



11-2. java.util 패키지

혼자 공부하는 자바 (신용권 저)

목차

- 시작하기 전에
- Date 클래스
- Calendar 클래스
- 키워드로 끝내는 핵심 포인트
- 확인문제

시작하기 전에

[핵심 키워드] : Date 클래스, Calendar 클래스

[핵심 포인트]

java.util 패키지는 프로그램 개발에서 자주 사용되는 자료구조일 뿐만 아니라, 날짜 정보를 제공하는 유용한 API를 포함하고 있다. 날짜 정보를 제공하는 API에 대해 알아본다.

시작하기 전에

- ❖ 날짜, 시간, 리스트, 벡터, 해시 테이블, 컬렉션 등 다양한 유틸리티 클래스와 인터페이스를 제공
- ❖ java.util 패키지가 제공하는 주요 클래스

클래스	설명
Arrays	배열을 비교, 복사, 정렬 등 조작할 때 사용한다.
Calendar	날짜와 시간 정보가 필요할 때 사용한다.
Date	밀리초 단위의 현재 시각이 필요할 때 사용한다.
StringTokenizer	특정 문자로 구분된 문자열을 뽑아낼 때 사용한다.
Random	난수가 필요할 때 사용한다.

Arrays 클래스

❖ Arrays 클래스가 제공하는 주요 정적 메서드

메서드	설명
List asList(배열)	배열을 리스트로 변환한다.
int binarySearch(배열, 키)	배열에서 키 값이 있는 인덱스를 반환한다.
배열 copyOf(배열, 길이)	원본 배열을 길이만큼 복사한다.
배열 copyOfRange(배열, 시작, 끝)	원본 배열을 지정한 영역만큼 복사한다.
boolean equals(배열, 배열)	두 배열의 동일 여부를 비교한다.
void fill(배열, 값)	배열을 지정된 값으로 저장한다.
void fill(배열, 시작, 끝, 값)	배열의 지정된 영역에 지정된 값을 저장한다.
void sort(배열)	배열을 오름차순으로 정렬한다.

Arrays 클래스

```
1 package sec03;
2
3 import java.util.Arrays;
4
5 public class ArraysDemo {
6     public static void main(String[] args) {
7         char[] a1 = { 'J', 'a', 'v', 'a' };
8         char[] a2 = Arrays.copyOf(a1, a1.length);
9         System.out.println(a2);
10
11         String[] sa = { "케이크", "애플", "도넛", "바나나" };
12         print(sa);
13
14         Arrays.sort(sa);
15         print(sa);
16
17         System.out.println(Arrays.binarySearch(sa, "애플"));
18
19         Arrays.fill(sa, 2, 4, "기타");
20         print(sa);
21     }
22
23     static void print(Object[] oa) {
24         for (Object o : oa)
25             System.out.print(o + " ");
26         System.out.println();
27     }
28 }
```

Date 클래스

❖ Date 클래스

- 날짜를 표현하는 클래스
- Date는 객체 간 날짜 정보 주고받을 때 매개 변수나 리턴 타입으로 주로 사용

```
Date now = new Date();
```

- 원하는 날짜 형식의 문자열 얻기 위해 java.text 패키지의 SimpleDateFormat 클래스와 함께 사용

```
SimpleDateFormat sdf = new SimpleDateFormat("yyyy년 MM월 dd일 hh시 mm분 ss초");
```

- format() 메소드 호출

```
String strNow = sdf.format(now);
```

Date 클래스

❖ 예시 - 현재 날짜 출력하기

```
01 package sec02.exam01;
02
03 import java.text.*;
04 import java.util.*;
05
06 public class DateExample {
07     public static void main(String[] args) {
08         Date now = new Date();
09         String strNow1 = now.toString();
10         System.out.println(strNow1);
11
12         SimpleDateFormat sdf =
13             new SimpleDateFormat("yyyy년 MM월 dd일 hh시 mm분 ss초");
14         String strNow2 = sdf.format(now);
15         System.out.println(strNow2);
16     }
17 }
```

실행결과

Mon Mar 11 17:19:08 KST 2024
2024년 03월 11일 05시 19분 08초

Calendar 클래스

❖ Calendar 클래스

- 추상 클래스이므로 new 연산자 사용하여 인스턴스 생성 불가
- getInstance() 메소드 이용하여 현재 운영체제에 설정된 시간대 기준으로 한 Calendar 하위 객체 얻을 수 있음

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

```
int year    = now.get(Calendar.YEAR);        //연도를 리턴
int month   = now.get(Calendar.MONTH) + 1;    //월을 리턴
int day     = now.get(Calendar.DAY_OF_MONTH); //일을 리턴
int week    = now.get(Calendar.DAY_OF_WEEK);  //요일을 리턴
int amPm    = now.get(Calendar.AM_PM);        //오전/오후를 리턴
int hour    = now.get(Calendar.HOUR);         //시를 리턴
int minute  = now.get(Calendar.MINUTE);       //분을 리턴
int second  = now.get(Calendar.SECOND);       //초를 리턴
```

Calendar 클래스

■ 예시 - 운영체제의 시간대 기준으로 Calendar 얻기

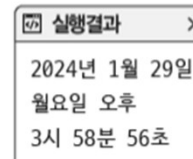
```
01 package sec02.exam02;
02
03 import java.util.*;
04
05 public class CalendarExample {
06     public static void main(String[] args) {
07         Calendar now = Calendar.getInstance();
08
09         int year = now.get(Calendar.YEAR);
10         int month = now.get(Calendar.MONTH) + 1;
11         int day = now.get(Calendar.DAY_OF_MONTH);
12
13         int week = now.get(Calendar.DAY_OF_WEEK);
14         String strWeek = null;
15         switch(week) {
16             case Calendar.MONDAY:
17                 strWeek = "월";
18                 break;
19             case Calendar.TUESDAY:
20                 strWeek = "화";
```

Calendar 클래스

```
21         break;
22     case Calendar.WEDNESDAY:
23         strWeek = "수";
24         break;
25     case Calendar.THURSDAY:
26         strWeek = "목";
27         break;
28     case Calendar.FRIDAY:
29         strWeek = "금";
30         break;
31     case Calendar.SATURDAY:
32         strWeek = "토";
33         break;
34     default:
35         strWeek = "일";
36     }
37
38     int amPm = now.get(Calendar.AM_PM);
39     String strAmPm = null;
```

Calendar 클래스

```
40     if(amPm == Calendar.AM) {
41         strAmPm = "오전";
42     } else {
43         strAmPm = "오후";
44     }
45
46     int hour = now.get(Calendar.HOUR);
47     int minute = now.get(Calendar.MINUTE);
48     int second = now.get(Calendar.SECOND);
49
50     System.out.print(year + "년 ");
51     System.out.print(month + "월 ");
52     System.out.println(day + "일 ");
53     System.out.print(strWeek + "요일 ");
54     System.out.println(strAmPm + " ");
55     System.out.print(hour + "시 ");
56     System.out.print(minute + "분 ");
57     System.out.println(second + "초 ");
58 }
59 }
```



실행결과

2024년 1월 29일
월요일 오후
3시 58분 56초

StringTokenizer 클래스

- ❖ 문자열을 토큰으로 분리하는 데 사용
- ❖ 토큰은 공백이나 줄 바꿈 등 구분자를 사용해 문자열을 분리
- ❖ StringTokenizer 클래스의 주요 생성자

생성자	설명
<code>StringTokenizer(String s)</code>	주어진 문자열을 기본 구분자로 파싱한 StringTokenizer 객체를 생성한다.
<code>StringTokenizer(String s, String delim)</code>	주어진 문자열을 delim 구분자로 파싱한 StringTokenizer 객체를 생성한다.

- ❖ 기본 구분자는 공백, 탭, 줄 바꿈, 복귀, 용지 먹임 문자

StringTokenizer 클래스

❖ StringTokenizer 클래스가 제공하는 주요 메서드

메서드	설명
int countTokens()	남아 있는 토큰의 개수를 반환한다.
boolean hasMoreTokens()	남아 있는 토큰이 있는지 여부를 반환한다.
String nextToken()	다음 토큰을 꺼내 온다.

StringTokenizer 클래스

```
3 import java.util.StringTokenizer;
4
5 public class StringTokenizerDemo {
6     public static void main(String[] args) {
7         String s = "of the people, by the people, for the people";
8
9         StringTokenizer st = new StringTokenizer(s, " ,");
10
11         System.out.println(st.countTokens());
12
13         while (st.hasMoreTokens()) {
14             System.out.print "[" + st.nextToken() + " ] ");
15         }
16     }
17 }
```

Random 클래스

생성자	설명
Random()	Random 객체를 생성한다.
Random(long seed)	주어진 시드를 사용하는 Random 객체를 생성한다.

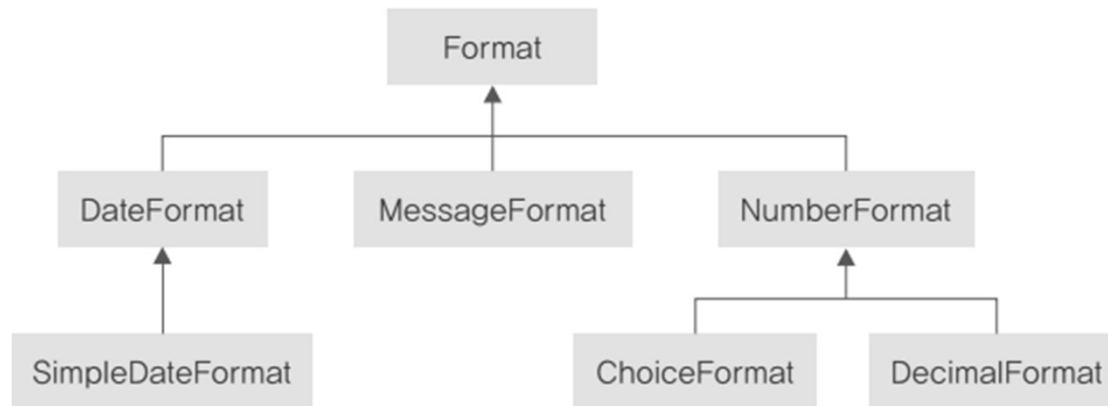
메서드	설명
boolean nextBoolean()	논리 타입 난수를 발생시킨다.
double nextDouble()	0.0 이상 1.0 미만의 double 타입 난수를 발생시킨다.
float nextFloat()	0.0 이상 1.0 미만의 float 타입 난수를 발생시킨다.
double nextGaussian()	평균, 표준편차가 0.0 및 1.0인 정규분포 난수를 발생시킨다.
int nextInt()	int 타입의 난수를 발생시킨다.
int nextInt(int n)	0~(n-1) 사이의 int 타입 난수를 발생시킨다.
long nextLong()	long 타입의 난수를 발생시킨다.
void setSeed(long seed)	시드 값을 설정한다.

Random 클래스

```
3 import java.util.Random;
4
5 public class RandomDemo {
6     public static void main(String[] args) {
7         Random r = new Random();
8
9         for (int i = 0; i < 5; i++)
10             System.out.print(r.nextInt(100) + " ");
11     }
12 }
```

java.text 패키지

- ❖ 현지화가 필요한 데이터의 효율적 처리를 위한 패키지
- ❖ 특히 패키지의 Format 클래스는 지역에 민감한 데이터를 현장에 맞게 문자열로 표현하고 포맷할 수 있도록 지원
- ❖ Format 클래스의 계층 구조



SimpleDateFormat 클래스

- ❖ 날짜 정보를 현지화하는 클래스로 날짜를 텍스트로 포맷하거나 텍스트를 날짜로 파싱
- ❖ SimpleDateFormat 클래스에서 사용할 수 있는 패턴 기호

패턴 기호	설명	패턴 기호	설명
y	연	h	시(1~12)
M	월	H	시(0~23)
w	월 구분 없는 주	k	시(1~24)
W	주	K	시(0~11)
d	일	m	분
D	월 구분 없는 일	s	초
E	요일	S	밀리초
a	오전과 오후	z	타임존

SimpleDateFormat 클래스

❖ 사용 방법

```
SimpleDateFormat f = new SimpleDateFormat("패턴");  
String s = f.format(new Date());
```

```
SimpleDateFormat f = new SimpleDateFormat("패턴");  
Date d = f.parse("날짜 문자열");
```

SimpleDateFormat 클래스

```
3 import java.text.ParseException;
4 import java.text.SimpleDateFormat;
5 import java.util.Calendar;
6 import java.util.Date;
7 import java.util.Locale;
8
9 public class SimpleDateFormatDemo {
10     public static void main(String[] args) {
11         Date d = new Date();
12
13         SimpleDateFormat sdf1 = new SimpleDateFormat("E yyyy.MM.dd 'at' hh:mm:ss a z");
14         System.out.println(sdf1.format(d));
15
16         SimpleDateFormat sdf2 = new SimpleDateFormat("yyyy-MM-dd");
17
18         try {
19             d = sdf2.parse("2023-02-28");
20         } catch (ParseException e) {
21         }
22
23         System.out.println(sdf1.format(d));
24
25         Calendar calendar = Calendar.getInstance();
26
27         SimpleDateFormat sdf3 = new SimpleDateFormat("E yyyy.MM.dd", Locale.US);
28         System.out.println(sdf3.format(calendar.getTime()));
29
30         SimpleDateFormat sdf4 = new SimpleDateFormat("E yyyy.MM.dd", Locale.KOREA);
31         System.out.println(sdf4.format(calendar.getTime()));
32     }
33 }
```

MessageFormat 클래스

- ❖ 문자열을 특정 포맷에 맞추어 깔끔하게 처리하는 클래스
- ❖ MessageFormat 객체 생성

```
MessageFormat(String pattern)
MessageFormat(String pattern, Locale locale)
static String format(String pattern, Object... arguments)
```

- ❖ 사용 방법

```
MessageFormat.format("{0} * {1} = {2}", 3, 4, 3 * 4);
```



```
MessageFormat.format("{0} * {1} = {2}", new Object[]{ 3, 4, 3 * 4 });
```

MessageFormat 클래스

```
3 import java.text.MessageFormat;
4
5 public class MessageFormatDemo {
6     public static void main(String[] args) {
7         String java = "Java";
8         int version = 8;
9
10        String s = MessageFormat.format("language : {1}\nversion : {0}", version, java);
11
12        System.out.println(s);
13
14        Object[] data = { java, version };
15
16        MessageFormat f = new MessageFormat("language : {0}\nversion : {1}");
17
18        System.out.println(f.format(data));
19    }
20 }
```

DecimalFormat 클래스

- ❖ 정수, 실수를 포함한 다양한 종류의 수를 과학적 표기, 퍼센트 표시, 화폐 표시 등으로 포맷 지원
- ❖ DecimalFormat 클래스에서 사용할 수 있는 패턴 기호

패턴 기호	설명	사용 예	1234567.890의 반환 값
#	10진수	#	12345678
0	선행 제로 10진수	0000000000.00	01234567.89
.	소수점	#,000	1234567.890
,	구분자	#,###,##	1,234,567.89
+ 또는 -	양수 또는 음수	-#.0	-1234567.9
E	지수	###E00	1.23E06
;	패턴 구분	+#,##;-#.#	+1234567.9
%	백분율	#,00%	123456789.00%

- ❖ 사용 방법

```
DecimalFormat f = new DecimalFormat("패턴");  
String s = f.format(숫자);
```


DecimalFormat 클래스

```
3 import java.text.DecimalFormat;
4
5 public class DecimalFormatDemo {
6     public static void main(String[] args) {
7         DecimalFormat f1 = new DecimalFormat("#");
8         DecimalFormat f2 = new DecimalFormat("0000000000.00");
9         DecimalFormat f3 = new DecimalFormat("#.000");
10        DecimalFormat f4 = new DecimalFormat("#,###.##");
11        DecimalFormat f5 = new DecimalFormat("-#.##");
12        DecimalFormat f6 = new DecimalFormat("#.##E00");
13        DecimalFormat f7 = new DecimalFormat("+#.##;-#.##");
14        DecimalFormat f8 = new DecimalFormat("#.00%");
15
16        System.out.println(f1.format(1234567.890));
17        System.out.println(f2.format(1234567.890));
18        System.out.println(f3.format(1234567.890));
19        System.out.println(f4.format(1234567.890));
20        System.out.println(f5.format(1234567.890));
21        System.out.println(f6.format(1234567.890));
22        System.out.println(f7.format(1234567.890));
23        System.out.println(f7.format(-1234567.890));
24        System.out.println(f8.format(1234567.890));
25    }
26 }
```



키워드로 끝내는 핵심 포인트

- Date 클래스 : 날짜를 표현하는 클래스, 객체 간 날짜 정보 주고받을 때 매개 변수나 리턴 타입으로 주로 사용.
- Calendar 클래스 : 달력을 표현하는 클래스. 추상 클래스이므로 new 연산자 사용한 인스턴스 생성이 불가능. Calendar 클래스의 정적 메소드인 getInstance() 메소드 이용하여 현재 운영체제에 설정된 시간대를 기준으로 한 Calendar 하위 객체 얻을 수 있음

확인문제

- ❖ Date와 SimpleDateFormat 클래스를 이용해서 오늘의 날짜를 다음과 같이 출력하는 프로그램을 작성해보세요

2024년 05월 08일 수요일 10시 30분

- ❖ Calendar 클래스를 이용해서 1번과 동일한 실행결과를 출력하는 프로그램을 작성해보세요.



Thank You!