

기본 API

자바에서 제공하는 API 중 가장 많이 사용 되는 클래스들로

기본적인 클래스를 가지고 있는 패키지.

java.lang 패키지에 있는 클래스들은 import 없이 바로 사용할 수 있는 클래스들이다.

앞에서 String, System 클래스를 import 하지 않고 사용 할수 있었던 이유가 바로 이것 때문이다

String new 연산자를 이용하지 않고 인스턴스를 만들 수 있다

new 연산자를 이용해서도 인스턴스를 만들 수 있다

String str1 = "hello"; hello 는 상수 영역에 하나 만들어진다

String str2 = "hello"; str2 도 같은 str1 상수 영역을 가르킨다

String str3 = new String("hello") new 가 나오게되면

새로운 영역 heap 에만든다 한번 생성된 클래스는 변하지 않으며

문자를 자르든 지지고 볶고 다해도 처음 그 클래스는 변하지 않는다

equals() 메소드와 비교연산자 == 차이점

== 비교를 위한 연산자

주소의 값을 비교한다. 주소의 값이란 실제 내용이 값이 아닌 자료의 위치의 값

equals() 메소드

equals() 메소드는 객체끼리 내용 비교한다.

equalsIgnoreCase : 대소문자 구분없이 객체내용비교할때

수학적 계산에 사용되는 클래스 Math

```
1 package com.lec.ex3_math;
2 // Math.ceil (올릴 실수) => double 값 return    ex : Math.ceil(9.1) => 10.0
3 // Math.round (반올림할 실수) => long 값 return  ex : Math.round(9.1) => 9
4 // Math.floor (버릴 실수) => double 값 return    ex : Math.floor(9.9) => 9.0
5 public class Ex02_round {
6     public static void main(String[] args) {
7         System.out.println("소수점에서 반올림, 올림, 버림"); // Math
8         System.out.println("9.15를 올림 : " + Math.ceil(9.15));
9         System.out.println("9.15를 반 올림 : " + Math.round(9.15));
10        System.out.println("9.15를 버림 : " + Math.floor(9.15));
11        System.out.println("소수점 한자리에서 반올림, 올림, 버림");
12        System.out.println("9.15를 올림 : " + Math.ceil(9.15 * 10)/10);
13        System.out.println("9.15를 올림 : " + Math.round(9.15 * 10)/10.0);
14        System.out.println("9.15를 버림 : " + Math.floor(9.15 * 10)/10);
15
16        System.out.println("일의 자리에서 반올림, 올림, ");
17        System.out.println("85를 올림 (90) : " + (int)Math.ceil(85/10.0) * 10);
18        System.out.println("85를 반올림 (90) : " + Math.round(85/10.0) * 10);
19        System.out.println("85를 내림 (80) : " + Math.floor(85/10.0) * 10);
20    }
21 }
```

Math.ceil(올릴 실수값) => double 값 리턴

Math.round(반올림할 실수값) => 정수값 리턴

Math.floor(내릴 실수값) => double 값 리턴

소수점 첫자리에서 반올림, 올림, 버림

ceil(9.15) = 9.15을 올림:9.2

$9.15 \times 10 = 91.5 / 10 = 9.15$ 에서 올림 = 9.2

round (9.15) =9.15를 반올림:9.2

$9.15 \times 10 = 91.5 / 10.0 = 9.15$ (round는 정수값 이여서 .0 필요)

floor(9.15)= 9.15 를 내림 :9.1 $9.15 \times 10 = 91.5 / 10 = 9.15$

문자열에 관련된 대표적 클래스 : String

String은 객체 자료형이다.

String은 대문자로 시작하므로, 기초데이터가 아닌 객체데이터이다

하지만 뒤에 생성자 (new)를 사용하지 않고 기초데이터를 만들 때처럼 초기화 한다

String의 주요 기능들(메소드)

String concat(String str) : 저장된 문자열과 str문자열을 결합

String substring(int begin) : 시작위치부터 마지막까지의 문자열을 반환

int length() : 문자열 길이를 반환

String toUpperCase() : 대문자로 반환

String toLowerCase() : 소문자로 반환

char charAt(int index) : index 위치의 문자를 반환

int indexOf(char ch) : 첫번째 ch문자의 위치를 반환

int lastIndexOf(char ch) : 마지막 ch문자의 위치를 반환

boolean equals(String str) : 지정된문자열과 str문자열이 같은지 비교

boolean equalsIgnoreCase(String str) : 대소문자구분없이 지정된문자열과 str문자열이 같은지 비교

String trim() : 문자열 앞뒤 공백제거

String replace(char old, char new) : 문자열 내의 old문자를 new문자로 반환

String replaceAll(String old, String new) : 문자열 내의 정규표현식 문자열을 new로 반환

※ **clone 메소드**(객체를 복제하는 메소드) –

객체 자신을 복제해서 새로운 객체를 생성하는 메소드

- Cloneable 인터페이스를 구현한 클래스의 인스턴스만 복제할 수 있 다.
- Object 클래스에 정의된 clone()은 인스턴스 변수의 값만 복제한다.
- 인스턴스 변수가 참조형일 때 참조하는 객체도 복제되게 오버라이딩해야 함

※ **getClass메소드** : 객체가 속하는 클래스의 정보(이름) 리턴

※ **getSuperclass메소드** : 슈퍼클래스의 정보를 리턴

Scanner, Wrapper, Timer

1. Scanner 및 Sysout

※ **Scanner** : 키보드에서 타이핑하는 문자열 또는 입출력 프로그래밍에서 값을 읽어올 때 무엇인가를 얻어 올 때 사용

2. Wrapper 클래스의 이해

※기초데이터를 객체데이터로 변환 합니다.

Wrapper클래스는 기초데이터를 객체데이터로 변화시키는 클래스

기초 데이터	객체 데이터
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
boolean	Boolean
char	Character

Timer 클래스 자바에서의 타이머는 2 가지 있다.

일정한 시간마다 작동되는 타이머 프래그램과 일정한 시간이 되면
한번 실행되는 타이머 프로그램.

※ Timer, TimerTask 클래스 Timer 객체는 일정한 시간이되면,

TimerTask 객체가 작동되도록 하거나 일정시간마다

TimerTask 객체가 작동되도록 한다.

TimerTask 클래스는 추상클래스이므로 ,

TimerTask 클래스를 상속받는 클래스를 만들어서 사용해야 한다