

# Java

Αντικειμενοστρεφής Προγραμματισμός

# Βασικά Χαρακτηριστικά OOP

- Object-oriented programming (OOP) is a method of programming based on a hierarchy of classes, and well-defined and cooperating objects
- Ο αντικειμενοστρεφής προγραμματισμός είναι μια μέθοδος προγραμματισμού βασισμένη σε μια ιεραρχία τάξεων και καλά ορισμένα αντικείμενα, τα οποία αλληλεπιδρούν μεταξύ τους

# Class (Τάξη)

- A class is a structure that defines the data and the methods to work on that data
- When you write programs in the Java language, all program data is wrapped in a class, whether it is a class you write or a class you use from the Java platform API libraries
- Τάξη είναι μια δομή που ορίζει δεδομένα και τις μεθόδους που επιδρούν πάνω στα δεδομένα

# Objects (Αντικείμενα)

- An instance is an executable copy of a class
- Another name for instance is object
- There can be any number of objects of a given class in memory at any one time
- Στιγμιότυπο είναι ένα εκτελέσιμο αντίγραφο μιας κλάσης
- Τα στιγμιότυπα τα λέμε και αντικείμενα
- Μπορεί να υπάρξει ένας οποιοσδήποτε αριθμός από αντικείμενα μιας τάξης στη μνήμη, ανά πάσα στιγμή

# Interface

- In the Java programming language, an *interface* is a reference type, similar to a class, that can contain *only* constants, method signatures, default methods, static methods, and nested types
- Method bodies exist only for default methods and static methods. Interfaces cannot be instantiated—they can only be implemented by classes or extended by other interfaces
- Στην Java, το interface είναι reference type, με ομοιότητες με την τάξη, το οποίο μπορεί να περιέχει μόνο σταθερές, υπογραφές μεθόδων, default μεθόδους, στατικές μεθόδους και εμφωλευμένους τύπους
- Σώμα μεθόδων υπάρχει μόνο σε αυτές που είναι δηλωμένες ως static ή default. Τα interfaces δεν παράγουν από μόνα τους αντικείμενα. Τα interfaces υλοποιούνται από τάξεις, ή επεκτείνονται από άλλα interfaces

# Προγραμματιστικά Συμβόλαια

- Implementing an interface allows a class to become more formal about the behavior it promises to provide
- Interfaces form a contract between the class and the outside world, and this contract is enforced at build time by the compiler
- If your class claims to implement an interface, all methods defined by that interface must appear in its source code before the class will successfully compile
- Το interface παρέχει ένα είδος «συμβολαίου» το οποίο περιμένουμε να «τηρήσει» μια τάξη και είναι κάτι που ελέγχεται κατά τη μεταγλώττιση



# Data Types

# Java Data Types

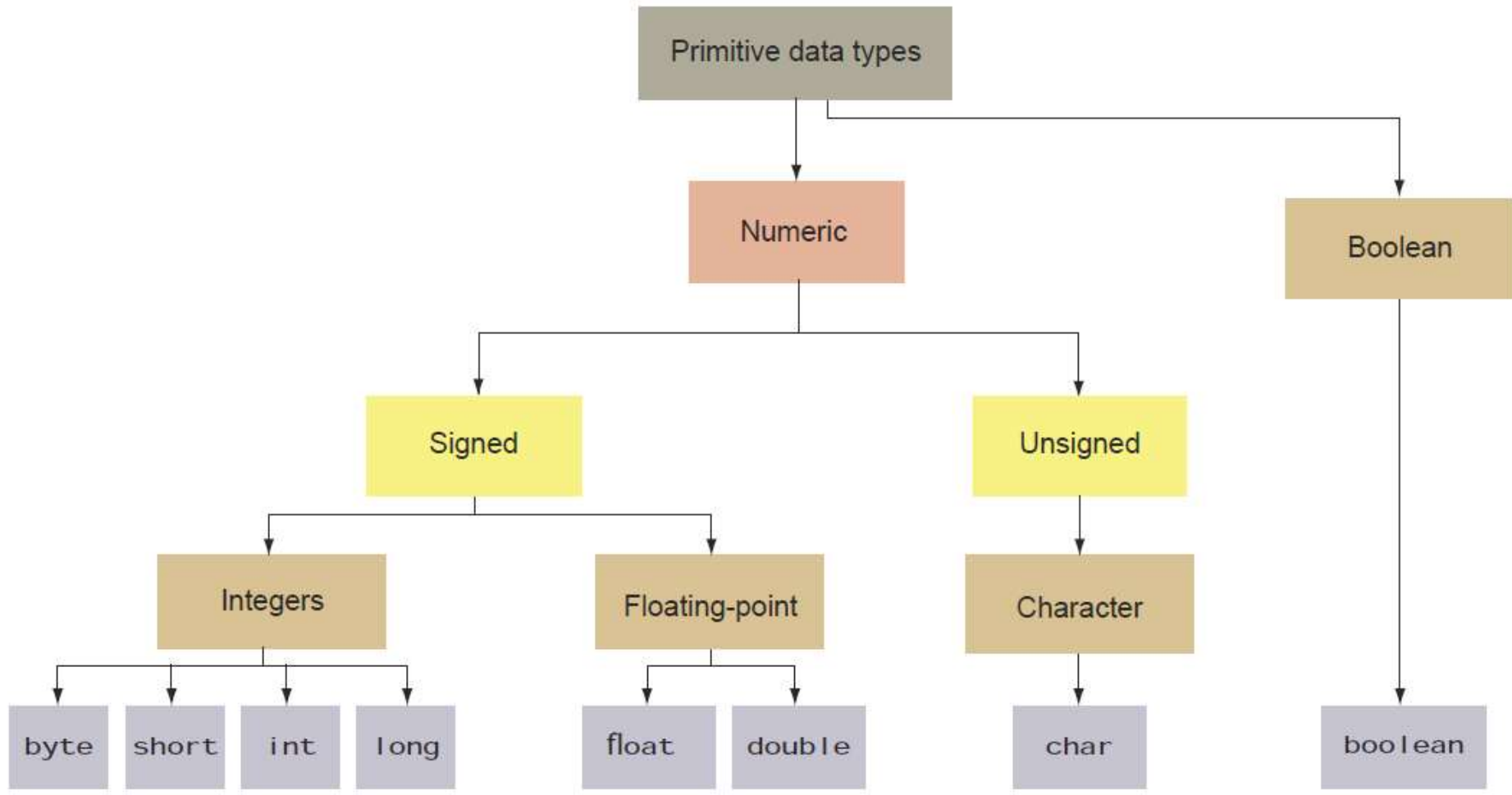
- Οι 2 μεγάλες κατηγορίες μεταβλητών στη Java είναι οι:
  - Primitive variables
  - Reference variables



# Primitive Data Types

- Στη Java υπάρχουν 8:

- char
- byte
- short
- int
- long
- float
- double
- boolean



# Ενδεικτικός πίνακας εύρους τιμών

Data type	Size	Range of values
byte	8 bits	-128 to 127, inclusive
short	16 bits	-32,768 to 32,767, inclusive
int	32 bits	-2,147,483,648 to 2,147,483,647, inclusive
long	64 bits	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807, inclusive

# Προσοχή στους char!



# Ονοματοδοσία για identifiers

Properties of valid identifiers	Properties of invalid identifiers
Unlimited length	Same spelling as a Java reserved word or keyword
Starts with a letter ( a–z, upper- or lowercase), a currency sign, or an underscore	Uses special characters: !, @, #, %, ^, &, *, (, ), ', :, ;, [, /, \, }
Can use a digit (not at the starting position)	Starts with a Java digit (0–9)
Can use an underscore (at any position)	
Can use a currency sign (at any position): ¢, \$, £, ¢, ¥, and others	

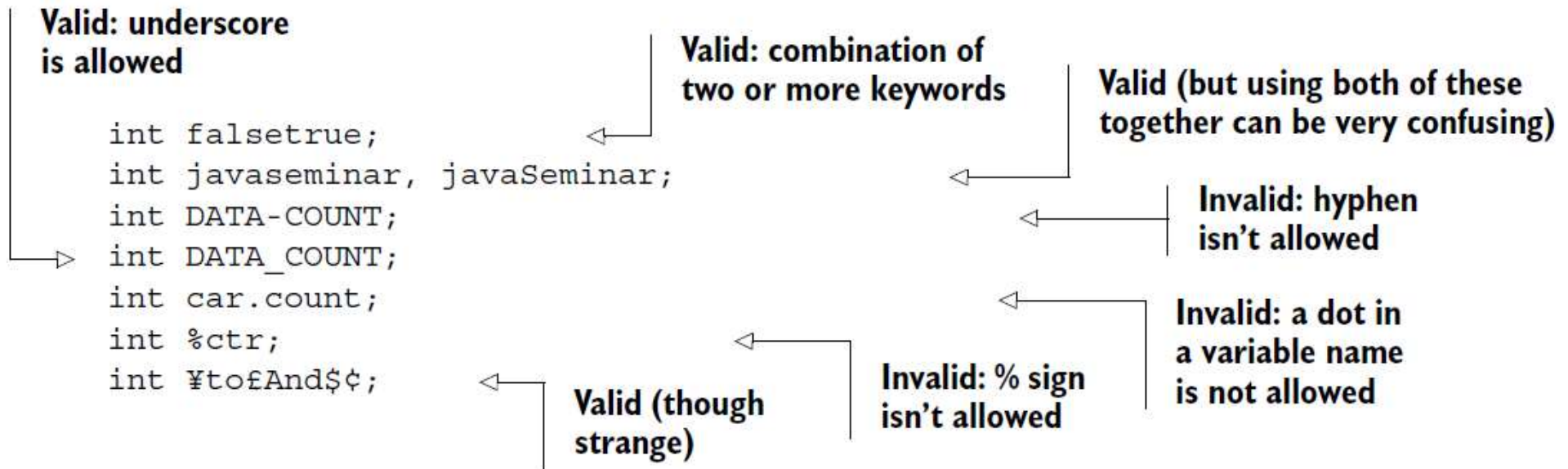
# Παραδείγματα

Examples of valid identifiers	Examples of invalid identifiers
<code>customerValueObject</code>	<code>7world</code> (identifier can't start with a digit)
<code>\$rate, £value, _sine</code>	<code>%value</code> (identifier can't use special char %)
<code>happy2Help, nullValue</code>	<code>Digital!, books@manning</code> (identifier can't use special char ! or @)
<code>Constant</code>	<code>null, true, false, goto</code> (identifier can't have the same name as a Java keyword or reserved word)

# Δεσμευμένες λέξεις Java

<code>abstract</code>	<code>default</code>	<code>goto</code>	<code>package</code>	<code>this</code>
<code>assert</code>	<code>do</code>	<code>if</code>	<code>private</code>	<code>throw</code>
<code>boolean</code>	<code>double</code>	<code>implements</code>	<code>protected</code>	<code>throws</code>
<code>break</code>	<code>else</code>	<code>import</code>	<code>public</code>	<code>transient</code>
<code>byte</code>	<code>enum</code>	<code>instanceof</code>	<code>return</code>	<code>true</code>
<code>case</code>	<code>extends</code>	<code>int</code>	<code>short</code>	<code>try</code>
<code>catch</code>	<code>false</code>	<code>interface</code>	<code>static</code>	<code>void</code>
<code>char</code>	<code>final</code>	<code>long</code>	<code>strictfp</code>	<code>volatile</code>
<code>class</code>	<code>finally</code>	<code>native</code>	<code>super</code>	<code>while</code>
<code>const</code>	<code>float</code>	<code>new</code>	<code>switch</code>	
<code>continue</code>	<code>for</code>	<code>null</code>	<code>synchronized</code>	

# Παραδείγματα



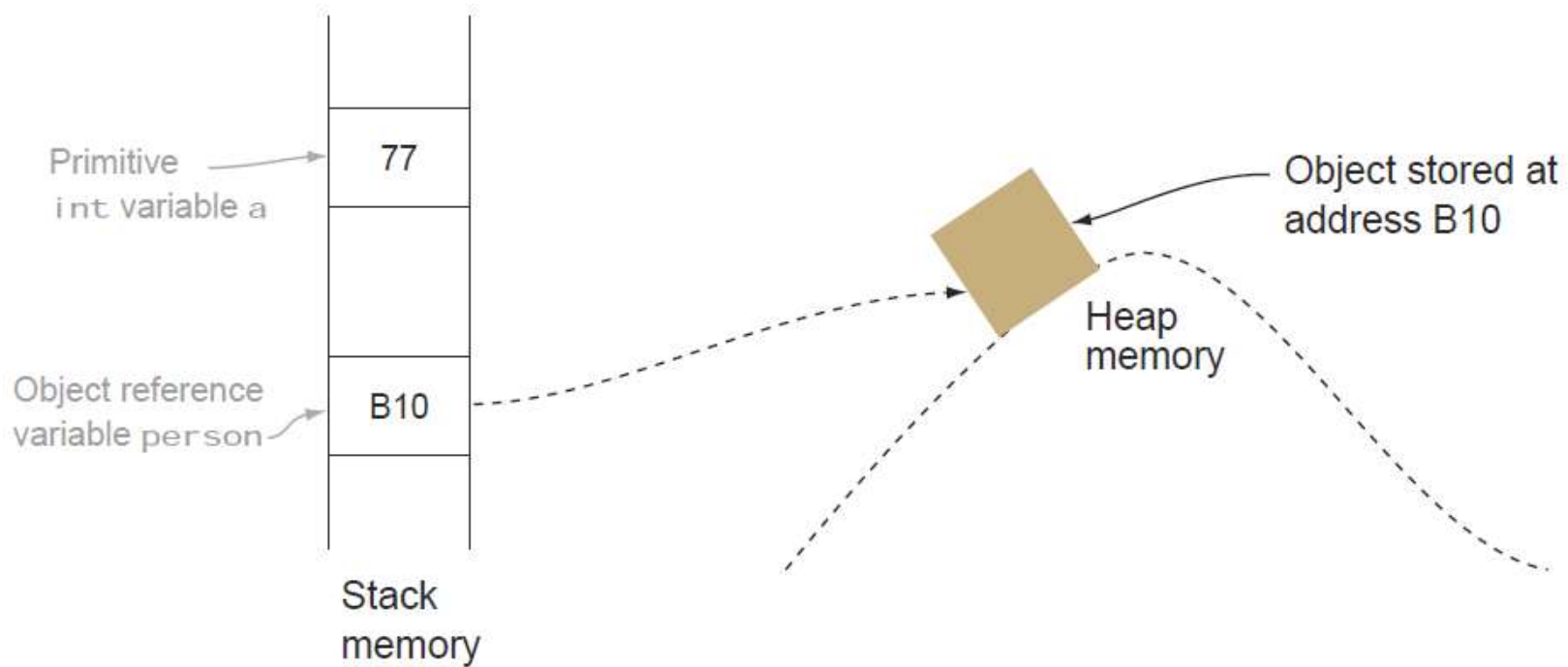


# Reference Variables

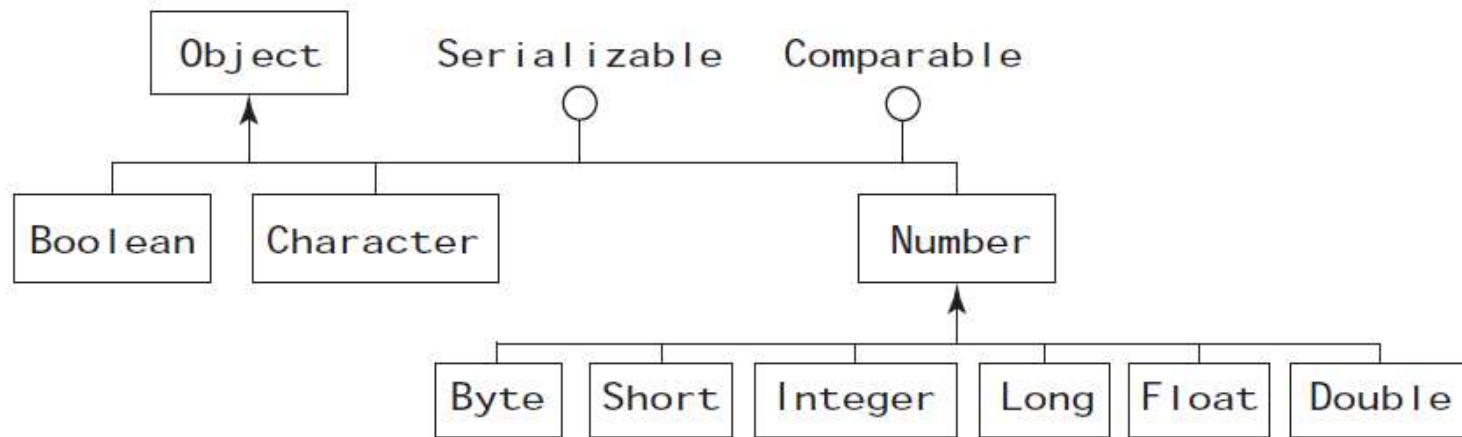
- Αλλιώς Object Variables
- Default τιμή -> null

# References Vs Values

```
int a = 77;  
Person person = new Person();
```



# Wrapper Classes



# Autoboxing - Unboxing

