Date Time API

Основные пакеты

Для работы со временем в Java реализованы следующие библиотеки:

- пакет **java.time** — это базовый пакет нового Date Time API. Все основные базовые классы являются частью этого пакета: *LocalDate, LocalTime, LocalDateTime, Instant, Period, Duration* и другие. Все эти классы являются неизменными и потокобезопасными. В большинстве случаев этих классов будет достаточно для большей части задач;

- пакет **java.time.chrono** — пакет с общими интерфейсами для не календарных систем ISO. Мы можем наследовать класс AbstractChronology для создания собственной календарной системы;

- пакет **java.time.format** — пакет с классами форматирования и парсинга времени и даты. В большинстве случаев мы не будем использовать их напрямую, потому что классы в пакете *java.time* предоставляют удобные методы для форматирования и парсинга;

- пакет **java.time.temporal** используется для работы с временными объектами. Например, с помощью него мы можем узнать первый или последний день месяца. Методы таких классов сразу заметны на фоне других, потому что всегда имеют формат ‘withXXX‘;

- пакет **java.time.zone** — классы для поддержки различных часовых поясов и правила их изменения.

Каждый из данных пакетов содержит классы для работы с датой и временем, а также с периодами времени.

Класс java.time.LocalDate

java.time.LocalDate — неизменяемый класс, который представляет объекты Date в формате по умолчанию yyyy-MM-dd. Мы можем использовать метод **now**(), чтобы получить текущую дату. Мы также можем предоставить в качестве аргументов год, месяц и день, чтобы создать экземпляр LocalDate. Этот класс предоставляет перегруженный метод *now()*, которому мы можем предоставить *ZoneId* для получения даты в конкретном часовом поясе. Этот класс предоставляет такую же функциональность, как java.sql.Date.

Пример:

```
// Получаем текущую дату
LocalDate today = LocalDate.now();
System.out.println("Текущая дата : " + today);
//Создадим LocalDate и в качестве аргументов
//укажем год месяц и день
LocalDate specificDate = LocalDate.of(2023, Month.NOVEMBER, 30);
```

System.out.println("Дата с указанием года, месяца и дня : " + specificDate);

Класс java.time.LocalTime

Класс **LocalTime** является неизменяемым и представляет собой время в читабельном виде. По умолчанию он предоставляет формат hh:mm:ss.zzz. Так же, как и LocalDate, этот класс обеспечивает поддержку часовых поясов и создание даты, передавая в качестве аргументов часы, минуты и секунды.

Пример:

```
// получаем текущее время
LocalTime time = LocalTime.now();
System.out.println("Получаем текущее время : " + time);
//Создаем LocalTime с использование своих данных в качестве параметров
LocalTime specificTime = LocalTime.of(23, 15, 11, 22);
System.out.println("Какое-то время дня : " + specificTime);
```

Вспомогательные методы Date API

Пример:

```
LocalDate today = LocalDate.now();
//Получаем год, проверям его на високосность
System.out.println("Год " + today.getYear() + " – высокосный? : " +today.isLeapYear());
//Сравниваем два LocalDate: до и после
System.out.println("Сегодня – это до 02.03.2023? : " + today.isBefore(LocalDate.of(2023,3,2)));
//Создаем LocalDateTime с LocalDate
System.out.println("Текущее время : " + today.atTime(LocalTime.now()));
//Операции сложения и вычитания с датами
System.out.println("9 дней после сегодняшнего дня будет: " + today.plusDays(9));
System.out.println("3 недели после сегодняшнего дня будет: " + today.plusWeeks(3));
System.out.println("20 месяцев после сегодняшнего дня будет: " + today.plusMonths(20));
System.out.println("9 дней до сегодняшнего дня будет: " + today.minusDays(9));
System.out.println("3 недели до сегодняшнего дня будет: " + today.minusWeeks(3));
System.out.println("20 месяцев до сегодняшнего дня будет: " + today.minusMonths(20));
// А теперь поиграемся с датой
```

System.out.println("**Первый день этого месяца :** " +
today.with(TemporalAdjusters.firstDayOfMonth()));

LocalDate lastDayOfYear = today.with(TemporalAdjusters.lastDayOfYear());

System.out.println("**Последний день этого года :** " + lastDayOfYear);

Period period = today.until(lastDayOfYear);

System.out.println("Находим время между двумя датами : « + period);

System.out.println("**В этом году осталось** " + period.getMonths() + " **месяц(ев)**");

Парсинг и форматирование даты

Пример:

LocalDate date = LocalDate.now();

// стандартный формат даты

System.out.println("стандартный формат даты для LocalDate : " + date);

// применяем **свой формат даты**

System.out.println(date.format(DateTimeFormatter.ofPattern("d::MMM::uuuu")));

System.out.println(date.format(DateTimeFormatter.BASIC_ISO_DATE));

LocalDateTime dateTime = LocalDateTime.now();

//**стандартный формат даты и времени**

System.out.println("стандартный формат даты LocalDateTime : " + dateTime);

//применяем **свой формат даты и времени**

System.out.println(dateTime.format(DateTimeFormatter.ofPattern("d::MMM::uuu u HH::mm::ss")));

System.out.println(dateTime.format(DateTimeFormatter.BASIC_ISO_DATE));

Instant timestamp = Instant.now();

//стандартный формат даты

System.out.println("стандартный формат : " + timestamp);