

API – java.lang 패키지



API – java.lang 패키지

* JAVA API(Application Programming Interface)

- API는 라이브러리라고 부르며 프로그램 개발에 자주 사용되는 클래스 및 인터페이스의 모음을 말합니다.
- 자바 표준 API 문서 주소: <https://docs.oracle.com/javase/8/docs/api>

* java.lang 패키지

- java.lang 패키지는 자바 프로그램의 기본적인 클래스들을 담고 있는 패키지입니다.
- 그렇기 때문에 **java.lang** 패키지에 있는 클래스와 인터페이스는 **import** 구문 없이 사용할 수 있습니다.

- java.lang 패키지 주요 클래스

1. Object: 자바 클래스의 최상위 클래스
2. System: 표준 입력장치(키보드)로부터 데이터를 입력받거나 표준 출력장치(모니터)로 출력하기 위해 사용
3. Class: 클래스를 메모리에 로딩할 때 사용
4. String: 문자열을 저장하고 문자열의 여러 가지 정보를 얻을 때 사용
5. StringBuffer, StringBuilder: 문자열을 저장하고 내부 문자열을 조작할 때 사용
6. Math: 수학 함수를 이용할 때 사용
7. Wrapper(Byte, Short, Integer, Long, Float, Double, Boolean, Character)
: 기본 데이터 타입의 객체를 만들 때 사용



Object클래스

*Object 클래스

- 클래스를 선언할 때 extends 키워드로 다른 클래스를 상속하지 않으면 묵시적으로 Object 클래스를 상속하게 됩니다. 따라서 모든 자바의 클래스는 Object 클래스의 자식이거나 자손클래스입니다.
- 모든 클래스가 Object를 상속하기 때문에 Object의 메서드는 모든 클래스에서 사용이 가능합니다.



System 클래스

*System 클래스

- System 클래스를 이용하면 운영체제의 일부 기능을 이용할 수 있습니다. 프로그램 종료, 키보드로 입력, 모니터로 출력, 메모리 정리, 현재 시간 읽기 등이 가능합니다.
- System 클래스의 모든 멤버는 static으로 구성되어 있어 클래스 이름으로 바로 접근이 가능합니다.

*System 클래스 주요 메서드

1. `exit()`: 현재 실행하고 있는 프로세스를 강제 종료시킵니다. 정상 종료일 경우 매개값으로 0을 주고, 비정상 종료인 경우 0 이외에 다른 값을 줍니다.
2. `currentTimeMillis()`
: 컴퓨터의 시계로부터 현재 시간을 읽어서 밀리세컨드($1/1000$ 초) 단위와 나노세컨드($1/10^9$ 초)단위의 long값을 리턴합니다. 주로 프로그램의 실행 소요 시간 측정으로 성능을 테스트할 때 사용합니다.
3. `getProperty()`: JVM이 시작할 때 자동 설정되는 시스템의 속성값을 구합니다.
4. `gc()`: Garbage Collector를 실행시킵니다.

String 클래스 - 문자열

* String 클래스

- String은 생성자를 사용해서 객체를 만들 필요 없이 기초 데이터를 사용할 때처럼 바로 초기화해서 사용할 수 있습니다.

- String 클래스의 주요 메서드

1. `charAt()`: 특정 인덱스의 글자를 리턴.
2. `indexOf()`: 특정 문자열의 시작 인덱스 값을 반환합니다. 만약 주어진 문자열이 포함되어 있지 않으면 -1을 리턴합니다.
3. `length()`: 문자열의 길이를 알려줍니다.
4. `replace()`: 특정 문자열을 변경합니다.
5. `substring()`: 인덱스를 기준으로 그 앞의 문자열을 잘라줍니다. 만약 매개값으로 인덱스를 2개 주면 처음 매개값 이상 두 번째 매개값 미만의 문자열을 제외하고 나머지를 삭제합니다.
6. `toUpperCase()`: 문자열을 대문자로 치환합니다.
7. `toLowerCase()`: 문자열을 소문자로 치환합니다.
8. `trim()`: 문자열의 앞, 뒤 공백을 제거합니다.
9. `equals()`: 문자열의 값을 비교합니다.
10. `valueOf()`: 기본 데이터 타입의 값들을 문자열로 변환합니다.
11. `split()`: 특정 문자열 기준으로 잘라서 배열에 리턴.



String의 단점을 보완한 StringBuilder

* String 클래스의 단점

- String 클래스는 아주 많이 쓰이고 좋은 기능들을 많이 가지고 있지만 메모리를 과소비한다는 문제점이 있습니다.
- String 객체는 처음 초기화된 데이터에 변화가 생기면 기존 객체를 재활용하지 않고 새로운 객체를 생성합니다.

* StringBuilder

-String 클래스의 단점인 새로운 객체를 생성하는 문제로 인해 속도가 느려지는 현상이 발생하기 때문에 이 문제를 해결하기 위해 속도적인 측면에서 개선된 클래스 입니다

- StringBuilder 클래스의 주요 메서드

1. append(): 기존 문자열의 뒤에 문자열을 추가합니다.
2. insert(): 특정 위치에 문자열을 추가합니다.
3. delete(): 문자열을 삭제합니다.
4. deleteCharAt(): 특정 인덱스의 문자 하나를 삭제합니다.
5. replace(): 문자열의 일부분을 다른 문자열로 대체합니다.
6. setCharAt(): 문자열에서 주어진 index의 문자를 다른 문자로 대체합니다.
7. toString(): StringBuilder 객체의 값을 문자열로 반환합니다.
8. reverse(): 문자열을 거꾸로 뒤집습니다.

StringBuilder 예시

```
public static void main(String[] args) {  
    long start, end;  
    start = System.currentTimeMillis();  
    String str = "A";  
    for(int i=0; i<300000; i++) {  
        str = str + "A";  
    }  
    end = System.currentTimeMillis();  
    System.out.println("String: " + (end-start)*0.001 + "초");  
  
    //빌더로 A 붙이기  
    start = System.currentTimeMillis();  
    StringBuilder sb = new StringBuilder("A");  
    for(int i=0; i<300000; i++) {  
        sb.append("A");  
    }  
    end = System.currentTimeMillis();  
    System.out.println("StringBuilder: " + (end-start)*0.001 + "초");  
}
```

for문 앞뒤로 시간을
측정하고
A를 300000번 붙여서 확인

실행결과
String: 34.961초
StringBuilder: 0.005초



Math클래스 - 수학관련

* Math 클래스

- Math 클래스는 수학 계산에 사용할 수 있는 메서드들을 제공합니다.
- Math 클래스가 제공하는 메서드는 모두 정적 메서드이므로 Math 클래스 이름으로 바로 사용이 가능합니다.

- Math 클래스 주요 메서드

1. `abs()`: 절대값을 구함.
2. `ceil()`: 올림값을 구함.
3. `floor()`: 내림값을 구함.
4. `max()`: 수를 비교하여 최대값을 구함.
5. `min()`: 수를 비교하여 최소값을 구함.
6. `random()`: 랜덤값을 구함($0.0 \leq \text{값} < 1.0$)
7. `rint()`: 현재 수에서 가까운 정수를 실수형태로 구함.
8. `round()`: 반올림값을 구함.



Wrapper 클래스

* Wrapper 클래스

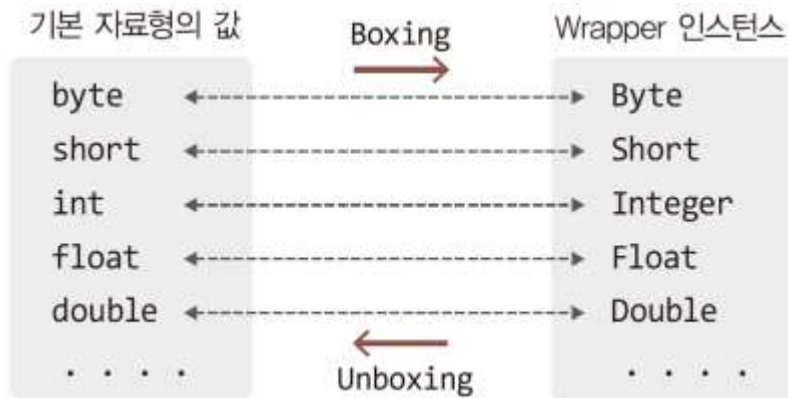
- 자바는 기본 타입의 값을 갖는 객체를 생성할 수 있습니다.
- 이러한 객체를 포장(Wrapper) 객체라고 하는데 그 이유는 기본 타입의 값을 객체 내부에 두고 포장하기 때문입니다.
- 포장 객체의 특징은 포장하고 있는 기본 타입의 값은 외부에서 변경할 수 없습니다. 만약 변경하고 싶으면 새로운 포장 객체를 생성해야 합니다.

다양한 Wrapper클래스

- Byte, Short, Integer, Long, Float, Double, Boolean, Character

Wrapper의 두가지 의미

- 기본 타입의 값을 포장 객체로 만드는 과정을 **박싱(Boxing)**이라 하며 포장 객체에서 기본 타입의 값을 얻어내는 과정을 **언박싱(Unboxing)**이라 합니다.



1.8버전 부터 AutoBoxing을 지원함

-문자열을 기본 타입 값으로 변환할 때도 많이 사용합니다. 대부분의 포장 클래스는 "**parse+기본타입**" 명으로 되어있는 static메서드를 제공합니다.

```
String s1 = "10";
String s2 = "5.43";

int a = Integer.parseInt(s1);
double b = Double.parseDouble(s2);
float c = Float.parseFloat("4.53");

System.out.println(a + b + c);
```

실제 Wrapper클래스는 문자열->기본형으로 바꾸는데 많이 사용된다
반드시 기억하자!

실행결과
19.96



Chapter 19

수고하셨습니다