

## Bài 5

# CALENDAR – DATE – ENUM

Company: DEVPRO VIỆT NAM

Website: [devpro.edu.vn](http://devpro.edu.vn)

Design by Minh An

1

## Nội dung

- Calendar
- Date
- Enum

Design by Minh An

2

## Calendar

java.util.Calendar

- Lớp Calendar cung cấp các phương thức tương tác với thời gian:
  - Lấy thời gian hiện tại của hệ thống
  - Thực hiện các phép toán với ngày và giờ.
  - ...
- Khởi tạo một Calendar:

```
Calendar cal = Calendar.getInstance();
```

Design by Minh An

3

## Các thuộc tính của Calendar

`get(Calendar.DAY_OF_WEEK)` - 1 (Calendar.SUNDAY) tới 7 (Calendar.SATURDAY).

`get(Calendar.YEAR)` - Năm (year)

`get(Calendar.MONTH)` → (Calendar.JANUARY) tới 11 (Calendar.DECEMBER).

`get(Calendar.DAY_OF_MONTH)` - 1 tới 31

`get(Calendar.DATE)` - 1 tới 31

`get(Calendar.HOUR_OF_DAY)` - 0 tới 23

`get(Calendar.MINUTE)` - 0 tới 59

`get(Calendar.SECOND)` - 0 tới 59

`get(Calendar.MILLISECOND)` - 0 tới 999

`get(Calendar.HOUR)` - 0 tới 11, được sử dụng cùng với `Calendar.AM_PM`.

Design by Minh An

4

## Các thuộc tính của Calendar

`get(Calendar.AM_PM)` → 0 (Calendar.AM) hoặc 1 (Calendar.PM).

`get(Calendar.DAY_OF_WEEK_IN_MONTH)` → `DAY_OF_MONTH` 1 tới 7 luôn luôn tương ứng với `DAY_OF_WEEK_IN_MONTH` 1; 8 tới 14 tương ứng với `DAY_OF_WEEK_IN_MONTH` 2, ...

`get(Calendar.DAY_OF_YEAR)` → 1 tới 366

`get(Calendar.ZONE_OFFSET)` → Giá trị GMT của múi giờ.

`get(Calendar.ERA)` → Biểu thị `AD` (GregorianCalendar.AM), `BC` (GregorianCalendar.BC).

Design by Minh An

5

## Các thuộc tính của Calendar

Ví dụ:

```
Calendar c = Calendar.getInstance();
int year = c.get(Calendar.YEAR);

// Trả về giá trị từ 0 - 11
int month = c.get(Calendar.MONTH);
int day = c.get(Calendar.DAY_OF_MONTH);
int hour = c.get(Calendar.HOUR_OF_DAY);
int minute = c.get(Calendar.MINUTE);
int second = c.get(Calendar.SECOND);
int millis = c.get(Calendar.MILLISECOND);
```

Design by Minh An

6

## Các phương thức của Calendar

```
void set(int calendarField, int value)
void set(int year, int month, int date)
void set(int year, int month, int day, int hour,
int minute, int second)
    // Thêm hoặc trừ một khoảng thời gian trên một
trường (field) của Calendar.
// Dựa trên quy tắc của bộ Lịch.
void add(int field, int amount)
    // Cuộn (lên/xuống) một trường (field) của
Calendar.
// roll(): Không làm ảnh hưởng tới các trường
khác của Calendar.
void roll(int calendarField, boolean up)
```

Design by Minh An

7

## Các phương thức của Calendar

```
// Cuộn lên (roll up) một trường của Calendar.
// roll(): Không làm ảnh hưởng tới các trường
khác của Calendar.
void roll(int calendarField, int amount)
// Trả về đối tượng Date dựa trên giá trị của
Calendar.
Date getTime()
void setTime(Date date)

// Trả về số mili giây của đối tượng Calendar
này.
long getTimeInMillis()
void setTimeInMillis(long millis)
void setTimeZone(TimeZone value)
```

Design by Minh An

8

## Các phương thức của Calendar

```
cal.set(Calendar.DAY_OF_MONTH, 9);
cal.set(Calendar.MONTH, Calendar.DECEMBER);
cal.set(Calendar.YEAR, 2018);
cal.set(Calendar.HOUR_OF_DAY, 10);
cal.set(Calendar.MINUTE, 59);
cal.set(Calendar.SECOND, 59);

// hiển thị thời gian hiện tại sau khi thay đổi
System.out.println(cal.getTime());

// hoặc chúng ta có thể thay đổi tất cả như sau
cal.set(2016, Calendar.APRIL, 30, 20, 23, 8);
System.out.println(cal.getTime());
```

Design by Minh An

9

## TimeZone

- Lớp TimeZone trong java được sử dụng để biểu diễn múi giờ. TimeZone là một lớp trừu tượng (abstract) kế thừa lớp Object và implements các giao diện Serializable, Cloneable.:

Phương thức	Mô tả
static String[] getAvailableIDs()	Nó được sử dụng để nhận được tất cả các ID có sẵn được hỗ trợ.
static TimeZone getDefault()	Nó được sử dụng để lấy TimeZone mặc định cho máy chủ hiện tại.
String getDisplayName()	Nó được sử dụng để trả về tên của múi giờ phù hợp để trình bày cho người dùng ở vùng miền mặc định.
String getID()	Nó được sử dụng để lấy ID của múi giờ hiện tại.
int getOffset(long date)	Nó được sử dụng để trả về offset của múi giờ này từ UTC tại một date nhất định.
void setID(String ID)	Nó được sử dụng để thiết lập ID cho múi giờ hiện tại.

Design by Minh An

10

## TimeZone

Ví dụ

```
String[] id = TimeZone.getAvailableIDs();
System.out.println("Danh sach id co san la: ");
for (int i = 0; i < id.length; i++){
    System.out.println(id[i]);
}

TimeZone zone = TimeZone.getTimeZone("Asia/Ho_Chi_Minh");
System.out.println("Offset: "+zone.getOffset(Calendar.ZONE_OFFSET));

String name = zone.getDisplayName();
System.out.println("Ten hien thi mac dinh cua mui gio: " + name);

TimeZone timezone = TimeZone.getTimeZone("Asia/Ho_Chi_Minh");
System.out.println("Gia tri cua ID la: " + timezone.getID());
```

Design by Minh An

11

## Date

`java.util.Date`

- Một lớp đại diện cho ngày tháng năm và thời gian. Tiếc là hầu hết các phương thức của nó đã bị lỗi thời, khuyến cáo là không nên sử dụng các phương thức đó, tuy nhiên lớp Date vẫn được sử dụng rộng rãi.

// Tạo một đối tượng Date mô tả thời điểm hiện tại.

```
Date date1 = new Date();
```

// Tạo một đối tượng Date theo thời gian là milli giây tính từ // 1-1-1970.

```
long millis = .....
```

```
Date date2 = new Date(millis);
```

Design by Minh An

12

## Date

Một số phương thức:

`compareTo(Date date)`

// So sánh 2 đối tượng date1 và date2.

//  $i < 0$  nghĩa là  $\text{date1} < \text{date2}$

//  $i = 0$  nghĩa là  $\text{date1} = \text{date2}$

//  $i > 0$  nghĩa là  $\text{date1} > \text{date2}$

`before(Date date)` - true / false - kiểm tra xem ngày này có phải là trước ngày kia không

`after(Date date)` - true / false - sau 1 ngày có sau 1 ngày không

Design by Minh An

13

## SimpleDateFormat

### Date → String

```
Date date = new Date();
```

```
DateFormat df = new SimpleDateFormat("dd-MM-yyyy HH:mm:ss");
```

```
String dateString = df.format(date);
```

### String → Date

```
String dateString = "23/04/2005 23:11:59";
```

```
DateFormat df = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");
```

```
Date date = df.parse(dateString);
```

Design by Minh An

14

## Enum

Enum trong java là một kiểu dữ liệu đặc biệt của Java được sử dụng để định nghĩa các tập hợp các hằng số. Cụ thể hơn, Java enum là một kiểu đặc biệt của lớp trong java. Một enum có thể chứa các trường, phương thức và Constructor.

```
public class EnumExample1 {  
    //define enum  
    enum Season {SPRING, SUMMER, FALL, WINTER;}  
    public static void main(String[] args) {  
        Season season = Season.WINTER;  
        System.out.println(season);  
    }  
}
```

Design by Minh An

15

## Enum

Định nghĩa enum bên ngoài class.

```
//define enum  
enum Season {SPRING, SUMMER, FALL, WINTER;}  
public class EnumExample2 {  
  
    public static void main(String[] args) {  
        for (Season season : Season.values())  
            System.out.println(season);  
    }  
}
```

Design by Minh An

16