

# Klausur im WS2016/17

## Modul Informationstechnik



Institut für Technik der Informationsverarbeitung – ITIV  
Prof. Dr.-Ing. E. Sax

### Modul Informationstechnik

Datum: 28.02.2017

Name:

Matr. Nr.:

ID:

Hörsaal:

Platz:

## Hinweise zur Klausur

### Hilfsmittel:

- Erlaubte Hilfsmittel sind Lineal und eine Notizen-/Formelsammlung, handgeschrieben DIN A4 Blatt (zweiseitig beschrieben).
- Verwenden Sie zum Bearbeiten der Aufgaben nur dokumentenechte Schreibgeräte – keinen Bleistift, keine Rotstifte!
- Alle nicht genannten Hilfsmittel sind untersagt. Dies beinhaltet jegliche Kommunikation mit anderen Personen sowie die Benutzung eines Taschenrechners.

### Klausurdauer:

Die Prüfungsdauer für die Klausur beträgt 120 Minuten.

### Klausurunterlagen:

Die Klausurunterlagen bestehen aus insgesamt 26 Seiten Aufgabenblättern (inklusive dieses Titelblatt, 8 Aufgabenblöcke und 1 zusätzliches Lösungsblatt).

**Bitte prüfen Sie vor der Bearbeitung der Aufgaben auf jeder Seite oben Ihre Matrikelnummer sowie Ihre ID und zusätzlich Ihren Namen auf der ersten Seite.**

Falls Sie zusätzliche Blätter zur Lösung der Aufgaben benötigen, fragen Sie nach zusätzlichem Lösungspapier bei der Aufsicht. Vermeiden Sie generell das Beschreiben der Rückseiten. Die Verwendung von eigenen Blättern ist nicht erlaubt. Geben Sie zu jeder Aufgabe ein detaillierten Rechenweg an. Lösungen ohne Rechenweg können trotz richtigem Ergebnis zu Punktabzug führen.

### Klausurabgabe:

In den letzten 30 Minuten der Klausur ist eine vorzeitige Abgabe der Klausur nicht möglich. Am Ende der Klausur bleiben Sie bitte sitzen. Alle Aufgaben- und Lösungsblätter sowie dieses Deckblatt sind in den ausgehändigten Umschläge abzugeben. Diese werden von der Aufsicht eingesammelt.

	Seite	≈ Pkt. [%]	Punkte
Aufgabe 1: Rechnerarchitekturen	2	16	
Aufgabe 2: Software Engineering	6	10	
Aufgabe 3: Programmiersprachen	9	12	
Aufgabe 4: Objektorientierung in C++	11	11	
Aufgabe 5: Datenstrukturen	15	12	
Aufgabe 6: Algorithmenanalyse	19	13	
Aufgabe 7: Hardwarenahe Programmierung	21	10	
Aufgabe 8: Projektmanagement	23	12	
			Σ

# Aufgabe 1: Rechnerarchitekturen

## Aufgabe 1.1: Verständnisfragen

A) Erklären Sie die Begriffe zeitliche und räumliche Parallelität in Bezug auf die Befehlsabarbeitung und geben Sie jeweils ein Beispiel für eine konkrete Maßnahme an.

---

---

---

---

---

B) Was versteht man in Bezug auf einen von-Neumann Rechner unter dem Begriff "von-Neumann Flaschenhals"? Wie wird dieser Flaschenhals in einem Harvard-Rechner entgegengewirkt?

---

---

---

---

C) Nennen Sie einen Vorteil und einen Nachteil von CISC Rechnerarchitekturen gegenüber RISC Rechnerarchitekturen.

---

---

D) Gegeben sei folgendes 32 Bit Datenwort:  $(2B\ 3C\ 4D\ 5E)_H$ . Füllen Sie Tabelle 1.1 entsprechend des Big Endian und Little Endian Speicherformats aus, beginnend bei Adresse  $10000_b$ . Nehmen Sie an, dass pro Speicheradresse genau 1 Byte gespeichert wird.

	Big Endian	Little Endian
Adresse	Hex	Hex
$10000_b$		
$10001_b$		
$10002_b$		
$10003_b$		

Tabelle 1.1: Endianness

### Aufgabe 1.2: Caches - Verständnisfragen

A) Nennen und erläutern Sie einen Vorteil und einen Nachteil der Cacheorganisation voll-assoziativ gegenüber direkt-abbildend. Gehen Sie von einem Cache mit  $m$  Cachezeilen aus und verwenden Sie diese Angabe zur Quantifizierung Ihrer Erläuterungen.

---



---



---



---

B) Erläutern Sie die Funktion der folgenden 4 Begriffe in der Cache-Organisation: Offset, Index, Tag, Valid-Bit

---



---



---



---

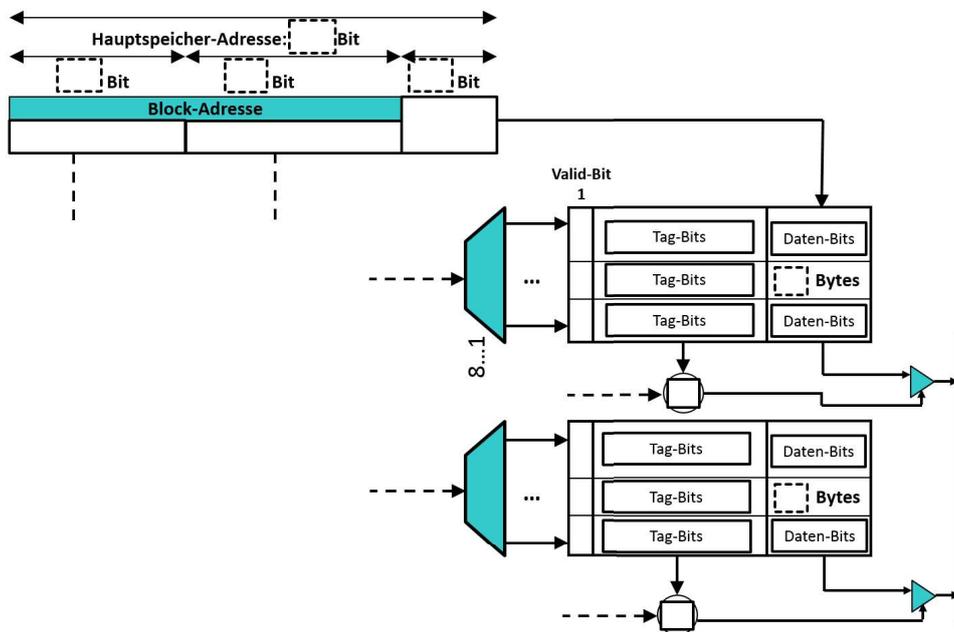
### Aufgabe 1.3: Cacheorganisation

Angenommen, ein Prozessor hat einen Hauptspeicher der Größe 1 GByte, wobei jedem einzelnen Byte im Speicher eine 64 Bit lange Adresse zugeordnet sei. Zusätzlich verfügt der Prozessor über einen Cache Baustein in SRAM Technologie, der 2048 KByte (nur Daten) vom Hauptspeicher aufnehmen kann (unabhängig vom notwendigen Speicher zur Speicherung der weiteren Bits, wie z.B. Tag). Der Cache sei 8-fach assoziativ und habe eine Blockgröße von 256 Bit. Zusätzlich wird auf dem Cache für jeden Block ein Valid-Bit vorgesehen.

**Annahmen:**

- Setzen Sie das Cache-Modell, wie es in der Vorlesung eingeführt wurde, voraus.

A) Berechnen Sie die Bitbreite von Offset, Index und Tag für den oben beschriebenen Cache. Beschriften Sie zudem die untenstehende Grafik indem Sie alle leeren Felder mit durchgehendem Rahmen mit Text oder Symbolen und Felder mit gestricheltem Rahmen mit Werten ausfüllen. Ergänzen Sie zudem die 4 gestrichelten Pfeile.




---



---



---

B) Berechnen Sie die insgesamt notwendige Speicherkapazität für den Cache-Baustein.

**Hinweis:** Falls Sie Aufgabenteil A) nicht gelöst haben, nehmen Sie folgende Werte an: Offset = 4 Bit, Index = 14 Bit





C) Erklären Sie die Begriffe statischer und dynamischer Test. Erläutern Sie zusätzlich, wieso statische Tests alleine meist nicht ausreichen.

---

---

---

---

---

---

---

D) Erklären Sie beides White-Box-Tests und Black-Box-Tests, sowie den entscheidenden Unterschied zwischen Beiden.

---

---

---

---

E) Erklären Sie den Begriff Unit-Test sowie den Unterschied zu einem Subsystemtest.

---

---

F) Nennen und erklären Sie 3 wesentliche Merkmale guter Software die in der Vorlesung behandelt wurden.

---

---

---

---

---

---

## Aufgabe 3: Programmiersprachen

A) Nennen Sie zu den folgenden Programmierparadigmen je eine Programmiersprache.

Imperativ:

Prozedural:

Objektorientiert:

B) Stellen Sie das grundlegende Unterscheidungsmerkmal von C und C++ dar und nennen Sie zwei wichtige Programmierelemente bzw. -konstrukte von C++, die nicht in C existieren. Lassen sich C++ Programme generell mit einem C Compiler übersetzen? Begründen Sie, ob dies zutrifft, es nicht zutrifft, beziehungsweise unter welchen Bedingungen dies gilt.

---

---

---

---

---

---

---

---

C) Innerhalb eines C-Programms findet unten stehende Berechnung statt. Wie lautet das Ergebnis der Berechnung? Begründen sie Ihre Antwort.

---

---

---

---

---

---

---

---

```
1   int a= 120; int b= 9; int c= 14; int x= 7;  
2   x = a / b & c;
```

D) Ein Algorithmus soll alle Primzahlen kleiner als eine angegebene Schranke bestimmen. Dabei sollen die Primzahlen auf der Konsole ausgegeben werden. Schreiben Sie ein Programm, das diesen Algorithmus für die obere Grenze  $C = 100$  umsetzt.

Tipp 1: Der Algorithmus lässt sich ohne Multiplikationen/Divisionen umsetzen.

Tipp 2: Es ist nützlich, sich die bereits bearbeiteten Zahlen mittels eines Hilfsarrays innerhalb des Algorithmus zu merken. Vergessen Sie nicht die Initialisierung des Arrays!



## Aufgabe 4: Objektorientierung in C++

### Aufgabe 4.1: Allgemeine Fragen

A) Was versteht man unter einer Klasse im Kontext der objektorientierten Programmierung? In welcher Relation stehen Objekte und Methoden zu den Klassen?

---

---

---

---

---

---

---

---

---

---

B) Erläutern Sie das Konzept der Vererbung in der objektorientierten Programmierung?

---

---

---

---

---

---

---

---

C) Wie kann man durch den Einsatz von Vererbung die Wiederverwendbarkeit von Programmteilen verbessern?

---

---

---

---

---

---

---

---

## Aufgabe 4.2: Programmanalyse

Gegeben sei der folgende C++ Code:

```
1 // File: task_4_2.h
2
3 #include <iostream>
4 using namespace std;
5
6 class A {
7 private:
8     int value;
9 protected:
10     void set_value(int v) { value = v; }
11 public:
12     A() {
13         value = 0;
14         cout << "Instantiated A from subclass ";
15     }
16     A(int i) {
17         value = i;
18         cout << "Instantiated A with " << i << endl;
19     }
20     ~A() { cout << "~A" << endl; }
21
22     int get_value() { return value; }
23     virtual void copy_from(A &a) {
24         set_value(a.get_value());
25         cout << "A::copy_from" << endl;
26     }
27 };
28
29 class B : public A {
30 public:
31     B() {
32         cout << "B" << endl;
33         set_value(10);
34     }
35     ~B() {
36         cout << "~B" << endl;
37     }
38     int get_value() { return 0; }
39     bool is_of_class_B() { return true; }
40 };
41
42 class C : public A {
43 public:
44     C() {
45         cout << "C" << endl;
46     }
47     void copy_from(A &a) {
48         A::copy_from(a);
49         cout << "C::copy_from" << endl;
50     }
51 };
```

A) Wozu dient das Schlüsselwort `virtual` in C++? Erläutern Sie die Bedeutung am Beispiel der Methode `copy_from` in den Klassendefinitionen von `A` und `C`.




---



---



---



---



---



---

B) In der nachfolgenden Tabelle sind einige aufeinanderfolgende Code-Zeilen aus der `main` Funktion eines C++ Programms aufgeführt. Es werden die oben definierten Klassen verwendet. Geben Sie für jede Zeile der Tabelle durch entsprechendes ankreuzen an, ob der jeweilige Ausdruck erlaubt ist oder ob der Compiler dort einen Fehler ausgeben würde. Wenn Sie eine Zeile als fehlerhaft kennzeichnen, geben Sie außerdem kurz den Grund dafür an.



Code-Zeile	Erlaubt?		Begründung (falls nicht erlaubt)
	Ja	Nein	
<code>A a1(2);</code>			
<code>a1.get_value() = 3;</code>			
<code>cout &lt;&lt; a1.value &lt;&lt; endl;</code>			
<code>A* b1 = new B();</code>			
<code>C* c1 = new A();</code>			
<code>b1-&gt;is_of_class_B();</code>			
<code>C* c2 = new C();</code> <code>c2-&gt;copy_from(*b1);</code>			
<code>A a2 = new A();</code>			



## Aufgabe 5: Datenstrukturen

### Aufgabe 5.1: Verständnisfragen

A) Erklären Sie was Iteratoren sind und welchen Vorteil sie beim Umgang mit Datenstrukturen bieten.

---

---

---

---

---

B) Erklären Sie den wesentlichen Unterschied zwischen den STL Containern **list**<> und **vector**<> und nennen Sie für beide Container jeweils einen Vor- und einen Nachteil.

---

---

---

---

---

---

---

---

---

---

## Aufgabe 5.2: Implementierung

```
1  class Database {
2  private:
3      std::map<std::string, MyDataType*> elements;
4
5  public:
6      // Gibt true zurueck, falls eine Element in der Datenbank ist, siehe
7      // Aufgabenteil A)
8      bool exists(const std::string& name) const;
9
10     // Gibt einen Zeiger auf ein Element zurueck oder NULL, falls Element nicht
11     // vorhanden, siehe Aufgabenteil B)
12     MyDataType* get(const std::string& name);
13
14     // fuegt ein Element hinzu, siehe Aufgabenteil C)
15     void add(const std::string& name, MyDataType& data);
16
17     // Entfernt ein Element aus der DB falls vorhanden und gibt es zurueck,
18     // ansonsten wird ein neues Objekt zurueckgegeben, siehe Aufgabenteil D)
19     MyDataType remove(const std::string& name);
20
21     // Gibt einen Iterator auf das gesuchte Element zurueck, s. Aufgabenteil E)
22     std::map<std::string, MyDataType*>::iterator
23     get_it(const std::string& name);
24 };
```

Die oben dargestellte Klasse soll eine Datenbank implementieren, die Daten des Typs *MyDataType* speichert, die jeweils einem eindeutigen Namen zugeordnet sind.

A) Implementieren Sie die Methode **exists()**:



B) Implementieren Sie die Methode **get()**:

C) Implementieren Sie die Methode **add()**:

D) Implementieren Sie die Methode **remove()**:

E) Implementieren Sie die Methode `get_it()`:

F) Was ist der Vorteil der gewählten Datenstruktur `map<>` gegenüber einer Implementierung mit `vector<>` oder `list<>`? Begründen Sie Ihre Antwort!

---

---

---

---

---

---

---

---

# Aufgabe 6: Algorithmenanalyse

## Aufgabe 6.1: Sortieralgorithmen

A) Erklären Sie das Prinzip "Teile und Herrsche" (engl. "Divide and Conquer"), wie es häufiger in Algorithmen zum Einsatz kommt. Erwähnen sie in ihrer Antwort insbesondere auch wie der Programmfluss von solchen Algorithmen üblicherweise aussieht und nennen Sie ein ihnen bekanntes Beispiel aus der Vorlesung.

---



---



---



---



---



---



---



---



---



---

B) Im Folgenden ist der aus der Vorlesung und Übung bekannte Insertion-Sort Algorithmus gegeben. Führen Sie eine Laufzeitanalyse für diesen Algorithmus durch und füllen Sie die Tabelle aus, indem Sie die Anzahl der Zeilenaufrufe im best-case und im worst-case angeben. Gehen Sie für Ihre Analyse dabei davon aus, dass das Array A die Länge "n" hat.

$$\text{Hinweis: } \sum_{j=1}^n (j) = \frac{n(n+1)}{2} \quad (6.1)$$

INSERTIONSORT (A)

1 for j = 2 to length(A)

2 do key = A[j]

3 i = j - 1

4 while i > 0 and A[i] > key

5 do A[i + 1] = A[i]

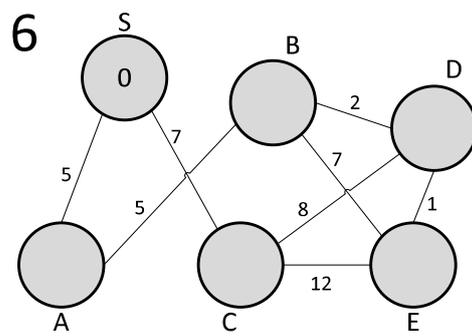
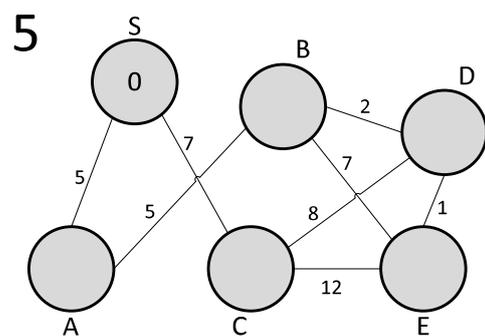
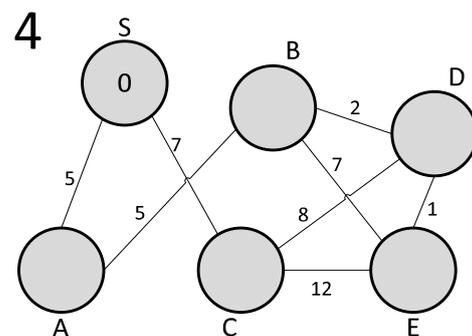
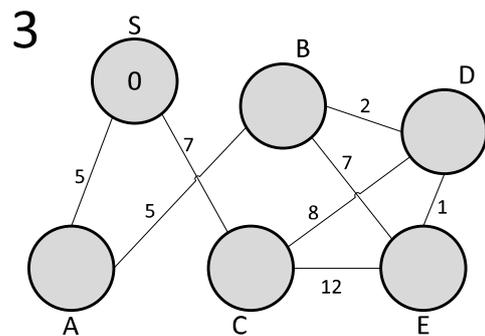
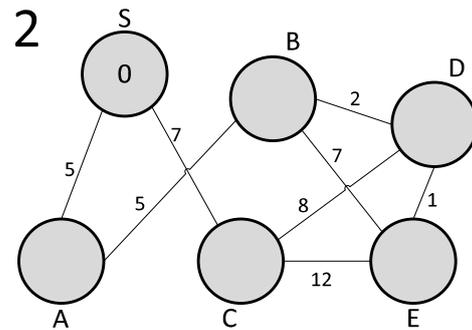
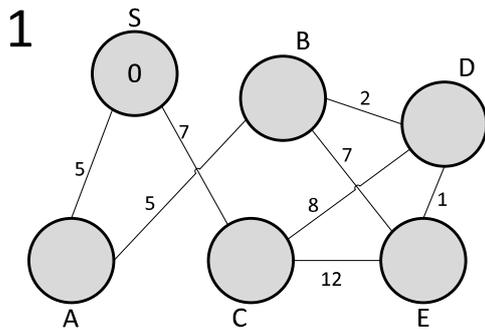
6 i = i - 1

7 A[i + 1] = key

Zeile	Zeit (worst-case)	Zeit (best-case)
1		
2		
3		
4		
5		
6		
7		

### Aufgabe 6.2: Dijkstra Algorithmus

A) Führen Sie den Dijkstra-Algorithmus auf dem folgenden Graphen aus. Nutzen Sie dabei die Knoten des vorliegenden Graphen, um die Kosten für deren Erreichen einzutragen, wie sie dem Algorithmus entsprechend in den einzelnen Schritten ermittelt werden. Geben Sie zusätzlich die Reihenfolge der permanent markierten Knoten an. Verwenden Sie "S" als Startpunkt und markieren sie abschließend die kürzeste Route zwischen dem Startpunkt "S" und dem Zielpunkt "E". Beachten Sie, dass in jedem Schritt ein Knoten markiert wird, d.h. in Schritt 1 wird der Startknoten markiert.



Reihenfolge der markierten Knoten:

Kürzester Pfad von S nach E:

# Aufgabe 7: Hardwarenahe Programmierung

## Aufgabe 7.1: Allgemeine Fragen

A) Mit dem Typqualifikator `volatile` wird in C und C++ spezifiziert, dass bei jedem Variablenzugriff der Wert direkt aus dem Speicher gelesen bzw. in den Speicher geschrieben wird. Wieso ist dies bei der hardwarenahen Programmierung besonders wichtig? Erklären Sie anhand eines DIO (Digital Input/Output).

*Hinweis:* Denken Sie an die memory-mapped Peripherieregister.

---

---

---

---

---

---

---

---

B) Sie haben zwei prinzipielle Möglichkeiten, um in einem Programm auf Veränderungen in der Peripherie zu reagieren (z.B. ADC Wert steht zur Abholung bereit): Interrupt und Polling. Was ist ein Interrupt? Was versteht man unter Polling?

---

---

---

---

---

---

---

---

C) Nennen Sie einen Vorteil und einen Nachteil von Interrupts gegenüber Polling.

*Hinweis:* Subjektive Punkte, wie z.B. "einfacher zu programmieren", werden nicht akzeptiert.

---

---

## Aufgabe 7.2: Register Programmierung

Gegeben sind folgende Informationen zum Analog-Digital-Wandler (ADC) eines Mikrocontrollers. Alle Register sind 32-Bit breit.

Basisadresse des ADC: 0xFFFF3C00

Offset	Register	Name
0x1C	Status-Register	SR
0x20	Last-Converted-Data-Register	LCDR

Tabelle 7.1: Offsets der ADC Register

- **SR, Bit 16:** DataReady (DRDY); gibt an, ob eine Wandlung erfolgreich abgeschlossen wurde und die Daten im LCDR zur Verfügung stehen. DRDY wird nach dem Lesen des LCDR automatisch gelöscht.  
0: Keine erfolgreiche Wandlung seit dem letzten Lesen des LCDR  
1: Erfolgreiche Wandlung seit dem letzten Lesen des LCDR; neue Daten im LCDR verfügbar.
- **LCDR:** Enthält den gewandelten Wert.

A) Schreiben Sie eine C++-Funktion `unsigned int readAdcValue()`, die den 32-Bit Wert des ADC ausliest und zurückgibt.

Die Funktion soll den ADC Wert nur zurückgeben, wenn seit dem letzten Lesen des Werts eine ADC Wandlung erfolgreich abgeschlossen wurde. Die Funktion soll 0xFFFFFFFF zurückgeben, wenn kein neuer Wert zur Verfügung steht (d.h. seit dem letzten Lesen wurde noch keine Wandlung erfolgreich abgeschlossen).



# Aufgabe 8: Projektmanagement

## Aufgabe 8.1: Verständnisfragen

A) Ordnen Sie die pragmatische Vorgehensweise für die Durchführung kleinerer SW-Projekte dem Ablauf nach. Beginnen Sie bei "1" für den ersten Schritt.

	Verschaffe dir Klarheit über die Aufgabenstellung
	Vervollständige die Dokumentation des entworfenen Programmsystems
	Finde den grundsätzlichen Aufbau der Eingabeobjekte bzw. des Eingabestroms
	Entwickle eine Lösungsidee für die jeweilige (Teil-)Aufgabe
	Transformiere die Lösungsidee in eine algorithmische Form
	Überprüfe die Korrektheit der algorithmischen Form der Lösung
	Fälle eine Entscheidung über die Entwurfsstrategie
	Transformiere den Algorithmus in ein Programm und überprüfe dessen syntaktische Korrektheit
	Teste das den Algorithmus repräsentierende Programm
	Beurteile die Qualität des Algorithmus und überarbeite ihn gegebenenfalls
	Definiere die (Algorithmen-)Schnittstelle(n)

B) Zu Beginn eines Projekts erstellt der Auftraggeber ein Dokument in dem er alle Anforderungen aus seiner Sicht beschreibt. Dieses Dokument wird dem Auftragnehmer übergeben, der ein zweites Dokument erstellt, in dem er die Anforderungen des Auftraggebers verfeinert und beschreibt, wie er plant diese umzusetzen.

Wie heißt das Dokument, das vom Auftraggeber erstellt wird?

Wie heißt das Dokument, das vom Auftragnehmer erstellt wird?

*Hinweis:* Aus der Lösung muss eindeutig hervorgehen, wer welches Dokument erstellt!

---



---

## Aufgabe 8.2: Projektplanung

Gegeben ist folgender Projektstrukturplan:

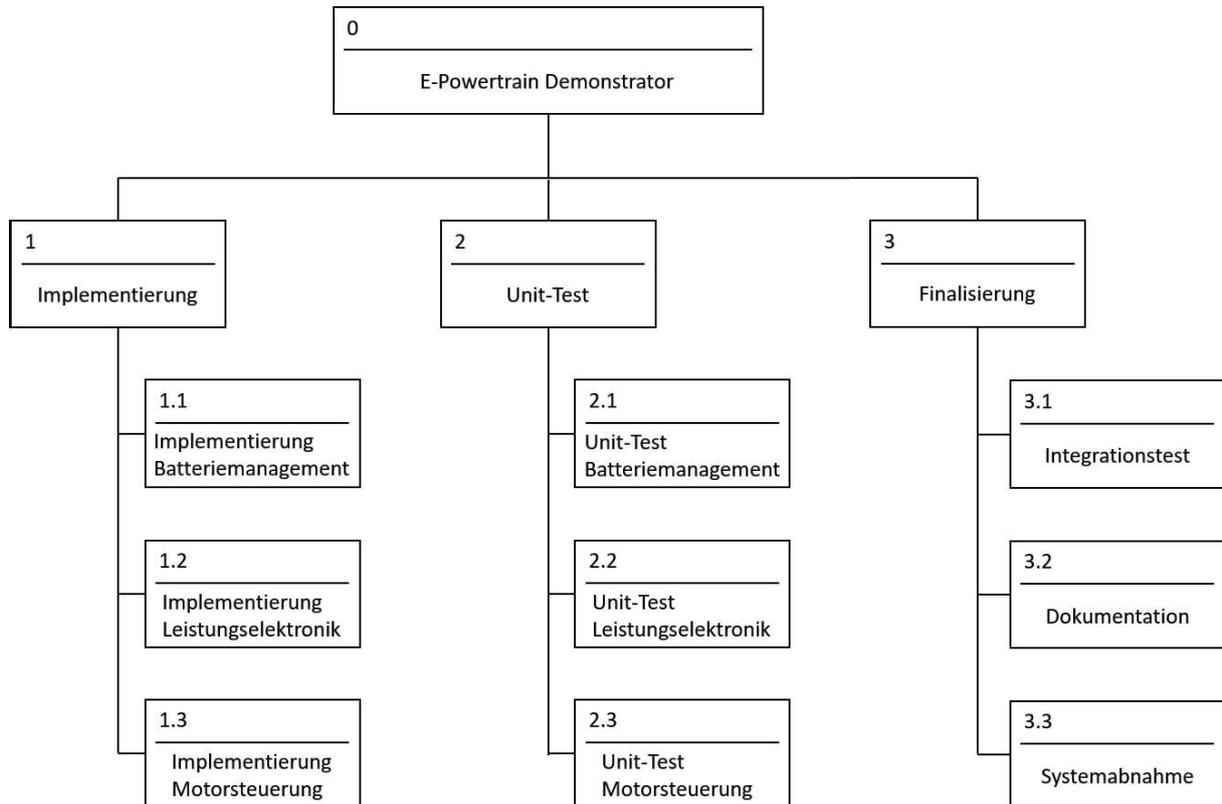


Abbildung 8.1: Projektstrukturplan

A) Der gegebene Projektstrukturplan (PSP) besteht aus drei Ebenen. Kennzeichnen Sie die drei Ebenen direkt oben in der Grafik und benennen Sie diese.

B) Aufbauend auf dem Projektstrukturplan wird der Projektablaufplan (PAP) erstellt. Welche zusätzliche Information ist im PAP, verglichen mit dem PSP, hauptsächlich enthalten?

---



---

C) Für die zeitliche Planung stehen folgende Zusatzinformationen zur Verfügung:

- Das Projektteam umfasst drei Mitglieder, die jeweils alle notwendigen Fähigkeiten mitbringen
- Jedes Arbeitspaket wird von genau einem Projektteam-Mitglied bearbeitet und ist nicht weiter aufteilbar
- Jedes Arbeitspaket soll so früh wie möglich bearbeitet werden
- Die Implementierung der Leistungselektronik und der Motorsteuerung dauert jeweils drei Wochen
- Die Implementierung des Batteriemangements dauert vier Wochen
- Jedes Arbeitspaket des Teilprojekts "Unit-Test" dauert zwei Wochen
- Implementierung und Unit-Test eines Moduls finden sequentiell statt
- Der Integrationstest dauert zwei Wochen und es müssen vor Beginn des Integrationstests alle Unit-Tests abgeschlossen sein
- Die Dokumentation dauert vier Wochen und es müssen vor Beginn der Dokumentation alle Unit-Tests abgeschlossen sein
- Die Systemabnahme dauert eine Woche und es müssen dafür alle Arbeitsergebnisse vorliegen

Vervollständigen Sie den unten stehenden Projektablaufplan auf Basis des gegebenen Projektstrukturplans und der Zusatzinformationen.

Arbeitspakete	Wochen														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Implementierung Batteriemangement															
Implementierung Leistungselektronik															
Implementierung Motorsteuerung															
Unit-Test Batteriemangement															
Unit-Test Leistungselektronik															
Unit-Test Motorsteuerung															
Integrationstest															
Dokumentation															
Systemabnahme															

Abbildung 8.2: Projektablaufplan

D) Markieren Sie im Projektablaufplan den kritischen Pfad für dieses Projekt.

## Zusätzliches Lösungsblatt: