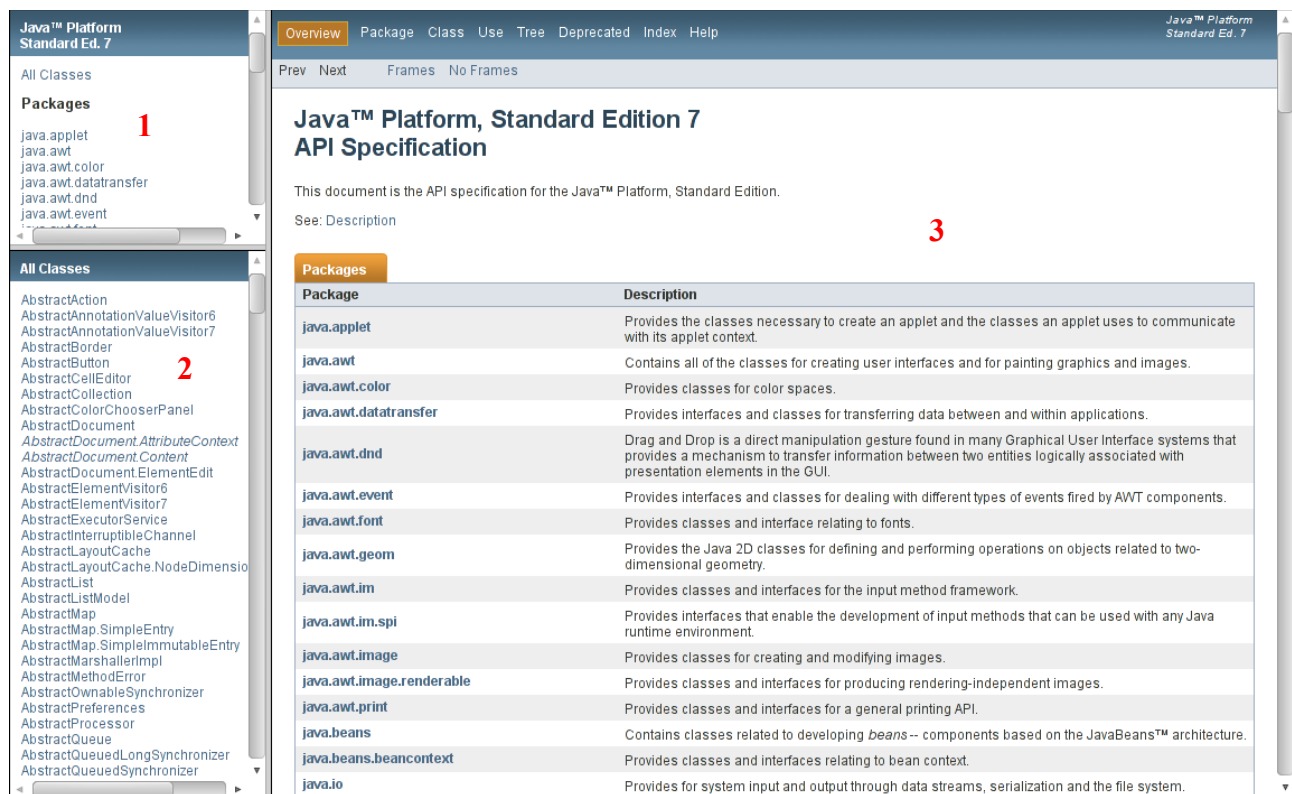


## Betreutes Programmieren 1

### C zu Java

#### Grundsätzliches:

Alle von Java unterstützten Funktionen und Klassen finden Sie in der dazugehörigen [API](#) (Applications Program Interface). Dazu sollte man wissen, dass alle *Funktionen* bestimmten *Klassen* zugeordnet sind. Hinzu kommt noch, dass diese Klassen wiederum in *Paketen* enthalten sind – allerdings ist dies für die Lösung der Aufgabe irrelevant und wird erst im Laufe des Semesters vertieft. Dennoch sollten Sie sofort damit anfangen, sich mit der API zu beschäftigen: Das Standardpaket, welches automatisch importiert ist, ist `java.lang`.



**Java™ Platform Standard Ed. 7**

Overview Package Class Use Tree Deprecated Index Help

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

Package	Description
<code>java.applet</code>	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
<code>java.awt</code>	Contains all of the classes for creating user interfaces and for painting graphics and images.
<code>java.awt.color</code>	Provides classes for color spaces.
<code>java.awt.datatransfer</code>	Provides interfaces and classes for transferring data between and within applications.
<code>java.awt.dnd</code>	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.
<code>java.awt.event</code>	Provides interfaces and classes for dealing with different types of events fired by AWT components.
<code>java.awt.font</code>	Provides classes and interface relating to fonts.
<code>java.awt.geom</code>	Provides the Java 2D classes for defining and performing operations on objects related to two-dimensional geometry.
<code>java.awt.im</code>	Provides classes and interfaces for the input method framework.
<code>java.awt.im.spi</code>	Provides interfaces that enable the development of input methods that can be used with any Java runtime environment.
<code>java.awt.image</code>	Provides classes for creating and modifying images.
<code>java.awt.image.renderable</code>	Provides classes and interfaces for producing rendering-independent images.
<code>java.awt.print</code>	Provides classes and interfaces for a general printing API.
<code>java.beans</code>	Contains classes related to developing <i>beans</i> -- components based on the JavaBeans™ architecture.
<code>java.beans.beancontext</code>	Provides classes and interfaces relating to bean context.
<code>java.io</code>	Provides for system input and output through data streams, serialization and the file system.

Java-API:

1: Alle Pakete; 2: Alle im gerade gewählten Paket enthaltenen Klassen; 3: Alle in der Gewählten Klasse enthaltenen Funktionen

## Wichtiges:

Den Java-Compiler **javac** finden Sie unter dem Pfad **C:\Program Files (x86)\Java\jdk<Version>\bin<sup>1</sup>**. Um diesen als temporäre Umgebungsvariable zu verwenden, gehen Sie wie folgt vor:

- Windows-Taste+R und „cmd“ (ohne „“) eingeben – alternativ öffnen Sie unter Programme die „Eingabeaufforderung“
- Geben Sie

```
set PATH="C:\Program Files (x86)\Java\jdk<Version>\bin";%PATH%
```

ein. Geben Sie anschließend

```
javac
```

ein, sollten Sie eine ähnliche Ausgabe wie diese erhalten:

```
javac
Usage: javac <options> <source files>
where possible options include:
-g                Generate all debugging info
-g:none          Generate no debugging info
-g:{lines,vars,source}  Generate only some debugging info
-nowarn          Generate no warnings
-verbose         Output messages about what the compiler is doing
...
```
- Wenn Sie die Eingabeaufforderung schließen und wieder öffnen, müssen Sie die obige Anweisung erneut ausführen.

---

<sup>1</sup> <Version> ist derzeit auf den Rechnern im CIP-Pool 1.7.0\_03, kann sich aber jederzeit ändern. Sehen Sie also auf der Festplatte nach, um den Namen des Unterverzeichnisses herauszufinden. Die Umgebungsvariable sollte allerdings auf den Rechnern in den CIP-Pools bereits gesetzt sein, sodass Sie sofort loslegen können.

## Aufgabe:

1. Schreiben Sie ein HelloWorld - Programm und führen Sie es mit Java aus. Dies geht so:
  1. Schreiben Sie folgenden Code in eine Text-Datei (tun Sie sich selber den Gefallen, den Code tatsächlich abzutippen!!)

```
public class HelloWorld
{
    public static void main (String args[])
    {
        System.out.println("Hello World!");
    }
}
```
  2. Speichern Sie die Datei als **HelloWorld.java** ab (der Dateiname MUSS gleich dem **Klassennamen** sein!!)
  3. Kompilieren Sie das Programm über die Kommandozeile mit

```
javac HelloWorld.java
```
  4. Führen Sie anschließend das Programm mittels

```
java HelloWorld
```

aus.
2. Der folgende Code ist in C geschrieben und liest von Kommandozeile beliebig viele Parameter ein, um sie dann hinsichtlich ihrer Palindromeigenschaft zu überprüfen. Ein Palindrom ist ein Wort, das vorwärts und rückwärts gelesen gleich ist. Beispiele: rotor, u4gg4u, ...

```
#include <stdio.h>
int main (int argc, char** argv)
{
    int i=0;
    if(argc==1)
        printf("\nKeine Parameter übergeben!\n");
    else
    {
        int j;
        for(i=1;i<argc;i++)
        {
            j=palindrom(argv[i]);
            if(j==0)
                printf("\n%s ist kein Palindrom\n",argv[i]);
            else
                printf("\n%s ist ein Palindrom\n",argv[i]);
        }
    }
    return 1;
}

int palindrom(char string[])
{
    int laenge = stringlaenge(string);
    int vorne = 0;
    int hinten = laenge-1;
    while(string[vorne]==string[hinten] && vorne<=hinten)
    {
        vorne++;
        hinten--;
    }
    if(vorne>hinten)
        return 1;
    else
        return 0;
}

int stringlaenge(char string[])
{
    int i=0;
    while(string[i]!='\0')
        i++;
    return i;
}
```

Kompilieren und Ausführen:

```
gcc -o palindrom palindrom.c
palindrom test otto rotor maus
```

Ausgabe:

```
test ist kein Palindrom
otto ist ein Palindrom
rotor ist ein Palindrom
maus ist kein Palindrom
```

Ihre Aufgabe ist es nun, diesen Code nach Java zu übersetzen. Dazu können Sie wie folgt vorgehen:

1. Öffnen Sie eine leere Textdatei
2. Kopieren Sie den C-Code in diese Datei
3. Löschen Sie die **include**-Anweisung  
*stdio.h* ist die Bibliothek für Ein- und Ausgaben auf Kommandozeile in C. In Java sind diese Anweisungen in der System-Klasse gekapselt (siehe HelloWorld-Programm)
4. Schreiben Sie in die erste Zeile nun:

```
public class Palindrom {
```

schließen Sie die Klammer am Ende der Datei. Und speichern Sie die Datei als: **Palindrom.java**.

In Java **müssen** alle Funktionen in einer Klasse gekapselt sein. Auch die *main*-Funktion.

5. Passen Sie nun den Prototyp der **main**-Funktion an:  
**public static void main (String args[])**
6. Die **if**-Anweisung **if (argc==1)** überprüft, ob Kommandozeilenparameter übergeben wurden. Allerdings findet sich im Prototypen der Java-main-Funktion kein solcher Parameter **argc**. Stattdessen erhält man ein Array mit Namen **args** vom Typ **String** – ein String-Array also. Eigentlich genau das gleiche, wie in C, nur dass in Java der Datentyp **String** bereits existiert und nicht als Array von **chars** implementiert ist. Gut, damit wäre das „Verschwinden“ von **char\*\* argv** erklärt. Aber wie erhält man nun die Anzahl der übergebenen Parameter? **args** ist ein Array und von jedem Array kann man in Java die Länge abfragen: **args.length**. Zu beachten ist allerdings, dass in Java als 0-tes Argument **nicht** der Programmname übergeben wird. Also muss **args.length==0** überprüft werden.
7. Ersetzen Sie nun **printf** durch die im HelloWorld-Programm gelernte Anweisung.
8. Im **else**-Zweig müssen Sie erstmal den **for**-Schleifenrumpf anpassen indem Sie **i** analog zu oben bei 0 starten lassen und **argc** durch **args.length** ersetzen.
9. Die **palindrom**-Funktion erwartet einen **String**. **args** ist ein Array von **Strings**, also wird **args[i]** übergeben.
10. Um **System.out.println** eine Variable zu übergeben, brauchen Sie nicht wie in C Formatierungszeichen zu benutzen, sondern es reicht ein „+“:  
**System.out.println(args[i]+\" ist kein Palindrom\");**
11. Die **main**-Funktion in Java ist **void** und benötigt somit keine **return**-Anweisung!
12. Nun muss noch der Prototyp der **palindrom**-Funktion angepasst werden:  
**int palindrom(String string)**  
Alternativ gibt es in Java auch einen Datentyp **boolean**, welcher Wahrheitswerte speichert. Eine Variable vom Typ **boolean** kann nur die Werte **true** oder **false** haben. Also sollte der Prototyp nochmals geändert werden:  
**boolean palindrom(String string)**  
Achtung: Ändern Sie **j** in **main** ebenso zu **boolean** und testen Sie in der **if**-Abfrage auf **false**.  
Auch wenn das noch etwas unverständlich wirkt: Von der Haupt-Klasse können keine Objekte erzeugt werden, daher müssen die Methoden statisch sein. Abgesehen davon sollte man immer zu jeder Methode einen Sichtbarkeit<sup>2</sup> angeben. Dadurch erhält man also den Prototypen:  
**public static boolean palindrom(String string)**
13. **String** ist eine Klasse, die Methoden beinhaltet. Erinnern Sie sich etwas an die Kapselung von Methoden in verschiedenen Dateien, deren Header man in C einbindet. So ähnlich läuft das mit den Klassen ab. Wenn man also nun die Instanz einer Klasse<sup>3</sup> hat, so kann man auf deren Methoden mittels „.“ zugreifen. Die **String**-Klasse hat eine Methode **int length()**, welche die Länge eines Strings zurückgibt. Somit kann also die Funktion **stringlaenge** gelöscht und die Länge eines Strings **string** mittels **string.length()** abgefragt bzw. einer Variablen zugewiesen werden.
14. So, in der **while**-Anweisung müssen Sie nun Buchstaben vergleichen, aber Sie haben ja nicht wie in C ein **char**-Array, sondern eine Instanz **string** der Klasse **String**, d.h. Sie benötigen, wie bei der Abfrage der Länge, eine Methode, welche Ihnen den Character eines Strings an einer bestimmten Position zurückgibt. Diese Methode heisst **char charAt(int index)**, erwartet als Übergabeparameter eine Variable vom Typ **int** und gibt ein **char** zurück. Aufgerufen wird diese wieder mittels „.“ auf **string**:  
**string.charAt(vorne)**
15. Als letzten Punkt müssen Sie nun noch die Rückgabewerte von 1 und 0 auf **true** und **false** ändern.
16. Kompilieren Sie nun den Code mittels  
**javac Palindrom.java**  
und führen Sie ihn mit  
**java Palindrom test otto rotor maus**  
aus.

---

<sup>2</sup> Das Wörtchen **public** bezeichnet so eine Sichtbarkeit. Daneben gibt es noch **private** und **protected**.

<sup>3</sup> Gemeint ist damit eine initialisierte Variable des betreffenden Typs.