

01.10

String 클래스란?

자바에서 문자열을 다루기 위해 존재하는 클래스입니다. final 클래스로 더 이상 확장할 수 없습니다.

String 클래스가 구현한 3개의 interface

1) Serializable

- 구현한 클래스가 **직렬화** 가능함을 명시하는 인터페이스(메소드 x → **Tag interface pattern**)
- 구현한 클래스의 인스턴스가 **파일로 저장**되거나 **다른 서버에 전송** 가능한 상태가 될 수 있습니다.
- 구현한 클래스는 **SerialVersionUID**를 가지며, 명시하지 않을 경우 컴파일러에 의해 부여됩니다.
- **SerialVersionUID**를 통해 같은 객체인지를 구분하고, 차이 발생 시 `InvalidClassException`가 발생합니다.

2) Comparable

- **객체 간 비교**를 위해 구현해야 하는 인터페이스.
- 구현할 메소드는 **compareTo()** 단 하나입니다. 자신과 파라미터를 비교하여 int를 리턴합니다.
- 비슷한 인터페이스로 `Comparator`가 있습니다. `compare()` 메서드를 구현하여 파라미터로 들어오는 두 변수를 비교합니다.

3) CharSequence

- **문자열을 다루는 클래스임**을 명시적으로 나타냅니다.

직렬화, Serializable

- 자바 시스템 내부 객체나 데이터를 외부 자바 시스템에서도 사용할 수 있도록 변환하는 기술
= JVM 내의 메모리에 상주한 객체, 데이터를 바이트 형태로 변환하는 기술

- **역직렬화 조건**

- 객체의 타입.class가 class path에 있어야 하고, SUID가 있어야 합니다.

- **상속**

- 상속 관계 있을 때, 하위 클래스도 직렬화됩니다.
- 부모 객체에서 Serializable를 구현하지 않았을 경우, 해당 속성이 자식 객체의 직렬화에서 제외됩니다.
- 인스턴스 변수의 타입이 아닌, 실제 인스턴스의 타입 대로 직렬화됩니다.
- 객체의 속성인 타입이 직렬화를 지원하지 않을 경우, NotSerializableException 발생
- 클래스의 멤버에 대한 타입 구분이 엄격합니다 (ex. int → long, String, StringBuilder 안됨)
- 사용되는 곳 - 서블릿 세션(파일 저장, 세션 클러스터링 등), 캐시(redis 등), 자바 RMI
- **transient** 키워드 - 직렬화 전송 시 선언된 변수는 직렬화 대상에서 제외

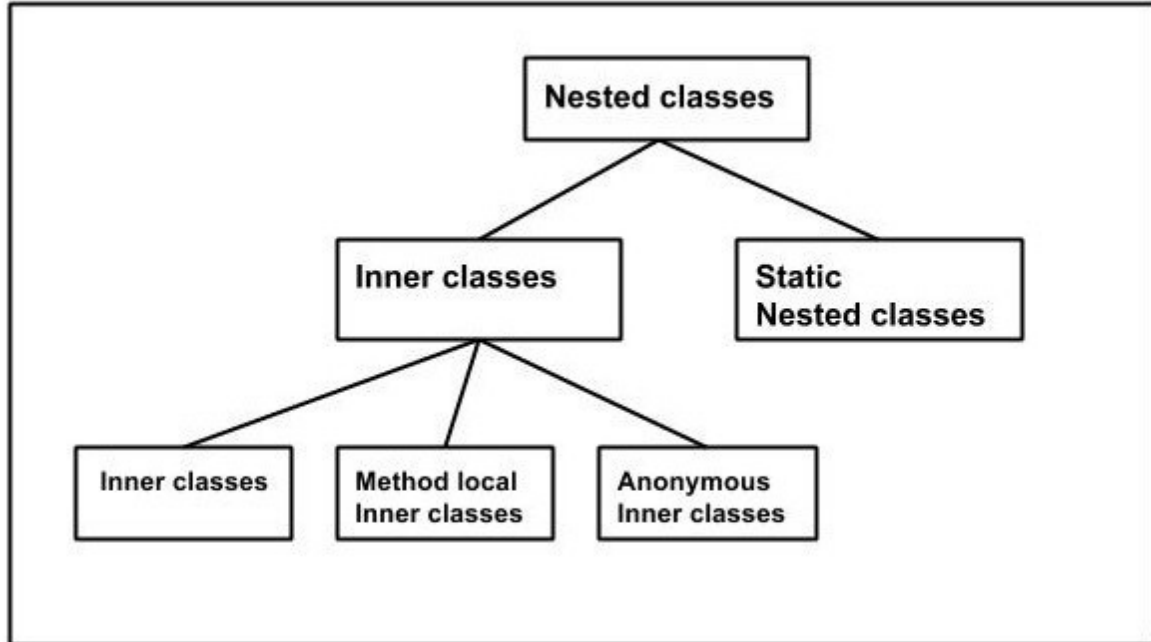
String, StringBuilder, StringBuffer

- 리터럴 → String Constant Pool에 저장
- StringBuilder → 스레드 세이프, StringBuffer → 스레드
 - 연산 시 새로운 객체 생성 x

String.intern()

- 동일한 값의 문자열이 Pool에 있을 경우, 해당 객체를 참조하는 객체를 리턴

📌 중첩 클래스



중첩 클래스의 사용 이유

- 한 곳에서만 사용되는 클래스를 논리적으로 묶어서 처리할 필요가 있을 때(Static Nested)
- 캡슐화가 필요할 때, 즉, 내부 구현을 감추고 싶을 때(Inner)
- 소스의 가독성과 유지 보수성을 높이고 싶을 때

Static Nested Class

- 컴파일 시 외부 클래스와 별개의 클래스 파일을 생성합니다.
- 논리적으로 클래스를 묶기 위해 사용합니다.
ex) Elementary - Student , University - Student
- 외부 클래스의 인스턴스 없이 내부의 인스턴스를 생성할 수 있습니다.
- 외부 클래스의 static 변수에 클래스명 참조 없이 접근이 가능합니다.
- Static Nested Class가 jvm에 초기화되는 과정
 - 1) 외부 클래스 로드~초기화

2) Static Nested 클래스의 클래스 정보 조회 시 → Static Nested 클래스 loading

ex) OuterClass.StaticClass;

3) Static Nested 클래스의 메서드 호출 시 → Static Nested 클래스 initializing

ex) OuterClass.StaticClass.methodA() *methodA는 static 메소드

Inner Class - Local Inner Class

- 하나의 클래스에서 어떤 공통적인 작업을 수행하는 클래스가 필요하고, 다른 클래스에서는 그 클래스가 전혀 필요 없을 때 사용합니다.
- 외부 클래스의 인스턴스에 대한 외부 참조를 가집니다. 따라서 외부 클래스의 모든 필드 혹은 메소드에 대한 접근이 가능합니다.
- 외부 참조를 가지고 있으므로 외부 클래스의 인스턴스가 gc되지 못하게 하는 요인이 될 수 있습니다.

Inner Class - AnonymousInner Class

- 클래스 이름, 객체 이름이 없으며, 다른 클래스나 메소드에서 참조할 수 없습니다.
- 애플리케이션 시작 시 클래스 패스 상에 타입이 존재하지 않으므로 빠른 애플리케이션 시작이 가능합니다.
- GUI에서 이벤트 리스너를 만들 때 사용될 수 있습니다.