Java Modifiers Matrix

The most complete Java Modifiers Matrix on the web

brought to you by:

http://www.JavaChamp.com

Authors: N. , Y. Ibrahim

Copyright 2009 JavaChamp.com

JavaChamp.com is an Open Certification Platform.

What does this mean?

JavaChamp is the best place to learn, share, and certify your professional skills.
We help you develop yourself in the field of computer science and programming

Here are the most significant features offered by JavaChamp:

Online Exams

Top quality mock exams for SCJP, SCEA, EJB, JMS, JPA.
Start Express or topic-wise customized exam.

- We offer you unlimited free mock exams
- Exams cover subjects like SCJP, SCEA, EJB, JMS, JPA,..
- You can take as many exams as you want and at any time and for no charges
- Each exam contains 20 multiple choice questions
- You can save the exams taken in your exams history
- Your exams history saves the exams you took, the scores you got, time took you to finish the exam, date of examination and also saves your answers to the questions for later revision
- You can re-take the same exam to monitor your progress
- Your exams history helps the system to offer you variant new questions every time you take a new exam, therefore we encourage you to register and maintain an exams history

Network

Find guidance through the maze, meet Study-Mates, Coaches or Trainees...
Studying together is fun, productive and helps you in building your professional network and collecting leads

Bookshelf

JavaChamp Bookshelf full of PDF eBooks...
Download PDF books with a selected sample of the JavaChamp question bank in SCJP, SCEA, EJB, JMS and more or read it online

JavaChamp Profile

You may publish your profile and connect to your colleagues and friends.

Content Channel

Be an Author and get recognition, leads, and more...
Contributing to the JavaChamp question bank will earn your recognition of your professional skills, expands your network, introduce you to potential leads

Join Us

Join the fast growing JavaChamp Community now.
JavaChamp Community is young and very dynamic, we would be thrilled to welcome you on board :o)

Java Modifiers Matrix:

The matrix is split into 3 categories:

- Access Modifiers
  - public
  - *implicit package*
  - protected
  - private

- Structural Modifiers
  - final
  - abstract
  - static
  - native

- Behavioral Modifiers
  - strictfp
  - transient / volatile
  - synchronized

Matrix symbols legend:

✓ Possible modifier choice
✓ Default modifier (java choice if not explicitly defined)
✓ Single choice (other modifiers are not allowed)

		Access Modifiers				Structural Modifiers				Behavioral Modifiers		
		public	*implicit package*	protected	private	final	abstract	static	native	strictfp	transient / volatile	synchronized
Class	Top level Class	✓	✓			✓	✓			✓		
	Class inside class	✓	✓	✓	✓	✓	✓	✓		✓		
	Class inside Interface	✓				✓	✓	✓		✓		
	Class inside method / code block					✓	✓			✓		
Interface	Top level Interface	✓	✓				✓			✓		
	Interface inside Class	✓	✓	✓	✓		✓	✓		✓		
	Interface inside method / code block	✓						✓				
Code Block	Inside Class							✓				
	Inside method											✓
enum	Top level Enumeration	✓	✓									
	Enum inside Class	✓	✓	✓	✓			✓				
	Enum inside Interface	✓						✓				
Method	Class	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓
	Interface	✓					✓					✓
Constructor	Class	✓	✓	✓	✓							
Data field (variable / constant)	Class	✓	✓	✓	✓	✓		✓			✓	
	Interface	✓				✓		✓				
	Method argument / local or code block local					✓						

Java modifiers and access rules:

- If class A can't access class B, then class A can't access any member (method or variable) in class B.
- For a subclass outside the package, the protected member can be accessed only through inheritance.
- It is illegal to have even a single abstract method in a class that is not explicitly declared abstract!
- You can have an abstract class with no abstract methods.
- The first concrete subclass of an abstract class must implement all abstract methods of the super class.
- Initialization code block are not allowed in interfaces
- Inner classes in interfaces must be **public static**
- A member/(inner) interface can only be defined inside a top-level class or interface not inside methods
- Static data fields can only be declared in static or top level types

Method illegal modifiers combinations:

- abstract and final — @Compiler Error: The abstract method can only set a visibility modifier, one of public or protected
- abstract and private — @Compiler Error: The abstract method can only set a visibility modifier, one of public or protected
- abstract and static — @Compiler Error: illegal combination of modifiers: abstract and static
- transient and final / static — @Compiler Error: A transient variable may not be declared as final or static

Java member visibility rules across packages:

	Member Visibility Rules						
	Access from			Access to			
	Package	Class/ Interface?		public	*package*	protected	private
1	same	same		✓	✓	✓	✓
2		inner		✓	✓	✓	✓
3		subclass		✓	✓	✓	
4		other		✓	✓	✓	
5	other	subclass		✓		inheritance	
6		other		✓			