

Der Programmaufrufstack

Unter dem (Programm)Aufrufstapel (englisch call stack, procedure stack) versteht man einen besonders genutzten Stapelspeicher, der zur Laufzeit eines Programms den Zustand der gerade aufgerufenen Unterprogramme enthält.

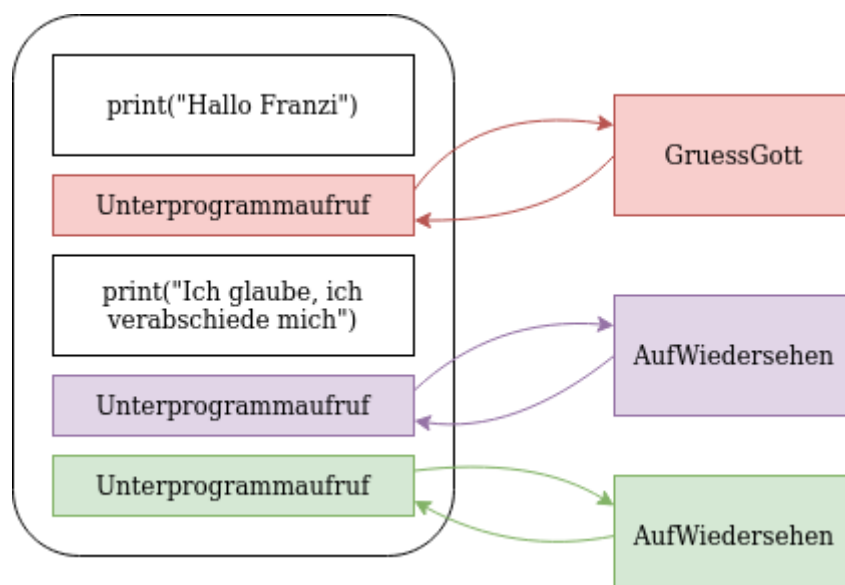
Was bedeutet das? Wir betrachten ein Beispiel:

```
public class Hallo
{
    public Hallo(String name)
    {
        System.out.println("Hallo " + name + "!");
        GruessGott(name);
        System.out.println("Ich glaube, gleich verabschiede ich mich...");
        AufWiedersehen();
        AufWiedersehen();
    }

    public void GruessGott(String name) {
        System.out.println("Grueß Gott, " + name + "!");
    }

    public void AufWiedersehen() {
        System.out.println("Ciao!");
    }
}
```

Beim Aufruf des Konstruktors werden also zwei weitere Methoden, also "Unterprogramme", aufgerufen, der Ablauf sieht etwa wie folgt aus.



Dabei fällt auf, dass der Programmablauf nach jedem Aufruf eines Unterprogramms zu der Stelle zurückkehren muss, von der dieses aufgerufen wurde. Das wird dadurch realisiert, dass beim Aufruf einer Methode/Funktion auf dem **Programmaufrufstack** für jede Methode ein Speicherbereich reserviert wird, in dem alle relevanten Infos festgehalten werden, unter anderem auch die Rücksprungadresse, also die Stelle, von der aus die Methode gestartet wurde.

Was passiert	Wie sieht der Stack aus?								
Der Konstruktor wird mit dem Argument name aufgerufen. Auf dem Stack wird Speicher für den Konstruktoraufruf reserviert. Es gibt keine Rücksprungadresse	<table> <tr> <td colspan="2">Konstruktor</td></tr> <tr> <td>name = "Franzi"</td><td>ruecksprungadresse=keine</td></tr> </table>	Konstruktor		name = "Franzi"	ruecksprungadresse=keine				
Konstruktor									
name = "Franzi"	ruecksprungadresse=keine								
Aus dem Konstruktor heraus wird die Methode GruessGott mit dem Argument name aufgerufen. Die Stelle für die Rückkehr nach Bearbeitung von GruessGott wird vermerkt.	<table> <tr> <td colspan="2">GruessGott</td></tr> <tr> <td>name = "Franzi"</td><td>ruecksprungadresse = 0001</td></tr> <tr> <td colspan="2">Konstruktor</td></tr> <tr> <td>name = "Franzi"</td><td>ruecksprungadresse=keine</td></tr> </table>	GruessGott		name = "Franzi"	ruecksprungadresse = 0001	Konstruktor		name = "Franzi"	ruecksprungadresse=keine
GruessGott									
name = "Franzi"	ruecksprungadresse = 0001								
Konstruktor									
name = "Franzi"	ruecksprungadresse=keine								
Die Bearbeitung von GruessGott ist beendet. Der zugehörige Speicherbereich im Call-Stack wird freigegeben. Im Konstruktor wird die Verabschiedung angekündigt.	<table> <tr> <td colspan="2">Konstruktor</td></tr> <tr> <td>name = "Franzi"</td><td>ruecksprungadresse=keine</td></tr> </table>	Konstruktor		name = "Franzi"	ruecksprungadresse=keine				
Konstruktor									
name = "Franzi"	ruecksprungadresse=keine								
Aus dem Konstruktor heraus wird die Methode AufWiedersehen aufgerufen. Keine Argumente, Rücksprungadresse wird vermerkt.	<table> <tr> <td colspan="2">AufWiedersehen</td></tr> <tr> <td>keine Argumente</td><td>ruecksprungadresse = 0002</td></tr> <tr> <td colspan="2">Konstruktor</td></tr> <tr> <td>name = "Franzi"</td><td>ruecksprungadresse=keine</td></tr> </table>	AufWiedersehen		keine Argumente	ruecksprungadresse = 0002	Konstruktor		name = "Franzi"	ruecksprungadresse=keine
AufWiedersehen									
keine Argumente	ruecksprungadresse = 0002								
Konstruktor									
name = "Franzi"	ruecksprungadresse=keine								
Der erste Aufruf von AufWiedersehen ist beendet. Der Programmablauf kehrt zum Konstruktor zurück.	<table> <tr> <td colspan="2">Konstruktor</td></tr> <tr> <td>name = "Franzi"</td><td>ruecksprungadresse=keine</td></tr> </table>	Konstruktor		name = "Franzi"	ruecksprungadresse=keine				
Konstruktor									
name = "Franzi"	ruecksprungadresse=keine								
Aus dem Konstruktor heraus wird die Methode AufWiedersehen erneut aufgerufen. Keine Argumente, Die neue Rücksprungadresse wird vermerkt.	<table> <tr> <td colspan="2">AufWiedersehen</td></tr> <tr> <td>keine Argumente</td><td>ruecksprungadresse = 0003</td></tr> <tr> <td colspan="2">Konstruktor</td></tr> <tr> <td>name = "Franzi"</td><td>ruecksprungadresse=keine</td></tr> </table>	AufWiedersehen		keine Argumente	ruecksprungadresse = 0003	Konstruktor		name = "Franzi"	ruecksprungadresse=keine
AufWiedersehen									
keine Argumente	ruecksprungadresse = 0003								
Konstruktor									
name = "Franzi"	ruecksprungadresse=keine								

Was passiert	Wie sieht der Stack aus?				
Der zweite Aufruf von AufWiedersehen ist beendet. Der Programmablauf kehrt zum Konstruktor zurück.	<table> <tr> <td colspan="2">Konstruktor</td></tr> <tr> <td>name = "Franzi"</td><td>ruecksprungadresse=keine</td></tr> </table>	Konstruktor		name = "Franzi"	ruecksprungadresse=keine
Konstruktor					
name = "Franzi"	ruecksprungadresse=keine				
Der Konstruktoraufruf ist beendet. Der Call-Stack ist leer.	<table> <tr> <td colspan="2"></td></tr> <tr> <td></td><td></td></tr> </table>				



(A1)

Welche Informationen kannst du den folgenden Darstellungen des Programmaufrufstacks entnehmen?

Quadrat	
a=5	ruecksprungadresse = 0003
Konstruktor	
keine Argumente	ruecksprungadresse=keine

Logarithmus	
c=5	ruecksprungadresse = 0025
Wurzel	
b=25	ruecksprungadresse = 0018
Quadrat	
a=5	ruecksprungadresse = 0007
Konstruktor	
keine Argumente	ruecksprungadresse=keine

From:
<https://info-bw.de/> -

Permanent link:
<https://info-bw.de/faecher:informatik:oberstufe:algorithmen:rekursion:programmaufrufstack:start?rev=1642065492>

Last update: **13.01.2022 09:18**

