JAVA Programming

Practice 10: Utility classes

BeomSeok Kim
Department of Computer Engineering
KyungHee University
passion0822@khu.ac.kr

Random class

- Random
  - ✓ 다양한 데이터 타입에서 난수를 발생시키는 클래스
    - ➢ 난수는 예상할 수 없는 입력 상태를 시뮬레이션할 때 많이 사용

[표 10-1] Random 클래스의 주요 생성자와 메소드

생성자	Random()	새로운 난수 generator를 생성한다.
	Random(long seed)	long형의 배정을 사용해 난수 generator를 생성한다.
메소드	boolean nextBoolean()	난수 generator 순서에서 균등하게 분포된 boolean형의 다음 난수를 반환한다.
	double nextDouble()	난수 generator 순서에서 균등하게 분포된 0.0에서 1.0 사이 double형의 다음 난수를 반환한다.
	float nextFloat()	난수 generator 순서에서 균등하게 분포된 0.0에서 1.0 사이 float형의 다음 난수를 반환한다.
	int nextInt()	난수 generator 순서에서 균등하게 분포된 int형의 다음 난수를 반환한다.
	long nextLong()	난수 generator 순서에서 균등하게 분포된 long형의 다음 난수를 반환한다.
	void setSeed(long seed)	단일 long형의 배정을 사용해 난수 generator의 배정을 설정한다.

[실습 10-1] 기초 1

- Random 클래스를 이용하여 난수 생성
  - ✓ 0~100 사이의 10개의 난수를 화면에 출력
  - ✓ 난수는 비슷한 숫자가 나오는 것을 방지하기 위해 매번 seed를 다른값으로 초기화

Random 구현 예

```
Random r = new Random();
r.setSeed(r.nextLong);
r.nextInt();
```

Date class

- Date
  - ✓ 날짜와 시간 정보를 쉽게 얻을 수 있는 방법을 제공
  - ✓ 시간이나 날짜는 프로그램을 개발할 때 가장 많이 필요한 요소임
    - ➢ 날짜와 시간을 자유자재로 사용할 수 있어야 한다.

[표 10-2] Date 클래스의 주요 메소드

메소드	설명
boolean after(Date when)	when 날짜 정보보다 이후인지 확인하는 진리값을 반환한다.
boolean before(Date when)	when 날짜 정보보다 이전인지 확인하는 진리값을 반환한다.
Object clone()	Date 객체를 복사하여 반환한다.
int compareTo(Date anotherDate)	anotherDate의 값과 비교한 결과를 반환한다(anotherDate 이전이면 0보다 작은 값, 같으면 0, 크면 0보다 큰 값을 반환한다).
boolean equals(Object obj)	Obj와 날짜가 동일한지 확인하는 진리값을 반환한다.

[실습 10-2] 기초 2

- Date 클래스를 이용하여 현재 시간을 출력
  - ✓ 현재 시간을 출력하기 위해 SimpleDateFormat 클래스를 이용

Date 구현 예

```
Date today = new Date();
```

SimpleDateFormat 구현 예

```
SimpleDateFormat dateForm = new SimpleDateFormat("yyyy.MM.dd hh:mm:ss");
dateForm.format(new Date());
```

Vector class

- Vector
  - ✓ Vector 크기를 동적으로 변화시킬 수 있는 일종의 배열구조
  - ✓ 한번 크기가 결정되면 변경이 불가능한 배열의 단점을 보완
    - ➢ 동적으로 확장해 사용할 수 있음
    - ➢ 배열보다는 Vector가 유리함
    - ➢ 그러나 프로그램의 응답 속도는 느려질 수 있음
      - 설정된 크기 초과시 기존 공간을 늘리지 않고 새로운 공 간을 잡은 다음 기존의 값을 복사하기 때문
      - 이때, 기존의 메모리 공간은 자동으로 비워짐
      - 더 복잡한 자료구조가 필요하다면 컬렉션 객체를 사용하는 것이 좋다.
  - ✓ Vector 클래스 내 데이터를 관리하는 다양한 메소드가 존재

Vector class

■ Vector

[표 10-4] Vector 클래스의 주요 생성자와 메소드

생성자	Vector()	내부 데이터 배열의 크기가 10이고, 수용량의 증가치가 0인 빈 벡터를 생성한다.
	Vector(int initialCapacity)	지정된 초기 수용량과 수용량의 증가치가 0인 빈 벡터를 생성한다.
	Vector(int initialCapacity, int capacityIncrement)	지정된 초기 수용량과 지정된 수용량의 증가치가 있는 빈 벡터를 생성한다.
메소드	boolean add(E e)	벡터의 마지막에 추가한다.
	void add(int index, E element)	벡터에 지정된 인덱스에 요소를 추가한다.
	void addElement(E obj)	지정된 객체를 벡터에 마지막으로 추가하고 크기를 1 늘린다.
	int capacity()	벡터의 수용량을 반환한다.
	void clear()	모든 요소를 벡터에서 삭제한다.
	boolean equals(Object o)	지정된 객체가 벡터와 동일한지 확인하는 진리값을 반환한다.
	boolean isEmpty()	벡터가 비어 있는지 확인하는 진리값을 반환한다.
	E remove(int index)	지정된 인덱스에 있는 벡터의 요소를 삭제한다.
	void setSize(int newSize)	지정된 크기만큼 벡터의 크기를 설정한다.
	void setElementAt(E obj, int index)	벡터에 지정된 인덱스의 요소에 객체를 설정한다.

[실습 10-3] 응용

- Random 클래스를 이용한 Vector 처리
  - ✓ Random 클래스를 이용하여 -100~100 사이 10개의 난수 발생
  - ✓ 각 발생시킨 난수를 Vector에 저장
  - ✓ Vector에 저장된 10개의 난수를 화면에 출력

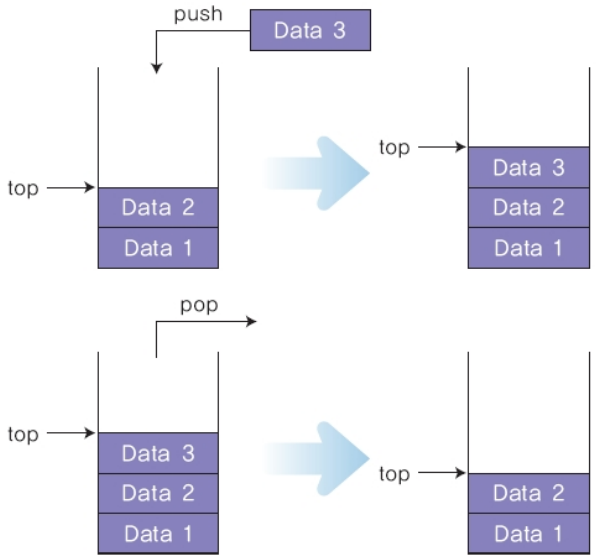
Vector 구현 예

```
Vector v = new Vector();
v.addElement(2);                    // 인덱스 0에 삽입
v.addElement(3.14);     // 인덱스 1에 삽입
v.add(1, 1);                // 인덱스 1에 삽입하고 기존 값들은 뒤로 이동

v.get(2);                           // 인덱스 2의 값 리턴
v.size();                           // Vector에 저장된 element 개수 리턴
```

Stack class

- Stack
  - ✓ 특수한 목적에서 만든 논리적 자료구조의 일종
  - ✓ 데이터를 한 방향으로만 다룰 수 있음
  - ✓ 마지막에 넣은 데이터가 먼저 나오게 되는 구조 LIFO(Last In First Out)
  - ✓ 스택에 저장된 데이터들은 배열처럼 index를 바탕으로 자료를 입력하고 출력하지만 항상 top 변수가 가리키는 위치에서 조작이 용이



[그림 10-2] Stack 클래스의 동작 원리

[실습 10-4] 기초 3

- Stack 클래스에 값을 순차적으로 입력하고 반환
  - ✓ 반복문을 이용해 3개의 임의의 수를 Stack에 Push
  - ✓ 다른 반복문을 이용해 Stack에 입력된 3개의 수를 Pop

Stack 구현 예

```
Stack st = new Stack();

st.push(10);                    // 10을 stack에 push
st.pop();                       // stack의 top 값을 리턴하고 동시에 pop
```

StringTockenizer class

- StringTokenizer
  - ✓ 문자열을 좀더 효과적으로 처리하려고 제공된 자바 클래스
    - ➢ 문자열에 있는 특정 단어나 기호를 찾는 등의 기능
    - ➢ 예) HTML 문자가 들어간 부분을 찾으려면 어떻게 해야 할까?

```
char str[ ] = "<HTML> 생각의 탄생 = 천재 </HTML>";
```

이런 작업을 토큰 개념을 이용해 쉽게 처리할 수 있도록 지원함

  - ✓ 토큰이란?
    - ➢ 입력된 문자열의 최소 단위. 문자열을 일정한 규칙으로 구분한 구분자

[실습 10-5] 기초 4

- StringTokenizer 클래스이용해 입력 문장에서 토큰을 기준으로 문자열 자르기
  - ✓ 토큰으로 콤마(,) 를 사용
  - ✓ 입력 문장
    - ➢ 자바에는 다양한 문자열 처리, 날짜 처리, 시간을 처리하는 메소드가 있습니다.

StringTokenizer 구현 예

```
String str = "입력 문장 ... ";
StringTokenizer st = new StringTokenizer(str, " ");          // str: 입력 문장, " ": 토큰

st.countTokens();                    // 토큰의 개수 리턴
st.hasMoreTokens();                  // 잔여 토큰 확인
st.nextToken();                      // 다음 토큰 리턴
```

[과제 10-1]

- 문장에서 문자열 개수와 각 문자열을 출력하는 프로그램
  - ✓ 입력 문장에서 문자열 개수를 세어 출력하고, 각 문자열을 문자열 번호와 함께 출력하는 프로그램 작성
  - ✓ 괄호, 따옴표 등의 내부는 하나의 문자열로 가정
  - ✓ 입력 문장 (ref: 한국일보, 2018.06.07)
    - ➢ 요즘 재계 분위기를 날씨에 비유하자면 '흐림'이다. 도널드 트럼프 미국 대통령이 점화한 글로벌 무역 전쟁 와중에 세탁기는 미국의 긴급수입제한조치(세이프가드), 태양광 발전 설비와 철강 등은 고율 관세의 덫에 걸렸다. 살얼음판을 걷고 있는 미・중 간 무역충돌이 거세지면 중간에 낀 우리 기업들의 피해는 더 커질 수밖에 없다. 국내에서는 지난해 인상된 법인세율이 올해부터 적용되고, 최저임금 인상으로 인한 혼란, 오는 7월 도입되는 주 최대 52시간 근무 등 첩첩산중이다. 설상가상 그동안 수출을 이끌어온 스마트폰 디스플레이 등이 휘청거리는데, 중국은 반도체 정도를 제외하면 거의 전 산업 분야에서 우리의 시장을 잠식해가고 있다. 이처럼 국내외적으로 리스크가 커져만 가는 상황에서 기업들은 '체질 개선'을 통해 미래를 준비 중이다. 4차 산업혁명의 조류에 확실히 올라타 경쟁의 판도를 바꾸겠다는 각오다.

Thank You!
Q&A