

Effective Java

아이템 16

'**public 클래스**에서는 public 필드가 아닌 **접근자 메서드**를 사용하라'

'public 클래스에서는 public 필드가 아닌 접근자 메서드를 사용하라'

참고 자료

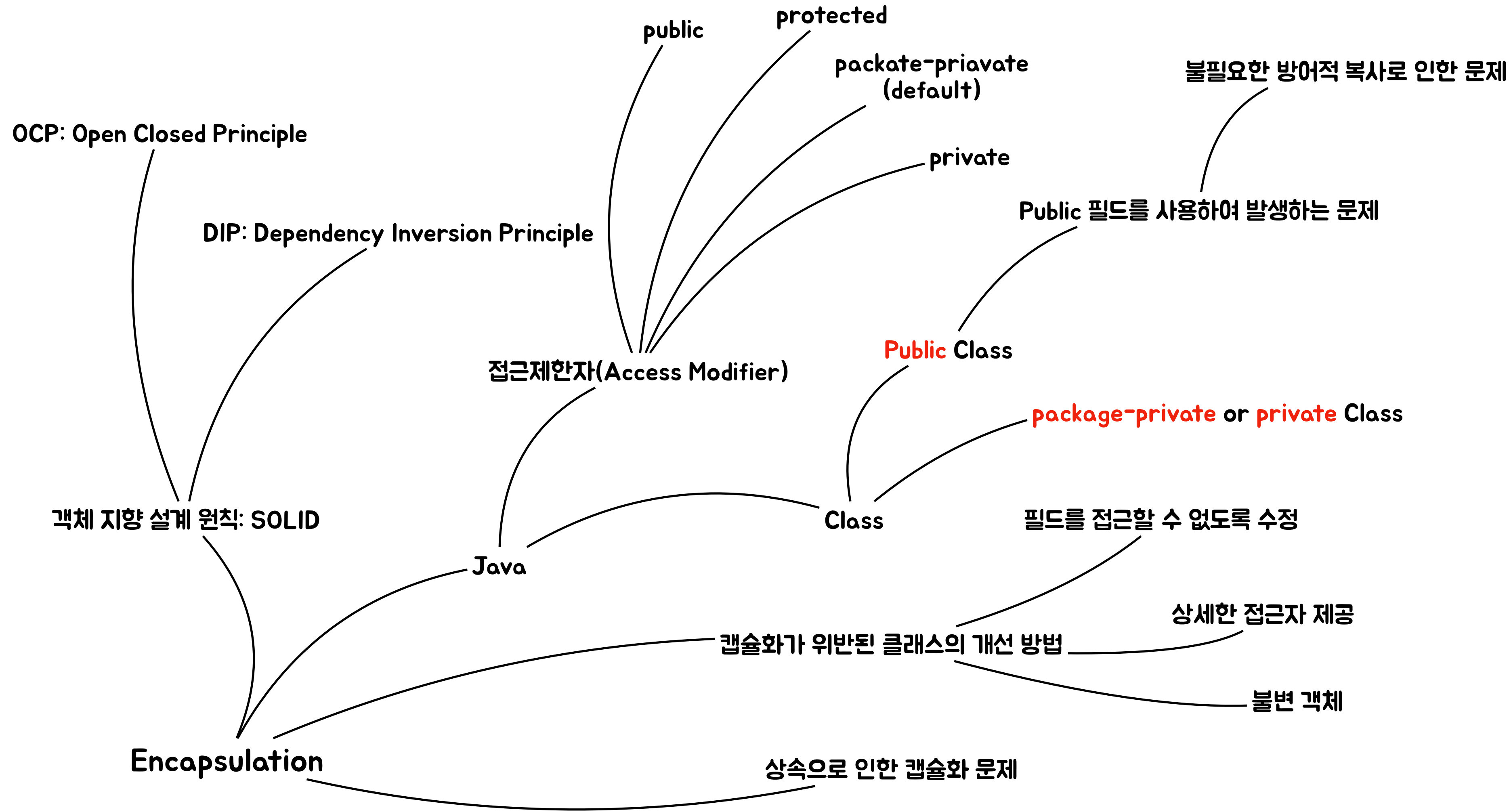
- The Java Tutorials (Controlling Access to Members of a Class)
- 오브젝트 - 조영호
- Access Modifiers in Java

- * Encapsulation
1. 객체 지향 설계 5가지 원칙 (SOLID)
 - DIP: 의존 역전 원칙
 - OCP: 개방 폐쇄 원칙
 2. Java 기본
 - Class
 - Access Modifier
 3. 클래스 용도에 따른 필드 설정
 - public 클래스(Dimension, Point)
 - 개선을 위한 노력
 - 필드의 접근 제한자를 수정
 - 상세한 접근자 제공
 - package-private & private 클래스

'public 클래스에서는 public 필드가 아닌 접근자 메서드를 사용하라'

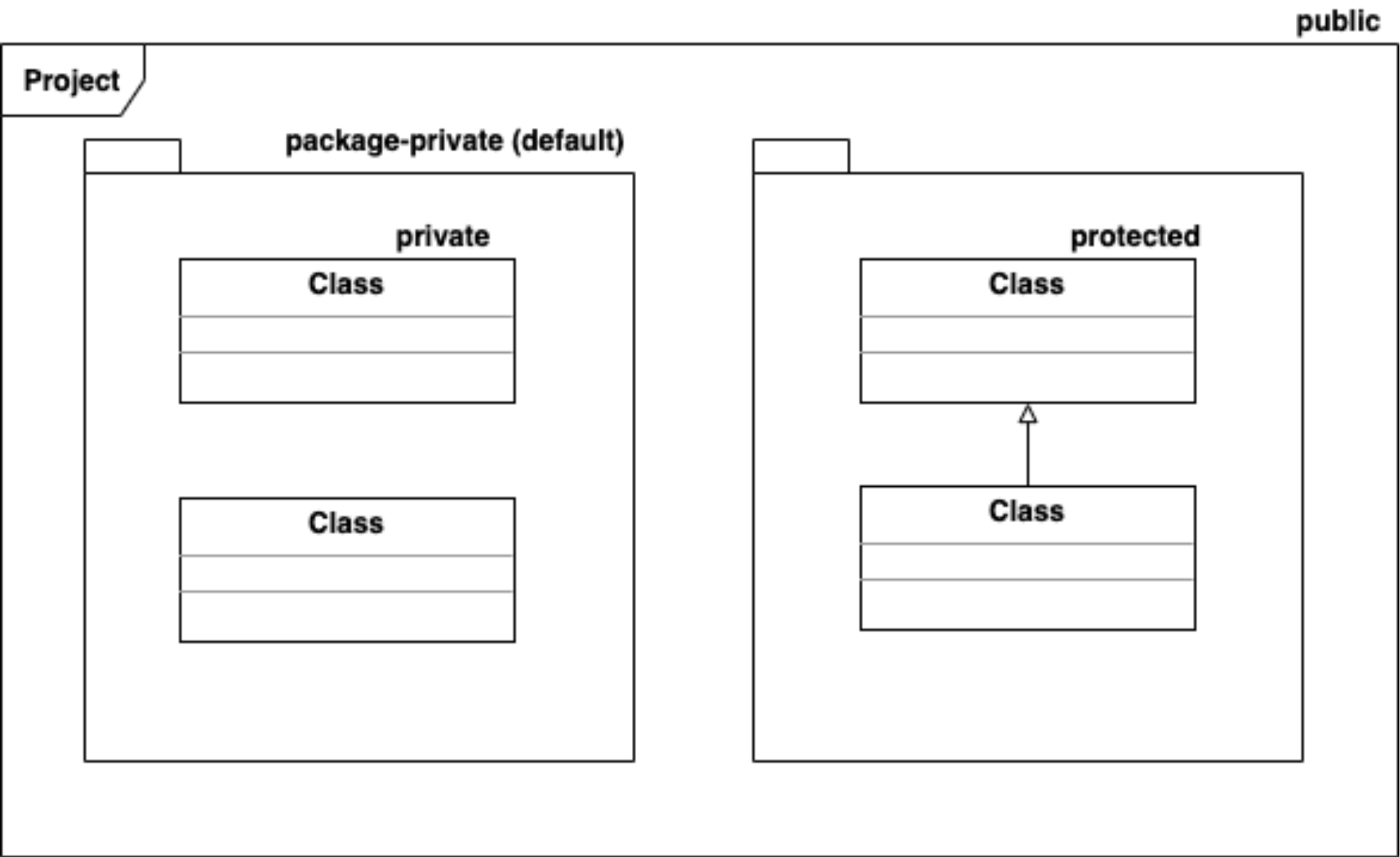
* Encapsulation

1. 객체 지향 설계 5가지 원칙 (SOLID)
 - DIP: 의존 역전 원칙
 - OCP: 개방 폐쇄 원칙
2. Java 기본
 - Class
 - Access Modifier
3. 클래스 용도에 따른 필드 설정
 - public 클래스(Dimension, Point)
 - 개선을 위한 노력
 - 필드의 접근 제한자를 수정
 - 상세한 접근자 제공
 - package-private & private 클래스



객체 지향 설계 5가지 원칙	* Encapsulation
	1. 객체 지향 설계 5가지 원칙 (SOLID) <ul style="list-style-type: none">- DIP: 의존 역전 원칙- OCP: 개방 폐쇄 원칙
	2. Java 기본 <ul style="list-style-type: none">- Class- Access Modifier
	3. 클래스 용도에 따른 필드 설정 <ul style="list-style-type: none">- public 클래스(Dimension, Point)- 개선을 위한 노력<ul style="list-style-type: none">- 필드의 접근 제한자를 수정- 상세한 접근자 제공- package-private & private 클래스
OCP 원칙 위반 <ul style="list-style-type: none">- '확장에는 열려 있고, 수정에는 닫혀 있어야 한다.'	
단일 책임 원칙 (Single Responsibility Principle)	
개방-폐쇄 원칙 (Open Cloned Prinsiple)	
리스코프 치환 (Liskov Substitution Principle)	
인터페이스 분할 원칙 (Interface Segregation Principle)	
의존 역전 원칙 (Dependency Inversion Principle)	
DIP 원칙 위반 <ul style="list-style-type: none">- '고수준 모듈은 저수준 모듈의 구현에 의존해서는 안된다.'	

- 1. 객체 지향 설계 5가지 원칙 (SOLID)
 - DIP: 의존 역전 원칙
 - OCP: 개방 폐쇄 원칙
- 2. Java 기본
 - Class
 - Access Modifier
- 3. 클래스 용도에 따른 필드 설정
 - public 클래스(Dimension, Point)
 - 개선을 위한 노력
 - 필드의 접근 제한자를 수정
 - 상세한 접근자 제공
 - package-private & private 클래스



	private	package-private (default)	protected	public
Same Class	○	○	○	○
Same package subclass	○	✗	○	○
Same package Non-subclass	✗	✗	○	○
Different package Subclass	✗	✗	○	○
Different package Non-subclass	✗	✗	✗	○

클래스 용도에 따른 필드 설정	* Encapsulation
<div>public 클래스</div> <div><ul style="list-style-type: none">- public 클래스임에도 캡슐화를 지키지 않은 예시<ul style="list-style-type: none">- <u>Dimension</u>- <u>Point</u>- public 클래스 코드 개선<ul style="list-style-type: none">- <u>AdvancedDimension</u>- <u>AdvancedPoint</u></div>	<div>1. 객체 지향 설계 5가지 원칙 (SOLID)</div> <div><ul style="list-style-type: none">- DIP: 의존 역전 원칙- OCP: 개방 폐쇄 원칙</div> <div>2. Java 기본</div> <div><ul style="list-style-type: none">- Class- Access Modifier</div> <div>3. 클래스 용도에 따른 필드 설정</div> <div><ul style="list-style-type: none">- public 클래스(Dimension, Point)- 개선을 위한 노력<ul style="list-style-type: none">- 필드의 접근 제한자를 수정- 상세한 접근자 제공- package-private & private 클래스</div>

클래스 용도에 따른 필드 설정	* Encapsulation
<div>package-private or private 중첩 클래스</div> <div><ul style="list-style-type: none">- package-private subclass를 포함하는 클래스 사례<ul style="list-style-type: none">- <u>Outer Class</u>- <u>TestCase</u></div>	<div>1. 객체 지향 설계 5가지 원칙 (SOLID)<ul style="list-style-type: none">- DIP: 의존 역전 원칙- OCP: 개방 폐쇄 원칙</div> <div>2. Java 기본<ul style="list-style-type: none">- Class- Access Modifier</div> <div>3. 클래스 용도에 따른 필드 설정<ul style="list-style-type: none">- public 클래스(Dimension, Point)- 개선을 위한 노력<ul style="list-style-type: none">- 필드의 접근 제한자를 수정- 상세한 접근자 제공- package-private & private 클래스</div>

<div data-bbox="1246 45 1336 101">정리</div>	<div data-bbox="2592 45 3325 774"><div data-bbox="2592 45 2888 101">* Encapsulation</div><div data-bbox="2628 138 3325 774"><div data-bbox="2628 138 3278 270">1. 객체 지향 설계 5가지 원칙 (SOLID)<ul style="list-style-type: none">- DIP: 의존 역전 원칙- OCP: 개방 폐쇄 원칙</div><div data-bbox="2628 318 3002 450">2. Java 기본<ul style="list-style-type: none">- Class- Access Modifier</div><div data-bbox="2628 499 3325 774">3. 클래스 용도에 따른 필드 설정<ul style="list-style-type: none">- public 클래스(Dimension, Point)- 개선을 위한 노력<ul style="list-style-type: none">- 필드의 접근 제한자를 수정- 상세한 접근자 제공- package-private & private 클래스</div></div></div>
<div data-bbox="383 832 2192 1046"><div data-bbox="1246 832 1336 889">정리</div><div data-bbox="383 889 2192 1046"><div data-bbox="383 889 1839 945">1. Public 클래스는 필드의 접근 제한자를 private로 설정하여 캡슐화 한다.</div><div data-bbox="383 945 2192 1001">2. package-private or private 클래스는 종종 public으로 설정하는 것이 유용할 수도 있다.</div><div data-bbox="383 1001 1462 1046">3. 객체 지향적인 설계를 위한 최소 조건: 데이터 캡슐화</div></div></div>	