Corso di Laurea in Ingegneria Informatica e Automatica
A. A. 2018/2019

Progettazione del Software

Giuseppe De Giacomo, Paolo Liberatore,
Massimo Mecella

Esercitazione 2

API Java

- Con API (Application Programming Interface) si indicano tutta una serie di procedure che sono a disposizione del programmatore.

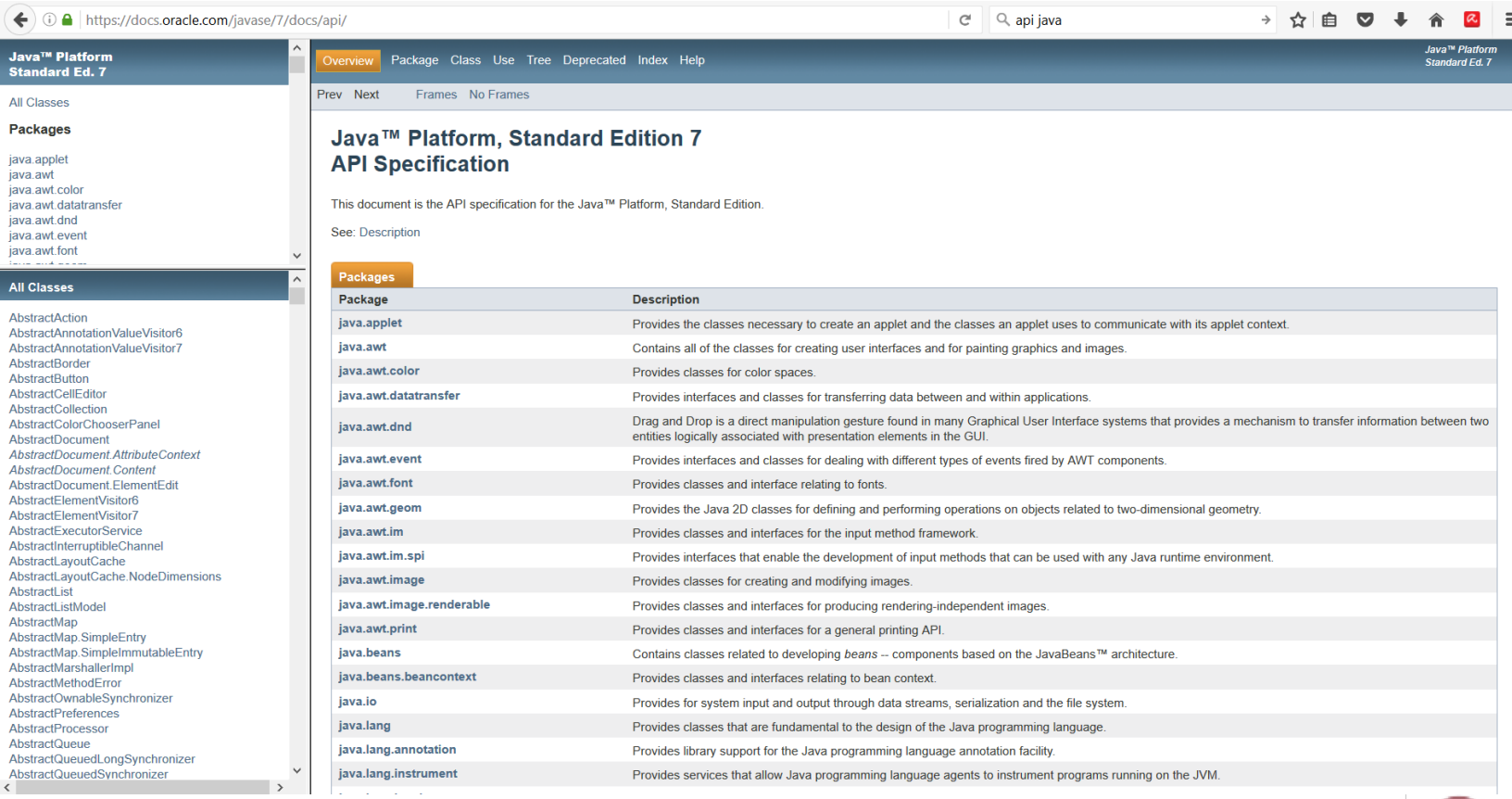
- In particolare le API Java mostrano tutte le classi offerte dal linguaggio e descrivono i metodi a disposizione in ogni classe.
  - Anche dette le librerie Java
- Le API per Java 8 sono disponibili all'URL https://docs.oracle.com/javase/8/docs/api/

API Java

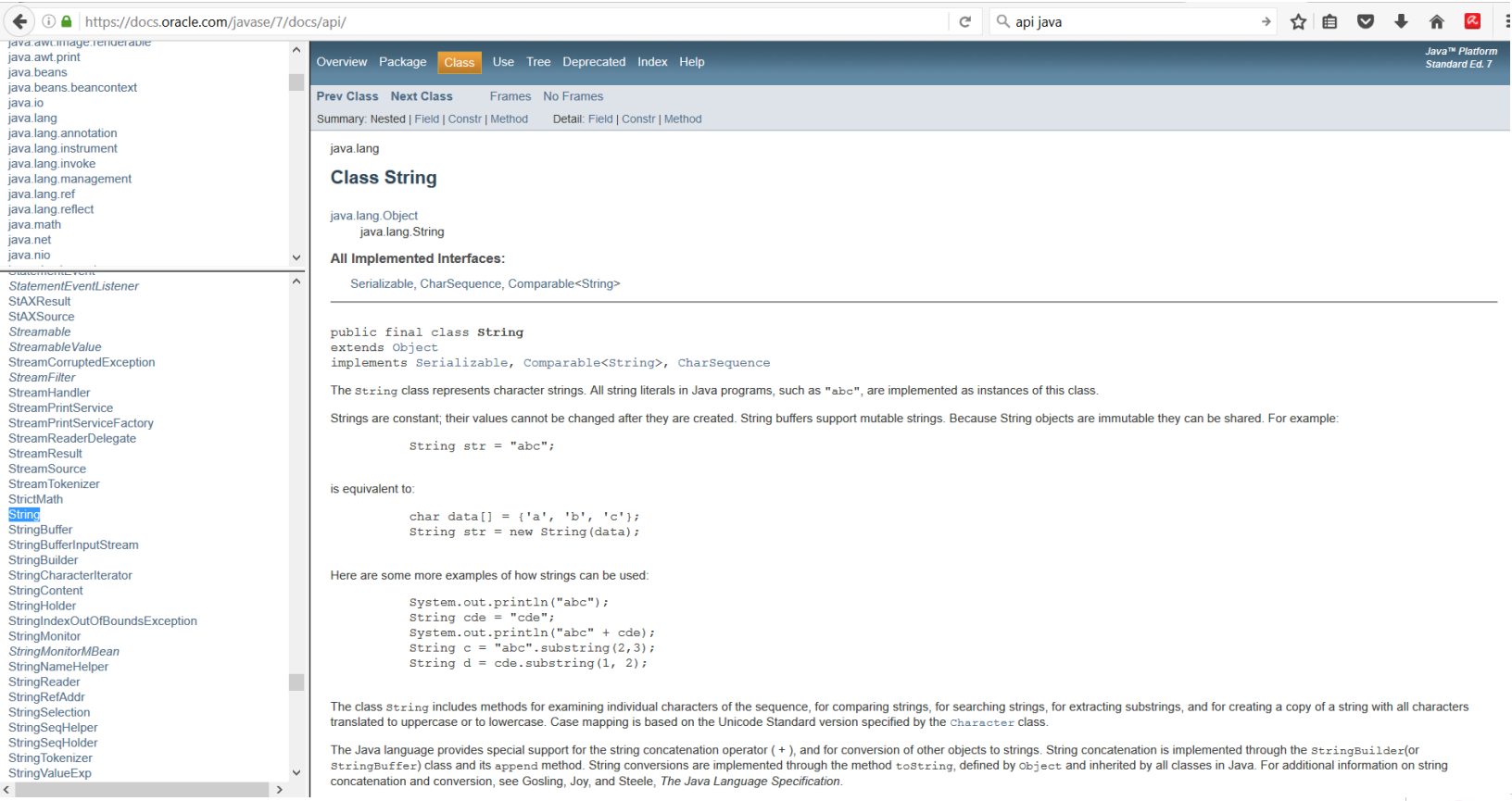
- Per ogni classe le API riportano e descrivono in dettaglio superclassi, sottoclassi, funzionalità ad alto livello della classe, campi, costruttori, metodi.

- In questa esercitazione ci concentreremo soprattutto sui metodi

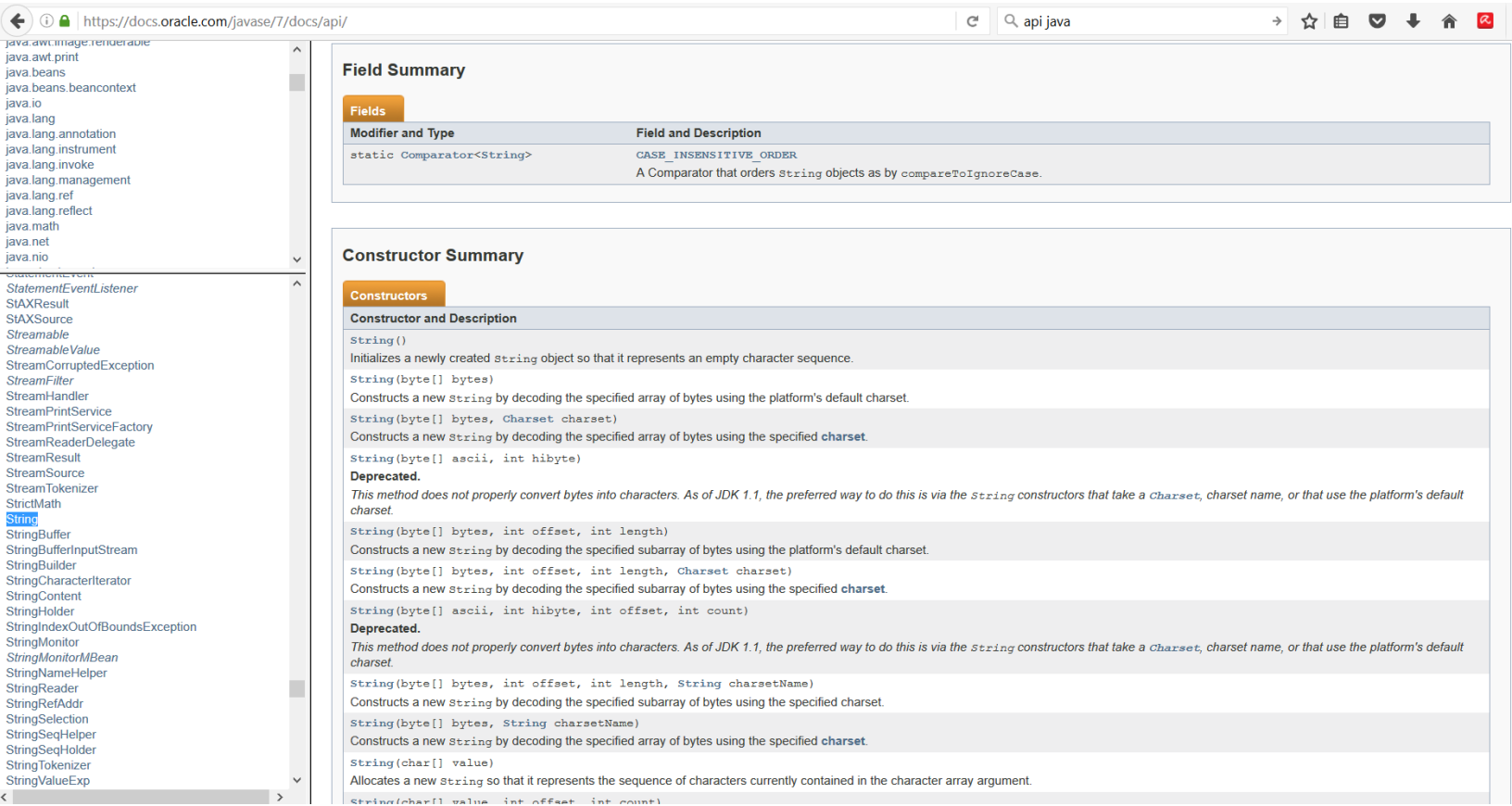
API Java


https://docs.oracle.com/javase/7/docs/api/
api java
Java™ Platform
Standard Ed. 7
All Classes
Packages
java.applet
java.awt
java.awt.color
java.awt.datatransfer
java.awt.dnd
java.awt.event
java.awt.font
All Classes
AbstractAction
AbstractAnnotationValueVisitor6
AbstractAnnotationValueVisitor7
AbstractBorder
AbstractButton
AbstractCellEditor
AbstractCollection
AbstractColorChooserPanel
AbstractDocument
AbstractDocument.AttributeContext
AbstractDocument.Content
AbstractDocument.ElementEdit
AbstractElementVisitor6
AbstractElementVisitor7
AbstractExecutorService
AbstractInterruptibleChannel
AbstractLayoutCache
AbstractLayoutCache.NodeDimensions
AbstractList
AbstractListModel
AbstractMap
AbstractMap.SimpleEntry
AbstractMap.SimpleImmutableEntry
AbstractMarshallerImpl
AbstractMethodError
AbstractOwnableSynchronizer
AbstractPreferences
AbstractProcessor
AbstractQueue
AbstractQueuedLongSynchronizer
AbstractQueuedSynchronizer
Overview
Package
Class
Use
Tree
Deprecated
Index
Help
Java™ Platform
Standard Ed. 7
Prev
Next
Frames
No Frames
Java™ Platform, Standard Edition 7
API Specification
This document is the API specification for the Java™ Platform, Standard Edition.
See: Description
Packages
Package
Description
java.applet
Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
java.awt
Contains all of the classes for creating user interfaces and for painting graphics and images.
java.awt.color
Provides classes for color spaces.
java.awt.datatransfer
Provides interfaces and classes for transferring data between and within applications.
java.awt.dnd
Drag and Drop is a direct manipulation gesture found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in the GUI.
java.awt.event
Provides interfaces and classes for dealing with different types of events fired by AWT components.
java.awt.font
Provides classes and interface relating to fonts.
java.awt.geom
Provides the Java 2D classes for defining and performing operations on objects related to two-dimensional geometry.
java.awt.im
Provides classes and interfaces for the input method framework.
java.awt.im.spi
Provides interfaces that enable the development of input methods that can be used with any Java runtime environment.
java.awt.image
Provides classes for creating and modifying images.
java.awt.image.renderable
Provides classes and interfaces for producing rendering-independent images.
java.awt.print
Provides classes and interfaces for a general printing API.
java.beans
Contains classes related to developing beans -- components based on the JavaBeans™ architecture.
java.beans.beancontext
Provides classes and interfaces relating to bean context.
java.io
Provides for system input and output through data streams, serialization and the file system.
java.lang
Provides classes that are fundamental to the design of the Java programming language.
java.lang.annotation
Provides library support for the Java programming language annotation facility.
java.lang.instrument
Provides services that allow Java programming language agents to instrument programs running on the JVM.

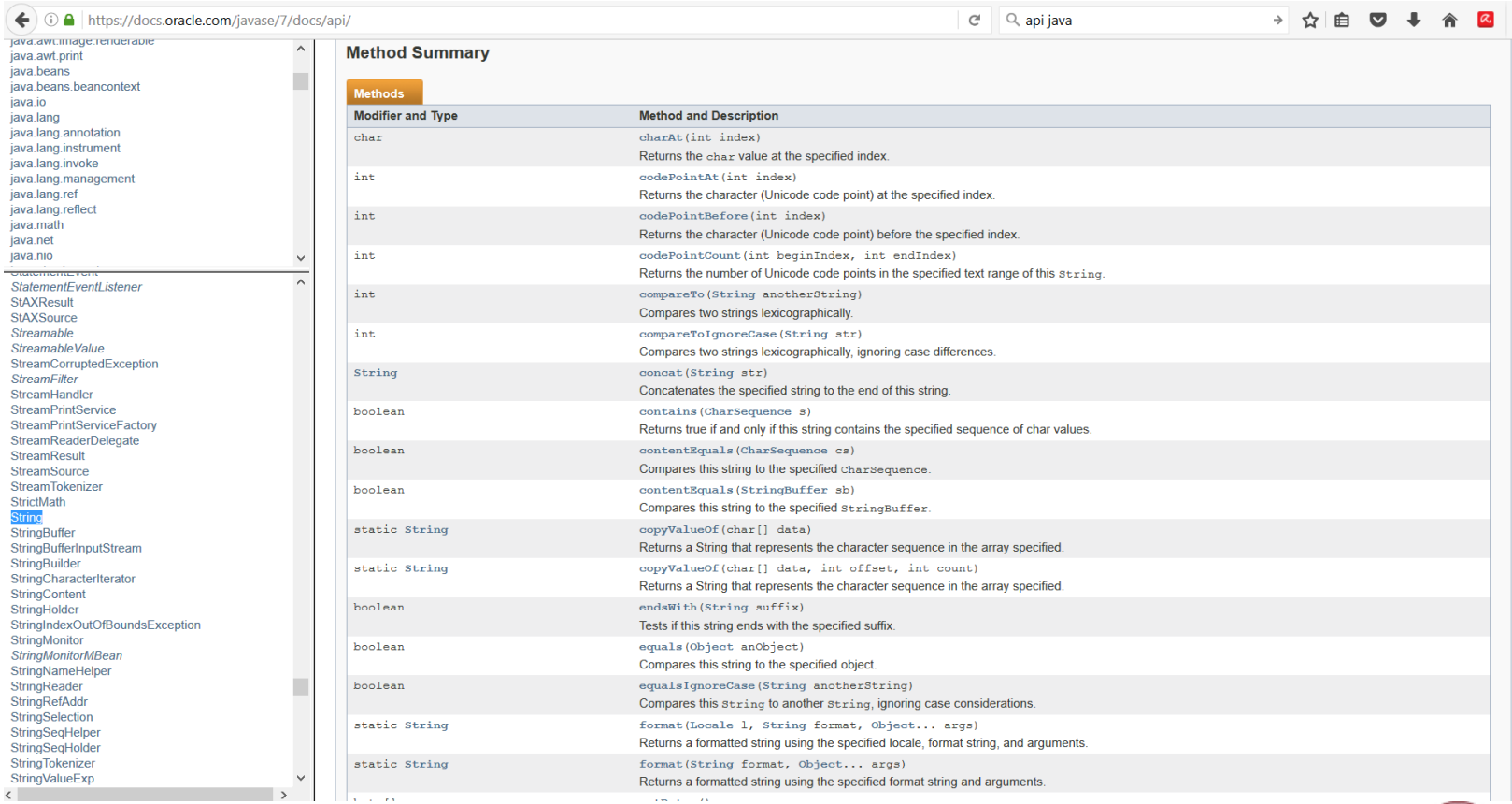
API Java
Esempio: la classe String



API Java
Esempio: la classe String



API Java
Esempio: la classe String



Intro

Nel risolvere gli esercizi

si faccia riferimento ai seguenti metodi della classe String:

- *toUpperCase()* per trasformare tutti i caratteri di una stringa in maiuscolo
- *substring(i,j)* per estrarre una sottostringa
- *contains(x)* per verificare che una stringa contenga la stringa x
- *lenght()* per ottenere la lunghezza della stringa
- *charAt(i)* per ottenere l'i-esimo carattere della stringa

Per utilizzare i metodi della classe JOptionPane, questa va importata dal package javax.swing

Per utilizzare i metodi della classe Scanner, questa va importata dal package java.util

Esercizio 1

Scrivere un programma Java che date due stringhe in input stampi *true* se una stringa è contenuta nell'altra, *false* altrimenti.

Implementare due varianti in cui le stringhe sono:

– Lette in input dalla console
– Lette in input da finestra di dialogo (JOptionPane)

Esercizio 2

Scrivere un programma Java che data una stringa interamente in minuscolo, stampi la stringa stessa con le lettere nelle posizioni dispari in maiuscolo.

Esempio: input = programmazione ; output= PrOgRaMmAzIoNe

Esercizio 3

Scrivere un programma Java che dato un intero rappresentante un anno, indichi se questo è bisestile.

Un anno è bisestile se è divisibile per 4 ma non per 100 oppure per 400.

Esercizio 4

Scrivere un programma Java che date due stringhe, stampi il rapporto tra le lunghezze delle stringhe stesse.

Esempio: input1 = ”lessie”, input2 = ”rex” il risultato deve essere 2,
input1 = ”rex”, input2 = ”lessie” il risultato deve essere 0.5

Esercizio 5

Scrivere un programma Java che chieda in input all'utente due stringhe, e stampi solo la stringa che abbia la somma dei codici ASCII delle proprie lettere maggiore e il valore di tale somma.

Esempio: se le stringhe passate sono "aaa" e "aab" il programma deve stampare "aab 292"

Esercizio 6

Scrivere un programma Java che calcoli l'area di triangoli.

Iterativamente il programma deve chiedere in input all'utente due valori numerici rappresentanti la base e l'altezza del triangolo corrente (leggendolo o da console o da JOptionPane) e deve scrivere il risultato in una finestra di output.
Il programma continua a chiedere valori e ad eseguire il calcolo finché non viene inserito un numero negativo.