

# BUSINESS INTELLIGENCE LABORATORY

## Java and Eclipse - recap

# Java

2

Java is a

- High-level
- Object oriented
- Functional (since Java 8)
- Multithreaded
- Architecture neutral
- Portable
- Robust
- Secure

computer programming language

[www.oracle.com/technetwork/java/index.html](http://www.oracle.com/technetwork/java/index.html)

## Java version history

From Wikipedia, the free encyclopedia

The Java language has undergone several changes. It uses *Java Specification Requests* (JSRs) to propose changes.

In addition to the language changes, much more has been introduced, and many of the original tools have been backported.

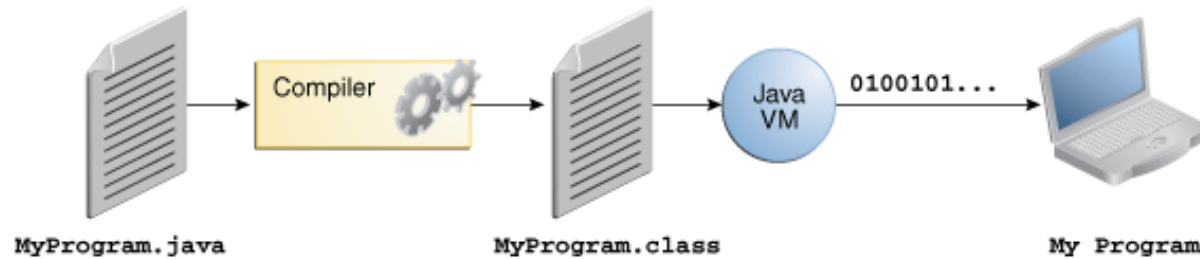
After the Java 7 release, Oracle promised to continue to support Java 6.

### Contents [hide]

- JDK Alpha and Beta (1995)
- JDK 1.0 (January 23, 1996)
- JDK 1.1 (February 19, 1997)
- J2SE 1.2 (December 8, 1998)
- J2SE 1.3 (May 8, 2000)
- J2SE 1.4 (February 6, 2002)
- J2SE 5.0 (September 30, 2004)
- Java SE 6 (December 11, 2006)
  - Java 6 updates
- Java SE 7 (July 28, 2011)
  - Java 7 updates
- Java SE 8 (March 18, 2014)
  - Java 8 updates
- Java SE 9
- Java SE 10
- Implementations
- References
- External links

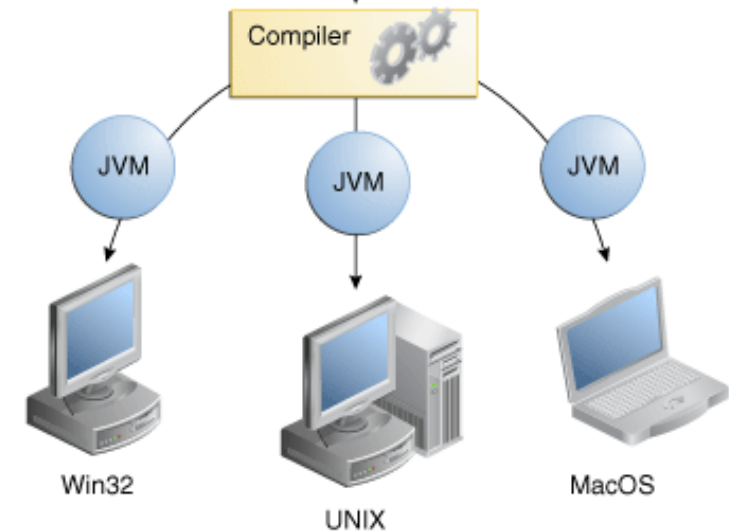
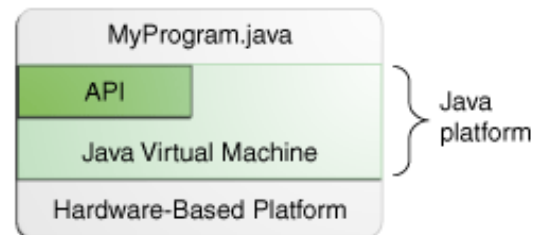
# Java platform

3



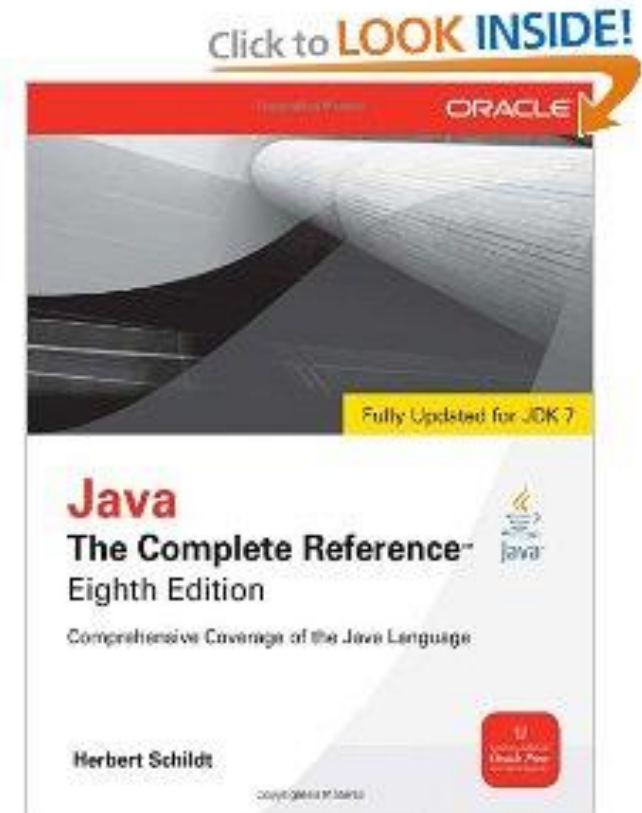
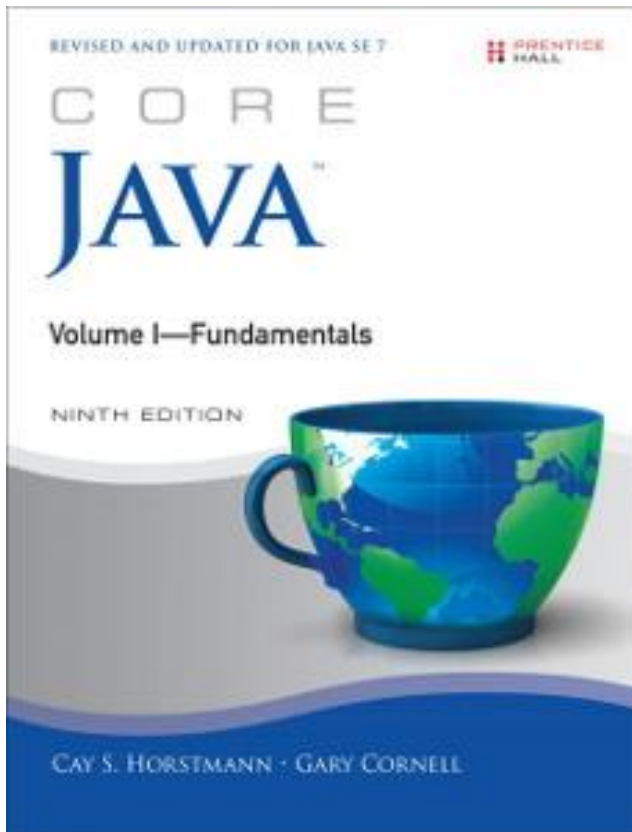
```
Java Program
class HelloWorldApp {
    public static void main(string [] args) {
        System.out.println("Hello World!");
    }
}
HelloWorldApp.java
```

[docs.oracle.com/javase/8/docs/api](https://docs.oracle.com/javase/8/docs/api)



# Java language: books and tutorials

4



[docs.oracle.com/javase/tutorial](http://docs.oracle.com/javase/tutorial)

# Eclipse

5

Eclipse is a multi-language software development environment

| Version Name ↕ | Date ↕              | Platform Version ↕                       |
|----------------|---------------------|--|
| N/A            | 21 June 2004        | 3.0 <sup>[18]</sup>                      |
| N/A            | 28 June 2005        | 3.1                                      |
| Callisto       | 30 June 2006        | 3.2                                      |
| Europa         | 29 June 2007        | 3.3                                      |
| Ganymede       | 25 June 2008        | 3.4                                      |
| Galileo        | 24 June 2009        | 3.5                                      |
| Helios         | 23 June 2010        | 3.6                                      |
| Indigo         | 22 June 2011        | 3.7                                      |
| Juno           | 27 June 2012        | 3.8 and 4.2 <sup>[25]</sup><br>[Notes 1] |
| Kepler         | 26 June 2013        | 4.3                                      |
| Luna           | 25 June 2014        | 4.4                                      |
| Mars           | 24 June 2015        | 4.5                                      |
| Neon           | 22 June 2016        | <b>4.6</b>                               |
| Oxygen         | June 2017 (planned) | 4.7                                      |

[www.eclipse.org/downloads](http://www.eclipse.org/downloads)



GETTING STARTED

MEMBERS

PROJECTS

MORE ▾

# Exercise: maximal subsequence

6

- Given an array of integers, e.g.
  - ▣ `int []a = { -2, 1, -3, 4, -1, 2, 1, -5, 4 };`
- and called
  - ▣  $S(h, k) = \sum_{i=h}^k a[i]$
- the sum of subsequence from h to k, find the maximal  $S(h, k)$ 
  - ▣  $\max S(h, k)$
- For the array above  $\max S(h, k) = S(3, 6) = 4 - 1 + 2 + 1 = 6$
  
- Variants: array of integers
  - ▣ passed on the command line
  - ▣ read from a text file (one int per line)

# OOP concepts

7

- An object-oriented program is made of objects.
  - ▣ Each object has a specific functionality that is exposed to its users, and a hidden implementation
    - Instance fields and methods
    - Encapsulation
- A *class* is the template or blueprint from which objects are made
  - ▣ Classes are cookie cutters. Objects are cookies

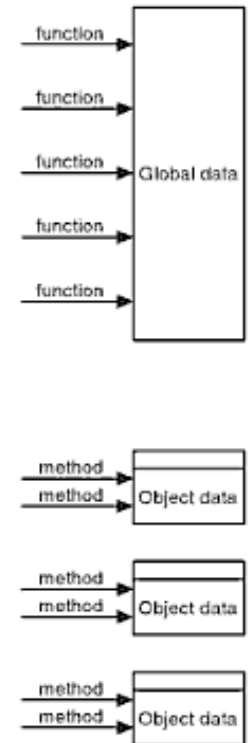


Figure 4-2: Procedural vs. OO programming

# Objects in Java

8

```
Date birthday = new Date();
```

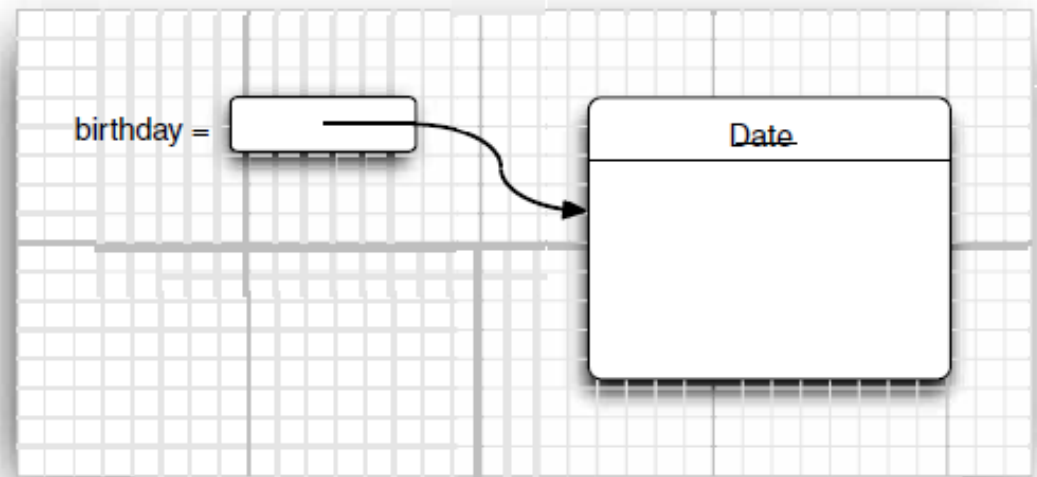


Figure 4-3 Creating a new object

```
Date today = birthday;
```

```
Date deadline; // initialized to null (no referenced object)
```



# OOP concepts

9

- Classes can be built by *extending* other classes
  - ▣ Inheritance and polymorphism
    - The Object class in Java is at the top of the class hierarchy

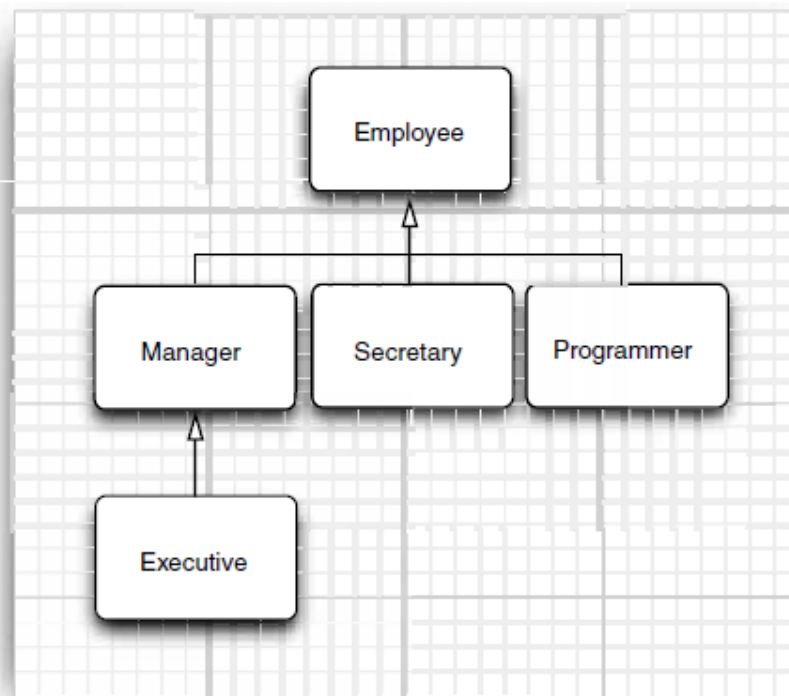


Figure 5-1 Employee inheritance hierarchy

# Collections in Java

10

- <https://docs.oracle.com/javase/tutorial/collections/>

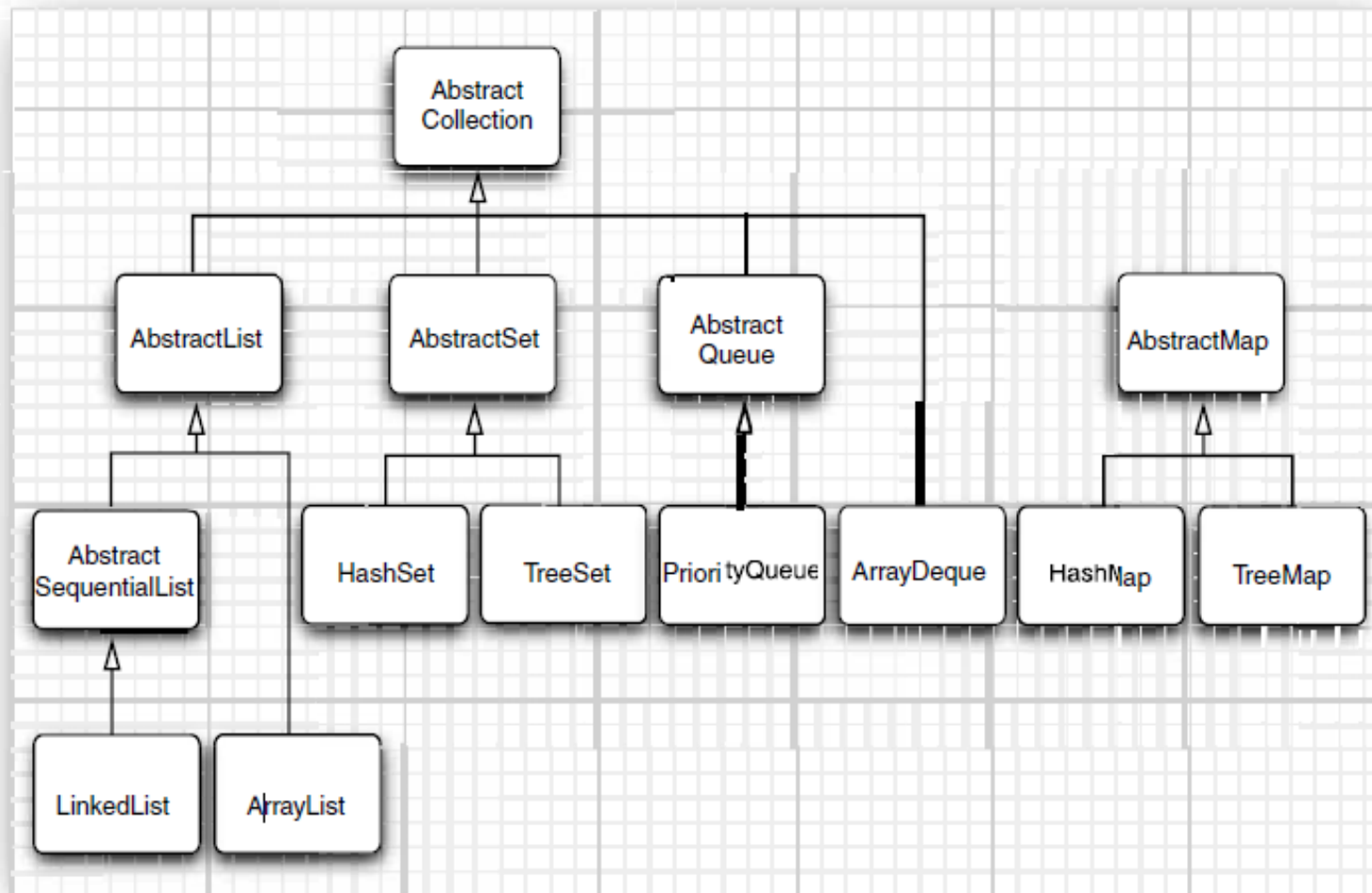


Figure 13-11 Classes in the collections framework

# Exercise: Relational algebra

11

- Code a relational tuple as a Java class
  - Assume only integer and string data types
- Code a relational table as a Collection of tuples
- Code relational operators (as much as possible):
  - Selection:  $\sigma_C(R)$
  - Projection:  $\pi_A(R)$
  - Renaming:  $\rho_{A \leftarrow B}(R)$
  - Distinct:  $\delta(R)$
  - Ordering:  $\tau_A(R)$
  - Grouping:  $A \gamma_{\text{COUNT}(*), \text{AS } B}$
  - Join:  $R \bowtie_{A=B} S$
- Test them on two relations Sales(CustomerName, Product, Amount) and Customer(Name, Country, Age) for the query plans of the following queries:
  - SELECT Name FROM Customer WHERE Age > 20 AND Country='Italy' ORDER BY Age
  - SELECT Country, Count(\*) AS TotSales FROM Sales JOIN Customer on CustomerName=Customer GROUP BY Country