URBAN EDGE

Advanced Java - 1

Collections Framework

A collection represents a group of object.
Java Collections provide Classes and Interfaces for us to be able to write code quickly and efficiently

Why do we need Collections

We need Collections for efficient storage and better manipulation of data in Java.
For ex: We use arrays to store integers but what if we want to

→ Resize this array?
→ Insert an element in between?
→ Delete an element in Array?
→ Apply certain operations to change this array?

How are collections available

Collections in Java are available as Classes and Interfaces. Following are few commonly used Collections in Java:

* ArrayList → For variable size Collection
* Set → For distinct collection
* Stack → A LIFO data structure
* HashMap → For storing key-value pairs

Collection class is available in java.util package
Collection class also provides static methods for sorting, searching etc.

URBAN EDGE

Date & Time in Java

Java.time → package for Date & time in Java
↳ from Java 8 onwards

Before java 8, java.util package used to hold the date and time classes. Now these classes are deprecated

How java stores a Date?

Date in Java is stored in the form of a long numer. This long number holds the number of milli seconds passed since 1 Jan 1970

Java assumes that 1900 is the start year which means it calculates years passed since 1900 whenever we ask it for years passed.

System.currentTimeMillis() returns no of milli seconds passed. Once no of ms are calculated, we can calculate minutes, seconds & years passed

Quick Quiz: Is it safe to store the no of ms in a variable of type long?

The Date class in Java

```
Date d = new Date();
System.out.println(d);
```

We can also use constructors provided by the Date class

Java Date class has few methods which can be used. For ex: getDate(), getDay() etc.

All these methods are depreciated

Calendar Class in Java
Calendar is an abstract class that provides calendar related methods in Java

Calendar.getInstance → returns a Calendar instance based on current time

```
Calendar a = Calendar.getInstance();
a.getTime() → prints time
```

Calendar class methods

1. get method is used to get year, date, min, second

```
a.get(Calendar.SECOND)
a.get(Calendar.MINUTE)
a.get(Calendar.DATE)
a.get(Calendar.YEAR)
```

In Calendar class get and set methods are very important.

2. getTime method returns a Date object

3. Other methods can be looked up from the Java docs!

Gregorian Calendar Class
This class is used to create an instance of gregorian calendar
We can change the year month & date using Set method!

URBAN EDGE

Time Zone

TimeZone class is used to create Time Zones in Java. Some of the important methods of TimeZone class are :

- getAvailableIDs() → get all the available IDs supported
- getDefault() → get the default timezone
- getID() → get the ID of a TimeZone

Always use java.time API for date and time. as it easy and provides many methods for advanced formatting of date and time.

Java.time package

- → Available from Java 8 onwards
- → Capable of storing even nano seconds

Following as some of the most commonly used classes from java.time package.

- LocalDate → Represents a Date
- LocalTime → Represents a Time --> It also gives nanoseconds
- LocalDateTime → Represents a Date + Time -->
- DateTimeFormatter → Formatter for displaying & parsing date-time objects

This is the way to instantiate these classes

```
LocalDate d = LocalDate.now();
sout(d);
```

--> It gives current date

same for LocalTime

DateTimeFormatter : -

```
import java.time.format.DateTimeFormatter;

LocalDateTime dt = LocalDateTime.now();

DateTimeFormatter df = DateTimeFormatter.ofPattern("");
;
String mydate = dt.format(df)
```

-----> give your pattern under quotes

-----> Prints date in your defined format

use java docs for more info and formats.