**객체지향 프로그래밍** | 최윤정 교수님 | 과제#8 Summary 이서현 2276219

**<Object-String>**

주요 내용: Object 클래스, 기본 메소드, 객체의 비교연산자 : == , equals(), Wrapper 클래스, 박싱과 언박싱, String 클래스, 기본 메소드, StringBuffer, StringTokenizer

**Object 클래스**

java.lang 패키지에 포함.

모든 클래스의 수퍼 클래스, 모든 클래스에 강제 상속.

모든 객체가 공통으로 가지는 객체의 속성을 나타내는 메소드 보유

\*주요 메소드: boolean equals(Object obj), Class getClass(), int hashCode(), String toString(), void notify(), void notifyAll(), void wait() 등

**toString() : 객체정보를 문자열로 변환**

String toString(): 객체정보를 문자열로 변환한다.

개발자는 자신만의 toString() 작성 필요. 필요시 Object의 toString()메소드를 오버라이딩한다.

**객체 비교와 equals()**

== 연산자 : 두 개의 레퍼런스 비교

boolean equals(Object obj) : 객체 내용이 같은지 비교

**Wrapper 클래스**

* 자바의 기본 타입을 클래스화한 8개 클래스

(기본 타입) int, char …

(Wrapper 클래스) Integer, Character …(이것들 외에는 첫글자 대문자)

* Wrapper 객체 생성 방법: (더 이상 생성자를 이용하지 않는다)

1. 기본 타입의 값으로 생성.
2. 문자열로 생성.

+ float 객체는 double 타입의 값으로 생성할 수 있다.

* Wrapper 객체들은 거의 유사하며 많은 메소드가 static 타입.

주요 메소드: static int parseInt(String s) : 문자열 s를 10진 정수로 변환한 값 리턴

static Integer valueOf(int i) : 정수 i를 담은 Integer 객체 리턴

**Wrapper 활용**

* Wrapper 객체로부터 기본 타입 값 알아내기, 문자열🡪기본 타입 변환, 기본 타입🡪문자열 변환

**박싱Boxing:** 기본 타입을 문자열로 변환

**언박싱Unboxing:** Wrapper 객체에 들어 있는 기본 타입의 값을 빼내는 것

**String의 특징과 객체 생성**

* java.lang.String
* String 클래스는 하나의 문자열로 표현가능
* String 생성자: String(), String(char[] value), String(String original), String(StringBuffer buffer)

**스트링 리터럴과 new String()**

* 리터럴로 생성: String s = "Hello"
* JVM이 리터럴 관리, 응용프로그램 내에서 공유됨
* String 객체로 생성: String t = new String("Hello");
* 힙 메모리에 String 객체 생성
* 스트링 객체의 주요 특징: 스트링 내용 비교 시 반드시 equals를 사용  
  ('=='는 참조대상, 주소를 비교한다)
* 스트링 클래스에서 제공하는 메소드를 사용한다

**StringBuffer 클래스** (Java.lang.StringBuffer)

* 가변 크기의 문자열 저장 클래스. String 클래스와 달리 문자열 변경 가능
* StringBuffer 객체의 크기는 스트링 길이에 따라 가변적 생성

StringBuffer sb = new StringBuffer("java");

**StringTokenizer 클래스** (java.util.StringTokenizer)

* 하나의 문자열을 구분 문자(delimiter)를 이용하여 여러 토큰(구분 문자로 분리된 문자열)로 분리
* String 클래스의 split() 메소드를 이용하여 동일한 구현 가능

**Math 클래스**(java.lang.Math): 산술 연산 메소드 제공

**Calendar 클래스**(java.util.Calendar): 시간과 날짜 정보 저장 관리용 추상클래스