The `ZonedDateTime` class represents a date and a time with time zone information. While creating an instance of ZonedDateTime, we need to provide a ZoneId. The `ZoneId` is an identifier used to represent different zones. Before we proceed towards `ZonedDateTime`, let's look at ZoneId briefly.

The below example shows how to get a ZoneId for a given Zone.

```
import java.time.ZoneId;
import java.util.Set;

class DateTimeDemo {
    public static void main(String args[]) {

        //Fetching the Zoneid for given Zone.
        ZoneId zoneId = ZoneId.of("America/Marigot");
        System.out.println("Zone Id " + zoneId);

        //Fetching a Set of all Zoneids
        Set<String> zoneIdList = ZoneId.getAvailableZoneIds();

        for (String zone : zoneIdList) {
            System.out.println(zone);
        }
    }
}
```

Run Save Reset

1) Creating a `ZonedDateTime` instance

We can create a `ZonedDateTime` instance using the `now()` or `of()` methods.

Below is an example, to show how to create a ZonedDateTime object.

```
import java.time.ZoneId;
import java.time.ZonedDateTime;

class DateTimeDemo {
    public static void main(String args[]) {

        // Fetching the current TimeZone
        ZonedDateTime zonedDateTime = ZonedDateTime.now();
        System.out.println(zonedDateTime);
        // fetching the ZoneId for Canada/Atlantic
        ZoneId zoneId = ZoneId.of("Canada/Atlantic");

        zonedDateTime =
                ZonedDateTime.of(2020, 10, 15, 23, 45, 59, 1234, zoneId);
        System.out.println(zonedDateTime);
    }
}
```

Run Save Reset

2) Fetching Date and Time of a ZonedDateTime

We can fetch the date and time fields of a `ZonedDateTime` instance using one of the following methods:

- `getYear()`
- `getMonth()`
- `getDayOfMonth()`
- `getDayOfWeek()`
- `getDayOfYear()`
- `getHour()`
- `getMinute()`
- `getSecond()`
- `getNano()`

The example below shows the usage of all these methods.

```
import java.time.DayOfWeek;
import java.time.Month;
import java.time.ZonedDateTime;

class DateTimeDemo {
    public static void main(String args[]) {

        ZonedDateTime zonedDateTime = ZonedDateTime.now();

        int year = zonedDateTime.getYear();
        System.out.println("Year is: " + year);

        Month month = zonedDateTime.getMonth();
        System.out.println("Month is: " + year);

        int dayOfMonth = zonedDateTime.getDayOfMonth();
        System.out.println("Day Of Month is: " + dayOfMonth);

        DayOfWeek dayOfWeek = zonedDateTime.getDayOfWeek();
        System.out.println("Day of week is: " + dayOfWeek);

        int dayOfYear = zonedDateTime.getDayOfYear();
        System.out.println("Day of year is: " + dayOfYear);

        int hour = zonedDateTime.getHour();
        System.out.println("Hour is: " + hour);

        int minute = zonedDateTime.getMinute();
```

Run Save Reset

3) Modifying date and time.

The `ZonedDateTime` class contains a set of methods used for modifying the date and time. Some of these methods are:

- `plusYears()`
- `plusMonths()`
- `plusDays()`
- `plusHours()`
- `plusMinutes()`
- `plusSeconds()`
- `plusNanos()`
- `minusYears()`
- `minusMonths()`
- `minusDays()`
- `minusHours()`
- `minusMinutes()`
- `minusSeconds()`
- `minusNanos()`

The example given below shows the usage of all these methods.

```
import java.time.ZonedDateTime;

class DateTimeDemo {

    public static void main(String args[]) {

        ZonedDateTime zonedDateTime = ZonedDateTime.now();

        System.out.println("Date after adding Year is: " + zonedDateTime.plusYears(1));

        System.out.println("Date after adding Month is: " + zonedDateTime.plusMonths(1));

        System.out.println("Date after adding days is: " + zonedDateTime.plusDays(15));

        System.out.println("Date after adding hours is: " + zonedDateTime.plusHours(15));

        System.out.println("Date after adding minutes is: " + zonedDateTime.plusMinutes(1));

        System.out.println("Date after adding seconds is: " + zonedDateTime.plusSeconds(15));

        System.out.println("Date after adding nanoseconds is: " + zonedDateTime.plusNanos(15));

        System.out.println("Date after subtracting Year is: " + zonedDateTime.minusYears(1));

        System.out.println("Date after subtractng Month is: " + zonedDateTime.minusMonths(1));

        System.out.println("Date after subtracting days is: " + zonedDateTime.minusDays(15));
```

Run Save Reset