

프로그래밍 공부의 기초!



Java

자바(Java)

접근 제한자

자바(Java)

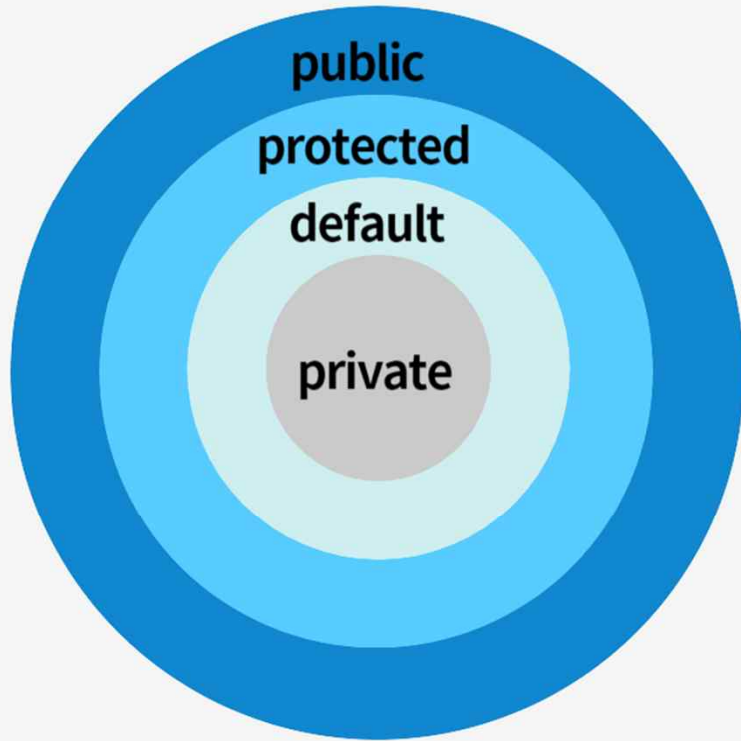
Quiz

- 클래스(AccessModifier1)에 메소드를 생성하여
x, y 인자 값을 전달 받아
- x 가 크면 “x이 y보다 크다.” 라는 문자열을
 - y 가 크면 “y이 x보다 크다.” 라는 문자열을
 - 같은 경우 “x와 y는 같다.” 라는 문자열을 반환 되게 하시오.

자바(Java)

접근 제어자 (access modifier) 정리

클래스, 메서드, 필드 등의 멤버에 대한 접근 권한을 설정하는 데 사용됨



private : 해당 클래스에서만 접근가능

default : 같은 패키지에서만 접근 가능

protected : 같은 패키지, 상속 관계만 접근 가능

public : 어디서든 접근 가능

자바(Java)

파일 관리

자바(Java)

Quiz

하나의 Class에서 모든 기능을 구현하는 것은 좋지 않은 방식으로
특정 기능별 Class 파일을 따로 만들어 관리하는것이 좋은 방식 입니다.
인적사항 출력 예제로 클래스별 기능을 다르게 구현 하시오.
(이름 출력, 나이 출력, 주소 출력, Main Class로 구분(EX04~EX07))

자바(Java)

Instance Member & Static Member

자바(Java)

Instance Member & Static Member

Field와 Method는 Instance Member와 Static Member로 구분됨

- Instance Member

객체가 생성되어야지만, 사용 할 수 있는 Field, Method를 의미
EX) 하나의 클래스로 여러 개의 객체를 생성 시 각 객체마다 서로 다른 값을 설정하는 경우

- Static Member

객체가 생성되지 않아도, 사용 할 수 있는 Field, Method를 의미한다.
EX) 하나의 클래스로 여러 개의 객체를 생성 후 모든 객체에서 공통 된 값을 설정해야하는 경우

자바(Java)

정보 은닉 & 캡슐화

자바(Java)

정보 은닉

클래스의 필드변수의 접근제한을 통해 클래스의 정보를 숨기는 행위를 의미

모든 멤버변수가 외부에 공개되어있을 경우 누구나 클래스의 멤버변수 값을 임의로 조작할 수 있는 상태가 만들어질 수 있음

클래스의 정보은닉을 구현하기 위해서는 멤버변수의 접근제한자 Private을 설정을 통해 외부에서의 접근제한을 명시

접근제한이 설정 된 멤버변수의 값은 객체내부의 검사 및 검증 작업을 진행 한 후 멤버변수의 값을 변경할 수 있도록 구현 위와 같은 형태로 클래스의 내부구조를 구성 하는것을 클래스의 정보은닉이라 표현

자바(Java)

캡슐화

하나의 온전한 기능을 수행 할 수 있는 클래스를 정의한 것을 의미

객체 지향 프로그래밍(OOP)에서 사용되는 중요한 개념 중 하나로, 관련된 데이터와 기능을 하나의 단위로 묶고, 외부에서의 접근을 제한하는 것을 의미 하며 , 메서드를 사용하여 데이터에 접근하도록 제어함으로써, 데이터에 대한 부적절한 직접적인 접근을 막음

자바(Java)

Getter & Setter

자바(Java)

Getter & Setter

접근제한이 설정된 필드변수의 접근하기 위한 메서드

Getter Method : 필드변수의 값을 사용 할 때 사용

Setter Method : 필드변수의 값을 설정 할 때 사용

Getter, Setter는 프로그래머들간 약속된 이름구조 (다른 이름 사용가능)

Getter Method 이름지정 기본형식 : get + Field name (getName)

Setter Method 이름지정 기본형식 : set + Field name (setName)

자바(Java)

생성자

자바(Java)

생성자

객체 생성 시 해당 객체의 초기화 작업을 담당

필드 초기화, 메서드 호출 등의 작업을 수행하며, 모든 클래스에는 반드시 하나 이상의 생성자가 존재해야함.

Constructor 특징

- 생성자의 이름은 반드시 클래스 이름과 동일해야함
- 클래스의 명시된 접근 제한자를 동일하게 물려받아 적용됨
- 생성자를 명시하지 않을 경우 기본 생성자가 자동 추가됨
- 생성자는 Method와 비슷한 형식을 사용하지만, 반환 값이 없음
- 생성자를 명시적으로 선언한 경우 기본 생성자는 자동 추가되지 않음

See you
Again!

→ 다음 이 시간에 만나요!