

16. Scheda di lavoro (Java Collections)

23/11/2018

Tipi primitivi: sono predefiniti dal linguaggio, ad essi corrispondono parole chiave ed operatori (ad esempio: tipo int, operatori +, -, ecc.)

Tipi riferimento: sono implementati mediante le **classi**, in particolare le operazioni sono implementate attraverso i **metodi**.

Per utilizzare i tipi di riferimento si consulta la guida ufficiale di Java, in particolare le API di Java SE <https://docs.oracle.com/javase/9/docs/api/overview-summary.html>

API → **Application Programming Interface**, cioè libreria di funzioni (metodi) di supporto alla programmazione.

Java SE → la **Standard Edition** della piattaforma Java (java SE) definisce le **API** che costituiscono il nucleo del linguaggio

Per gestire le **collection** andiamo quindi in Java SE, di qui in Java.base, di qui nel package java.util che contiene tra l'altro il **Collection Framework**.

Iniziamo a leggere l'introduzione della documentazione ufficiale sulle Collections

<https://docs.oracle.com/javase/tutorial/collections/intro/index.html>

A collection— sometimes called a container — is simply an object that groups multiple elements into a single unit. Collections are used to store, retrieve, manipulate, and communicate aggregate data. Typically, they represent data items that form a natural group, such as a poker hand (a collection of cards), a mail folder (a collection of letters), or a telephone directory (a mapping of names to phone numbers). If you have used the Java programming language — or just about any other programming language — you are already familiar with collections.

The Java Collections API's provide Java developers with a set of classes and interfaces that makes it easier to handle collections of objects. In a sense Collection's works a bit like arrays, except their size can change dynamically, and they have more advanced behaviour than arrays.

