

Systemprogrammierung – Probeklausur

Die Klausur besteht aus 7 Aufgaben, die zusammen 100 Punkte ergeben.

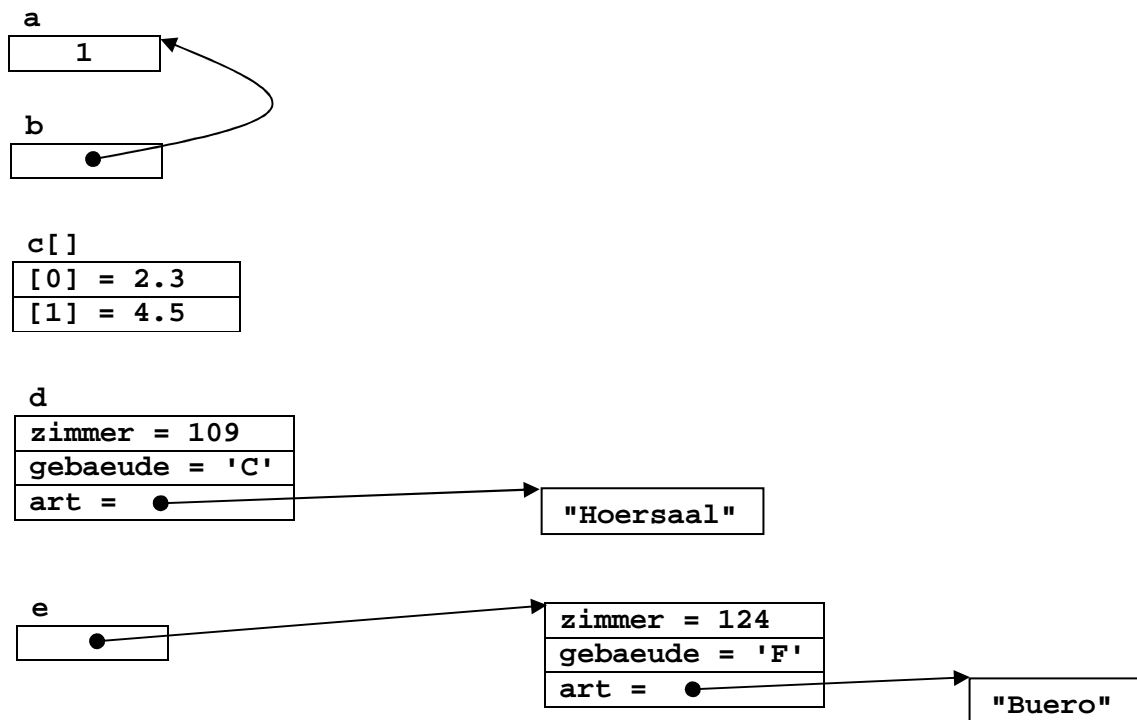
Aufgabe 1: Datentypen und Variablen (26 Punkte)

In der Vorlesung haben wir Speicherbelegungen mit Rechtecken grafisch dargestellt:

- Wurde der Speicher per Variablendefinition reserviert, stand über dem Rechteck der Variablenname, bei Feldern zusätzlich [].
- Bei Speicher, der nicht per Variablendefinition reserviert wurde, fehlte der Name über dem Rechteck, z.B. bei Stringliteralen und dynamisch allokiertem Heapspeicher.
- Bei Feldern und Strukturen wurde das Rechteck unterteilt und jeweils der Index des Feldelements bzw. der Name der Strukturkomponente angegeben
- Die Adresse eines Speicherbereichs haben wir als Pfeil auf den Speicherbereich gezeichnet.

- a) Erstellen Sie mit C-Definitionen und -Anweisungen die unten dargestellte Hauptspeicherbelegung:
- deklarieren Sie zuerst eventuelle benutzerdefinierte Typen
 - definieren Sie dann die fünf Variablen ohne Initialisierung (leiten Sie dazu die Typen der Variablen aus den C-Literalen und den Pfeilen in der Grafik ab)
 - weisen Sie zum Schluss allen Variablen die in der Grafik gezeigten Werte zu (lassen Sie dabei die Fehlerbehandlung für dynamischer Speicherreservierungen weg)

(20 Punkte)



- b) Wieviel Speicherplatz in Anzahl Byte belegt die Variable **d** auf dem Stack?

Geben Sie für die Architekturen ILP32 und LP64 jeweils die Mindestgröße ohne Ausrichtung sowie die tatsächliche Größe mit Ausrichtung der Datentypen an. Begründen Sie Ihre Angaben.

(6 Punkte)

Aufgabe 2: Übersetzungseinheiten (29 Punkte)

- a) Erstellen Sie eine C-Übersetzungseinheit, die der folgenden Java-Klasse **Quadrat** entspricht. Geben Sie die erforderlichen Dateien mit vollständigem Inhalt an. (8 Punkte)

```
public final class Quadrat {
    private Quadrat() { }

    public static double flaeche(final double seitenlaenge) {
        return zumquadrat(seitenlaenge);
    }

    private static double zumquadrat(final double d) {
        return d * d;
    }
}
```

- b) Die folgende Java-Klasse **Wuerfel** verwendet die Java-Klasse **Quadrat** aus a). Erstellen Sie eine entsprechende C-Übersetzungseinheit, die Ihre in a) erstellte Übersetzungseinheit verwendet. Geben Sie auch hier die erforderlichen Dateien mit vollständigem Inhalt an. (9 Punkte)

```
public final class Wuerfel {
    private Wuerfel() { }

    public static double oberflaeche(final double kantenlaenge) {
        return Quadrat.flaeche(kantenlaenge) * 6;
    }

    public static double volumen(final double kantenlaenge) {
        return Quadrat.flaeche(kantenlaenge) * kantenlaenge;
    }
}
```

- c) Die folgende Java-Klasse **WuerfelTest** testet die Java-Klasse **Wuerfel** aus b). Erstellen Sie eine entsprechende C-Übersetzungseinheit, die Ihre in b) erstellte Übersetzungseinheit testet. Geben Sie wiederum die erforderlichen Dateien mit vollständigem Inhalt an. (9 Punkte)

```
import static java.lang.System.out;

public final class WuerfelTest {
    private WuerfelTest() { }

    public static void main(final String[] args) {
        double k = Double.parseDouble(args[0]);
        double f = Wuerfel.oberflaeche(k);
        double v = Wuerfel.volumen(k);
        out.printf("Kantenlaenge %f, Oberflaeche %f, Volumen %f\n",
            k, f, v);
    }
}
```

*Hinweis: als Entsprechung zur Java-Klassenmethode `double Double.parseDouble(String)` gibt es bei C eine Funktion mit dem Prototyp `double atof(const char *)` in `stdlib.h`*

- d) Wird das Java-Programm aus c) ohne Kommandozeilenargumente gestartet, bricht es mit einer nicht gefangenen Ausnahme **ArrayIndexOutOfBoundsException** ab. Wie verhält sich das entsprechende C-Programm, wenn es wie das Java-Programm keine Fehlerbehandlung für zu wenige Kommandozeilenargumente enthält? Begründen Sie Ihre Antwort. (3 Punkte)

Hinweis: überlegen Sie, was `argc` und `argv` bei einem Aufruf ohne Kommandozeilenargumente enthalten und was dann an die Funktion `atof` übergeben wird.

Aufgabe 3: Strings (12 Punkte)

- a) In der C-Standardbibliothek gibt es die Funktion `size_t strlen(const char *s)`. Geben Sie eine eigene Implementierung der Funktion an. (6 Punkte)
- b) Schreiben Sie eine C-Anweisungsfolge, die der folgenden Java-Anweisungsfolge entspricht:

```
String s = "Halli";  
String t = "Hallo";  
String st = s + t;
```

Bei dynamischer Speicherreservierung dürfen Sie die Fehlerbehandlung weglassen. (6 Punkte)

Aufgabe 4: Make (12 Punkte)

*LaTeX ist eine Auszeichnungssprache zum Erstellen formatierter Dokumente. Mit einem beliebigen Texteditor wird eine Quelldatei mit Endung **.tex** erstellt, die den Inhalt des Dokuments mit den Auszeichnungen enthält.*

Beispiel:

```
% hello.tex - Our first LaTeX example!  
\documentclass{article}  
\begin{document}  
Hello World!  
\end{document}
```

Das Linux-Kommando **pdflatex** erstellt aus der **.tex**-Quelle ein formatiertes **.pdf**-Dokument. Dabei entstehen zusätzlich Hilfsdateien mit den Endungen **.aux** und **.log**.

Beispiel:

```
pdflatex hello.tex
```

 erstellt **hello.pdf** und als Nebenprodukt **hello.aux** sowie **hello.log**

Erstellen Sie ein **Makefile** mit folgendem Inhalt:

- einer Musterregel, die aus einer **.tex**-Datei die formatierte **.pdf**-Datei erstellt
- einer Regel mit Pseudoziel **all**, die **hello.pdf** erstellt
- einer Regel mit Pseudoziel **clean**, die alle bei **all** erstellten Dateien löscht

Halten Sie die in der Vorlesung besprochenen Stilregeln ein, insbesondere die Stilregeln zur Verwendung von Variablen.

Aufgabe 5: Fehlersuche (12 Punkte)

- a) In der Vorlesung wurden einige wichtige Arten von Laufzeitfehlern bei Programmen genannt. Welche waren das? (4 Punkte)
- b) Geben Sie für jede Fehlerart aus a) an, wie Sie die Fehlerursache suchen würden und welche Werkzeuge Sie dabei einsetzen würden. Begründen Sie die Werkzeugwahl. (8 Punkte)

Aufgabe 6: POSIX (3 Punkte)

Die Java-Bibliothek meldet Fehlersituationen, indem sie Ausnahmen wirft. Für unterschiedliche Fehler gibt es unterschiedliche Ausnahmeklassen. In C gibt es aber keine Ausnahmebehandlung. Wie melden POSIX-Funktionen Fehlersituationen und wie muss an der Aufrufstelle damit umgegangen werden?

Aufgabe 7: C++ (6 Punkte)

- a) Nennen Sie drei wichtige Konzepte, die C++ zu C hinzugefügt hat. (3 Punkte)
- b) Nennen Sie drei wichtige konzeptionelle Unterschiede zwischen Java und C++. (3 Punkte)