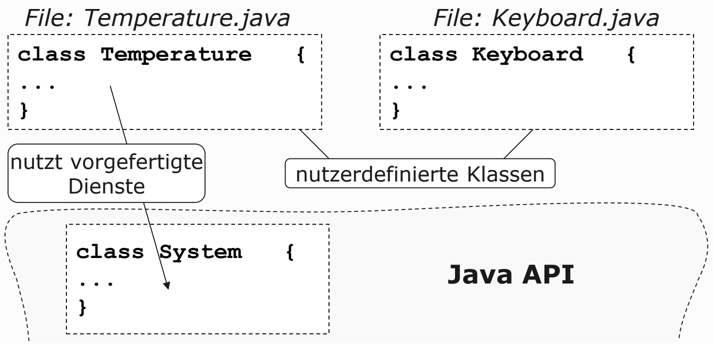
4. Standardbibliotheken: Java-API

Schwerpunkte

- Organisation des Java-API
- Beispielklassen: System, Printstream
- Import von Paketen und Klassen

API: Application Programming Interface

- Standardbibliothek
- Menge vordefinierter Komponenten, die zu jedem Java-(Compiler-)System gehören
- Kann von Nutzerprogrammen verwendet werden



File: Temperature.java

```
class Temperature   {
...
}
```

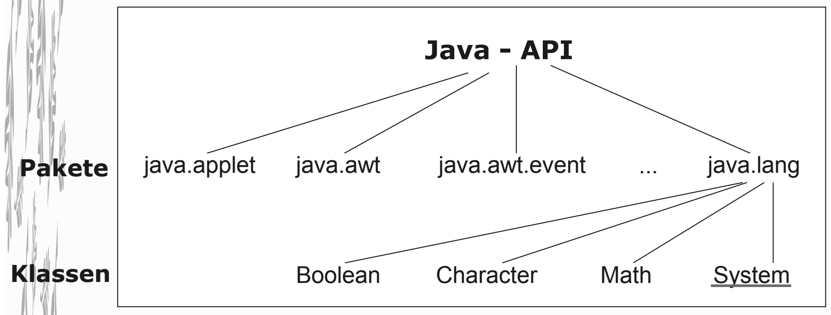
File: Keyboard.java

```
class Keyboard   {
...
}
```

```
class System   {
...
}
```

Organisation des Java-API

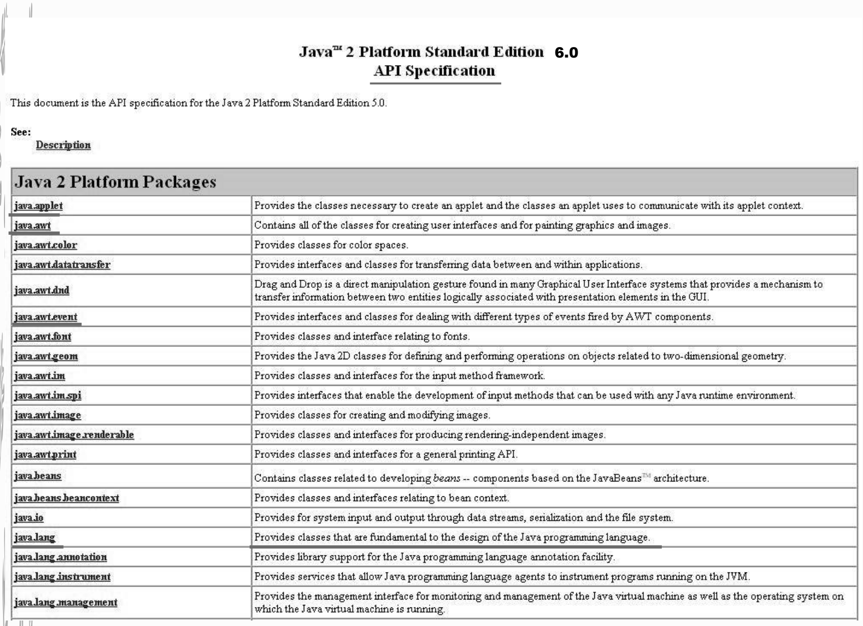
➔ GdP-Web-Seite: JDK 6 Dokumentation



Pakete: Sammlung von Klassen

Klassen: Softwarekomponenten

Paket-Name (z. B. java.awt.event) spiegelt Directory-Namen (Verzeichnis) wider: /java/awt/event



Java™ 2 Platform Standard Edition **6.0**
API Specification

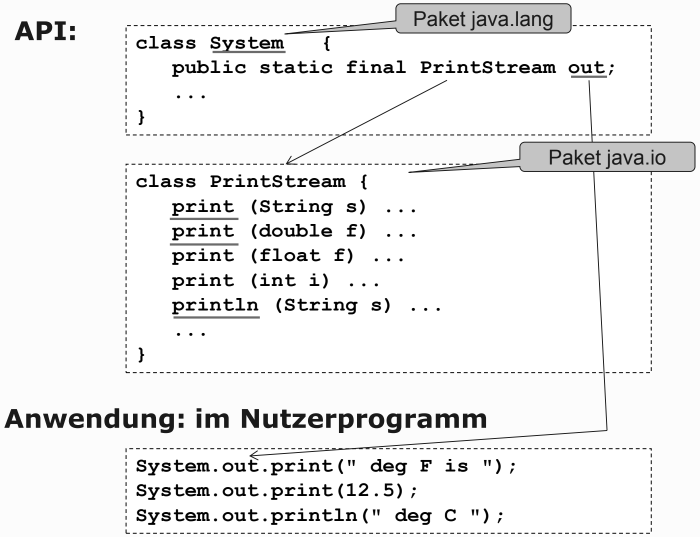
This document is the API specification for the Java 2 Platform Standard Edition 5.0.

See:
Description

Java 2 Platform Packages	
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.
java.lang.management	Provides the management interface for monitoring and management of the Java virtual machine as well as the operating system on which the Java virtual machine is running.

Gesamtdokument: 166 Pakete

Beispiel: Ausgabe mit API–Funktionen



API:

```
class System    {
   public static final PrintStream out;
   ...
}
```

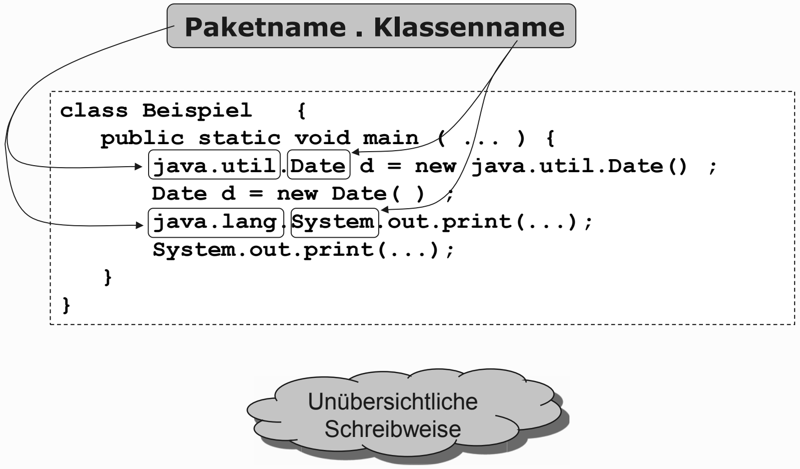
```
class PrintStream {
   print (String s) ...
   print (double f) ...
   print (float f) ...
   print (int i) ...
   println (String s) ...
   ...
}
```

Anwendung: im Nutzerprogramm

```
System.out.print(" deg F is ");
System.out.print(12.5);
System.out.println(" deg C ");
```

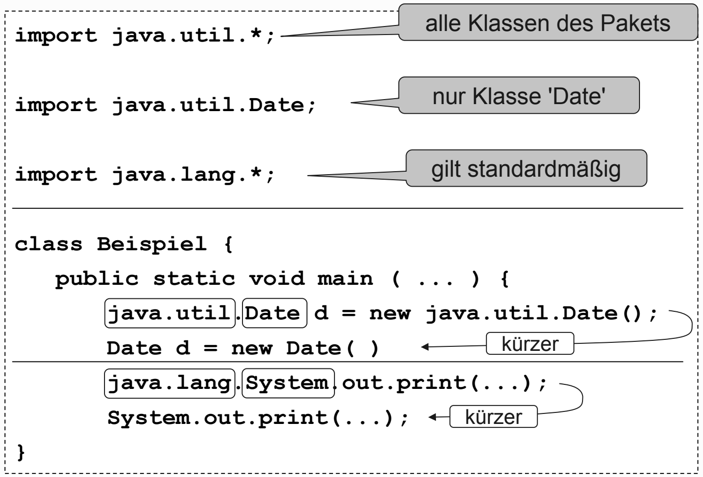
Import von Paketen und Klassen

Vollständiger Klassenname:



```
class Beispiel    {
   public static void main ( ... ) {
      java.util.Date d = new java.util.Date()  ;
      Date d = new Date( ) ;
      java.lang.System.out.print(...);
      System.out.print(...);
   }
}
```

Import-Anweisung: Paketname kann entfallen



```
import java.util.*;

import java.util.Date;

import java.lang.*;
```

```
class Beispiel {
   public static void main ( ... ) {
      java.util.Date d = new java.util.Date();
      Date d = new Date( )
      java.lang.System.out.print(...);
      System.out.print(...);
}
```