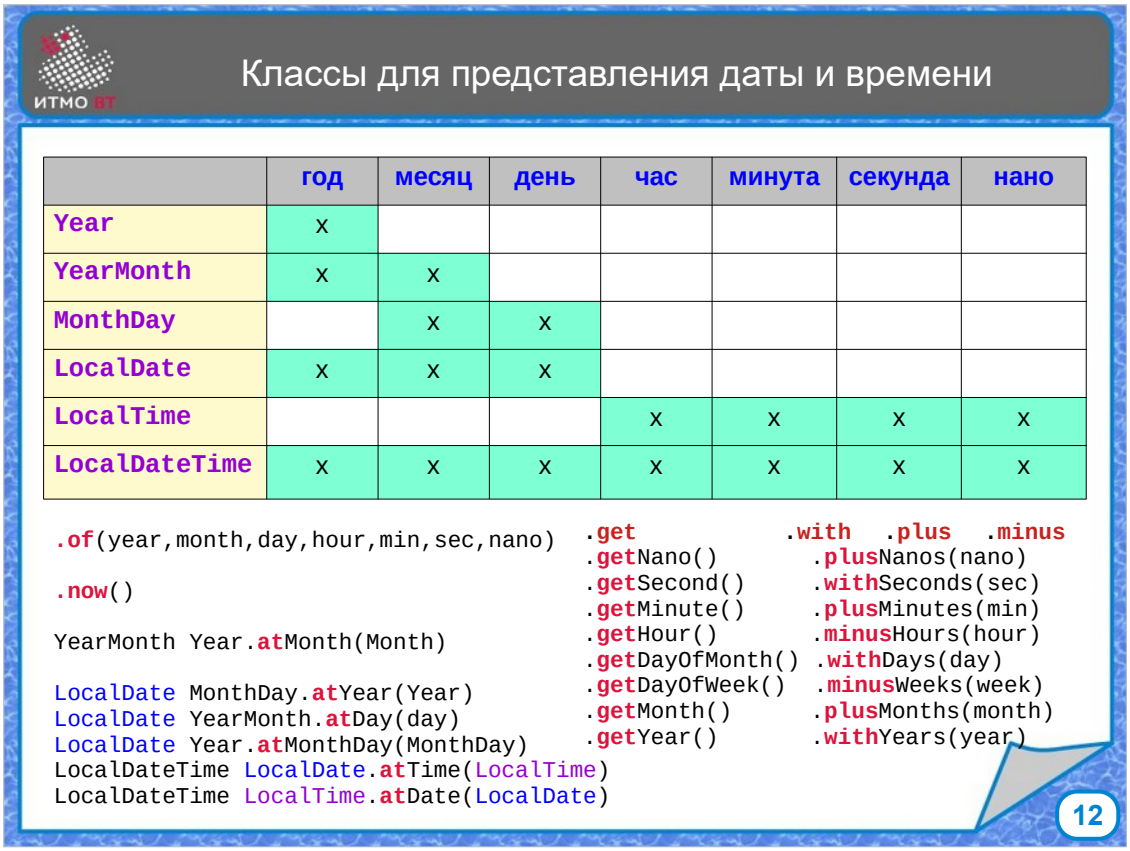
Методы plus, minus, with позволяют получить копию объекта путем сложения, вычитания или установки значения какого-либо поля.



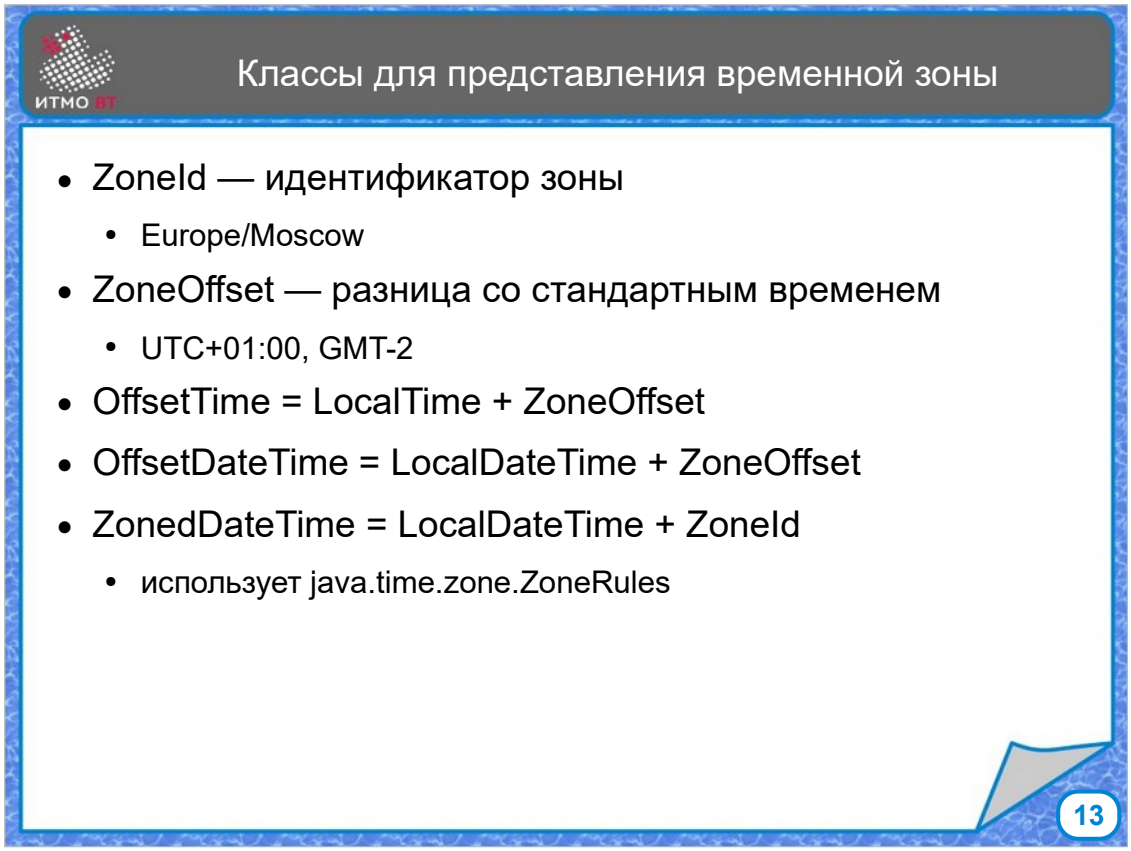
Метод parse позволяет создать объект из строкового представления, а метод formar - преобразовать объект в строку.



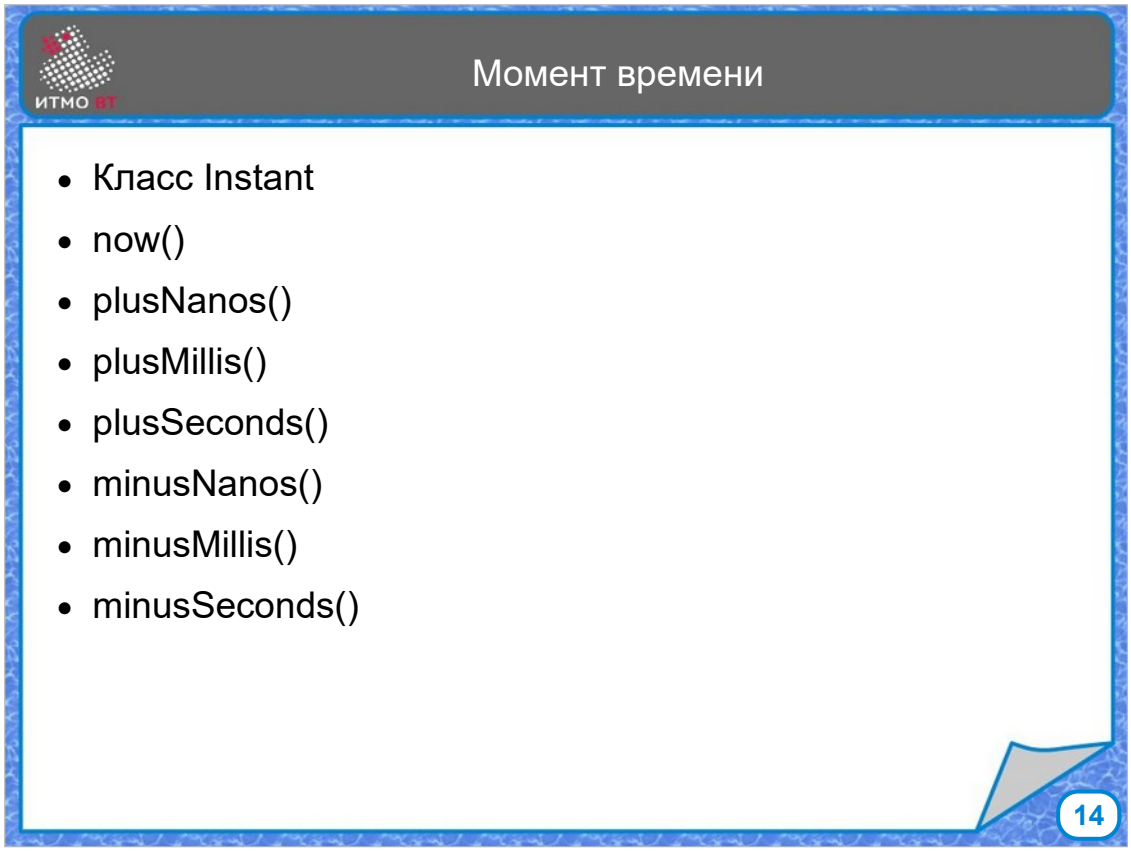
Пакет java.time содержит перечисляемые типы - дни недели и названия месяцев



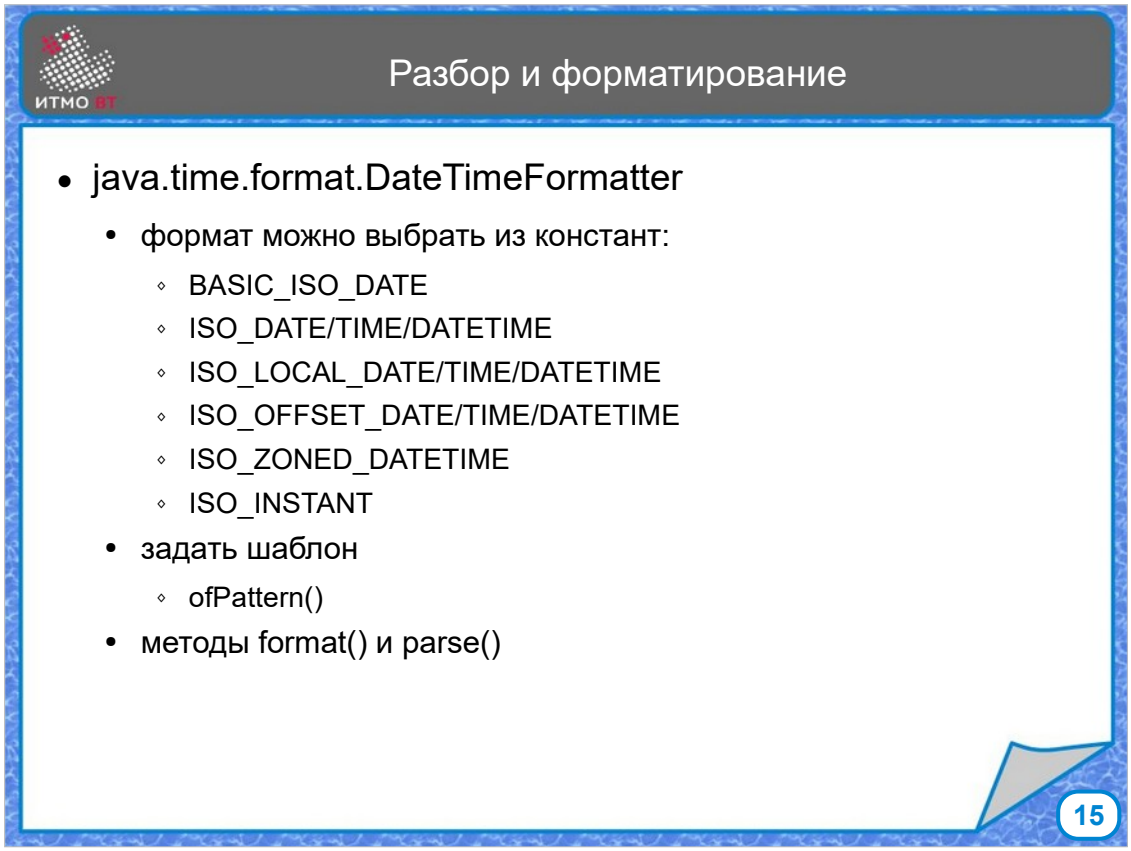
В таблице приведены основные классы для представления даты и времени, класс LocalDate состоит из года, месяца и дня, класс LocalTime - часы, минуты, секунды и наносекунды. класс LocalDateTime - комбинация даты и времени. Методы уже рассматривались, на слайде приведены примеры имеющихся методов.



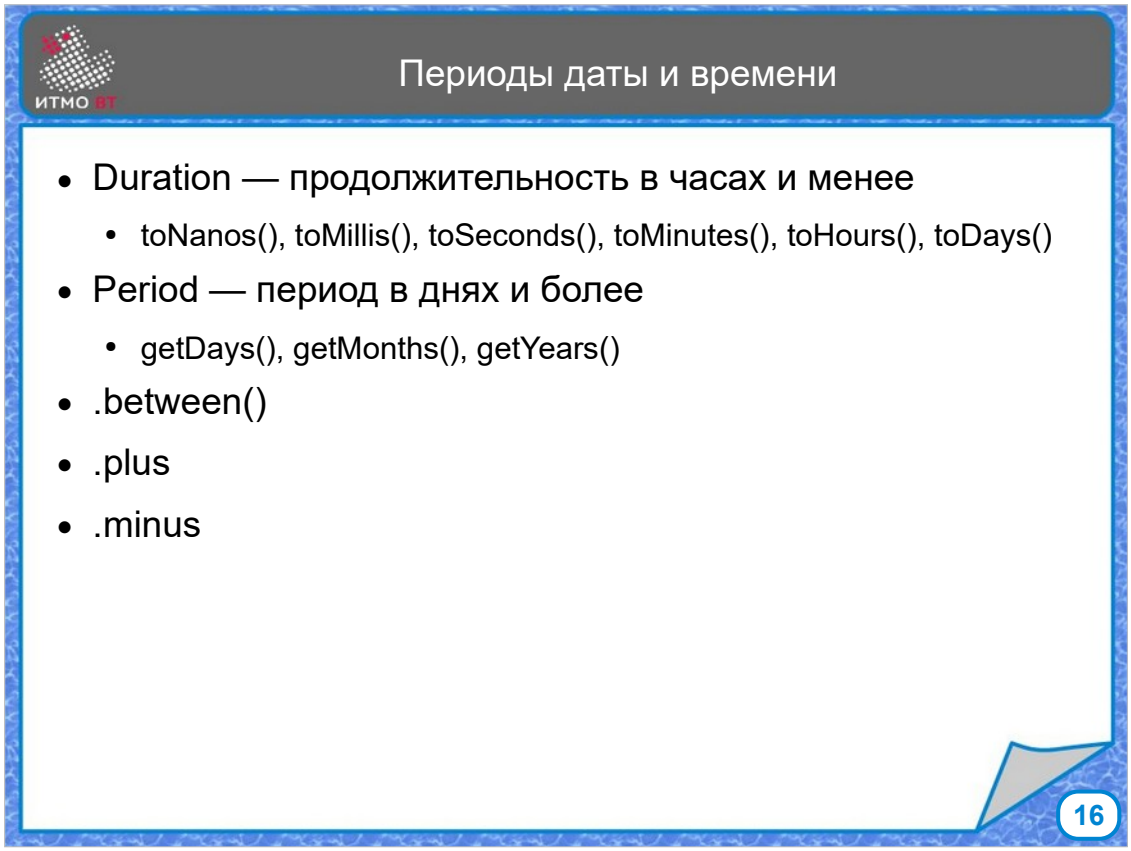
Классы, в которых учитывается часовой пояс или временная зона. Класс ZoneId предназначен для идентификации зоны. Класс ZoneOffset показывает разницу с всемирным временем. Классы OffsetTime и OffsetDateTime представляют время и дату для часового пояса без учета местных особенностей и локальных переводов времени. И класс ZonedDateTime - это класс, учитывающий время в конкретной местности. Он использует класс ZoneRules для точного представления конкретного момента времени в данной временной зоне с учетом всех переводов времени, в том числе происходивших в прошлом.



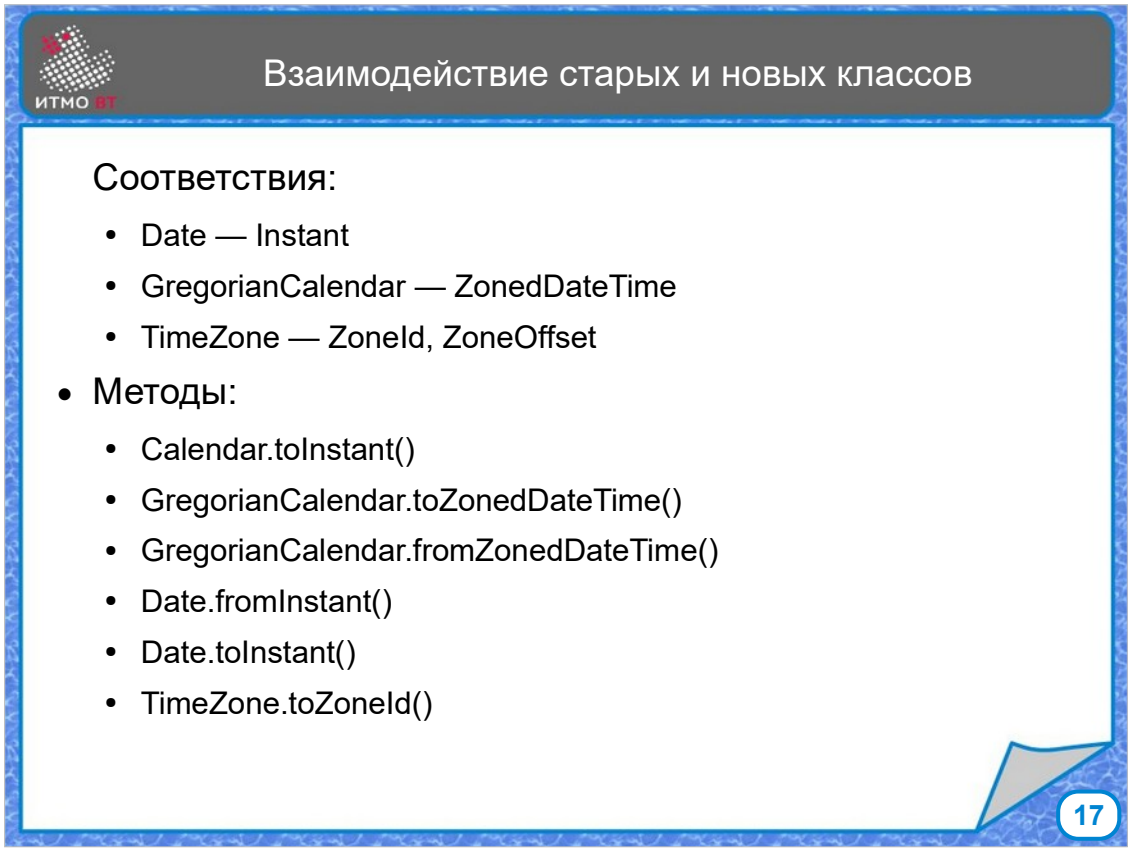
Класс Instant предствляет момент времени, представленный в виде количества миллисекунд, прошедших с начального момента UNIX epoch.



Для форматирования даты и времени предназначен класс DateTimeFormatter, который позволяет отформатировать дату и время в соответствии с заранее заданными шаблонами, задать шаблон самостоятельно, получить локализованный шаблон для конкретной местности. У этого класса есть методы format и parse, первый из которых предназначен для форматирования, а второй - для разбора строки по заданному формату.



Для задания периода времени между двумя датами или моментами времени есть два класса - для периодов времени, измеряющихся в часах или более мелких единицах - Duration, и периодов между датами, измеряющихся в днях, месяцах, годах - Period. Оба класса имеют методы between для формирования периода, а также plus и minus для увеличения или уменьшения периода.



Аналогом старого класса Date является класс Instant, аналогом GregorianCalendar - класс ZonedDateTime. Ну и аналогом класса TimeZone является класс ZoneId. У всех этих классов есть методы, позволяющие преобразовать данные для использования либо со старой, либо с новой библиотекой.