

Inhaltsverzeichnis

1. Betriebssysteme / Software.....	3
2. Datenbanken.....	6
3. Programmiersprachen.....	13
4. Softwareentwicklung.....	25
5. Web- und Netzwerkprogrammierung / Datensicherheit.....	37
6. Referenzlisten.....	44

VORBEREITUNG - Aufgabenstellung

Name: _____

Zulässige Hilfsmittel:

Referenzlisten: SQL , Java

1. Betriebssysteme / Software

Frage 1 OS: Begriffe

1. **Aufgabe:** Geben Sie eine Definition/Erklärung/Beispiel der folgenden Begriffe:

1. Ereignis- und Sperrsynchrisation
2. Semaphore

Antwort(en):

Ereignis- und Sperrsynchrisation:

Semaphore:

Frage 2 OS: Semaphore

Semaphore-Variablen werden u.a. für den Eintritt/Austritt in/aus einem kritischen Abschnitt verwendet.

Sie bestehen aus einem Zähler und einer Warteschlange:

```
struct sema{  
    int sema_count;  
    queue sema_queue;  
};
```

Aufgabe 1: Zeigen Sie den Pseudo-code, um in den kritischen Abschnitt zu gelangen.

Aufgabe 2: Zeigen Sie den Pseudo-code, um den kritischen Abschnitt zu verlassen.

Antwort 1:

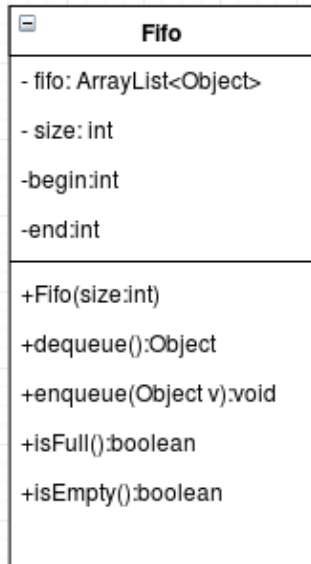
Passieren/Wait:

Antwort 2:

Verlassen/Notify:

Frage 3 OS: FIFO threadsafe

Aufgabe 1: Gegeben ist der Lückentext unten und das UML-Diagramm. Ersetzen Sie die Lücken, sodass der FIFO-Speicher threadsafe wird.



Beschreibung:

threadsafe Fifo als Ringbuffer-Speicher

Fifo() erzeugt das ArrayList-Objekt und initialisiert die Member-Variablen

dequeue() liefert das oberste Object-Element vom Fifo

enqueue() legt ein Object-Element in den Fifo

isFull() liefert true, wenn der Fifo voll ist. Sonst false.

isEmpty() liefert true, wenn der Fifo leer ist. Sonst false.

Der Fifo muss threadsafe implementiert werden. D.h. die notwendige Sperr- und Ereignis-Synchronisation ist zu implementieren.

Antwort(en):

```

class FIFO {
    private ArrayList<Object> queue;
    private int size, begin, end;

    public FIFO(int size){
        this.size = size+1;
        this.queue = new _____;
        begin = 0; end = 0;
    }

    _____ public _____ dequeue(){

        while (isEmpty())

            try {_____}catch(InterruptedException _ex){};

        Object value = queue.remove(_____);
        end++;
        end%=size;

        _____;

        return value;
    }

    _____ public boolean isEmpty(){
        if(begin == end){
            return _____;
        }
    }
  
```

```
        return _____;
    }

    _____ public boolean isFull(){
        if(end == (begin+1)%size){
            return _____;
        }
        return _____;
    }

    _____ public void enqueue(int item){

        while(this.isFull())

            try {_____}catch(InterruptedException _ex){};

        queue.add(begin,item);
        begin++;
        begin%=size;

        _____
    }
}
```

2. Datenbanken

Frage 4 DB: Check

1. **Aufgabe:** Wahr oder falsch?

Nr	Frage	WAHR	FALSCH
1	Zwischen 2 Entitäts-Typen können mehrere Beziehungen bestehen	O	O
2	Referentielle Integrität wird bei SQL mit der CHECK-Klausel realisiert	O	O
3	Ein Superschlüssel ist minimal	O	O
4	Die Attribute eines Fremdschlüssels dürfen unter keinerlei Umständen den Wert NULL annehmen	O	O
5	Inkonsistenz bedeutet, dass Informationen doppelt gespeichert werden	O	O
6	Mit der DEFAULT-Klausel wird in SQL dynamische Integrität garantiert.	O	O
7	Ein 'unique foreign key' modelliert eine 1:1 Beziehung	O	O
8	Erst die 2. Normalform führt den Begriff Schlüssel ein.	O	O

Frage 5 DB: Entity-Relationship-Diagramm

1. **Aufgabe:** Erstellen Sie aus den folgenden Angaben ein ER-Diagramm

Entity-Typ	Attribute
Angestellter	Nummer, Name, Gehalt
Abteilung	Nummer, Name
Projekt	Nummer, Beschreibung

Relationship-Typ
leiten (2x)
gehören zu
arbeiten an
sind Teilprojekte von
vertreten

Geben Sie für die folg. Beziehungen den Komplexitätsgrad an:

1. Jeder Angestellte gehört zu genau einer Abteilung
2. Jede Abteilung wird von genau einem Angestellten geleitet.
3. Zu einer Abteilung gehört mindestens ein Angestellter.
4. Ein Angestellter leitet höchstens eine Abteilung.
5. Jeder Angestellte arbeitet an mehreren Projekten
6. An einem Projekt arbeiten mehrere Angestellte.
7. Ein Projekt ist ein Teilprojekt von einem oder keinem anderen Projekt.
8. Ein Projekt kann beliebig viele Teilprojekte haben.
9. Ein Angestellter leitet mehrere Projekte
10. Ein Projekt wird von genau einem Angestellten geleitet.
11. Jeder Angestellte hat genau einen Stellvertreter.
12. Jeder Angestellte ist Stellvertreter von genau einem anderen Angestellten.

Antwort(en):

Frage 6 DB: SQL I

Gegeben:

Assistenten: {Persnr, Name, Chef}

Professoren: {Persnr, Name}

Es gilt:

1. Für jeden Professor arbeiten mehrere Assistenten
2. Jeder Assistent arbeitet für genau einen Professor.
3. Es kann nicht sein, dass ein Assistent für keinen Professor arbeitet.

1. Aufgabe:

Erstellen Sie die notwendigen SQL-Anweisungen:

1. Zeigen Sie die create table Anweisungen zum Erzeugen der notwendigen Tabellen.
2. Zeigen Sie (in SQL) die Festlegung der Primär- und Fremdschlüssel.
3. Wenn ein Professor gekündigt(gelöscht) wird, sollen auch alle seine Assistenten gelöscht werden.
4. Es darf kein Assistenten gespeichert werden, der noch keinen Chef hat.

2. Aufgabe:

Zeige die select-Anweisung,

1. um alle Assistenten aufzulisten (Assistenten.Persnr, Assistenten.Name und Professoren.Name), die für den Professor 'Sokrates' arbeiten.

Antwort(en):**1. Aufgabe:****2. Aufgabe:**

Frage 7 DB: is_uni

Gegeben seien die folgenden Relationen einer Universitätsdatenbank:

Studenten: [**MatrNr**, Name, Semester]

Professoren: [**PersNr**, Name]

Vorlesungen: [**VorlNr**, Titel]

hoeren: [**MatrNr**, **VorlNr**]

1. Aufgabe: group by

Welche Studenten (matrnr, name, anzahl_der_VL) besuchen wieviele Vorlesungen? Absteigend sortiert nach anzahlVL.

2. Aufgabe:

Schüler haben bei bestimmten Vorlesungen Fehlstunden.

1.) Zeige dies in einem ER-Diagramm.

2.) Erstelle die View Fehlstunden mit folgenden Inhalten: {Schülername, SummeFehlstunden}. Es sollen also pro Schüler die gesamten Fehlstunden angezeigt werden.

3. Aufgabe:

Ein Schüler ist pro Vorlesung ein sogenannter Sprecher.

1.) Zeige dies in einem ER-Diagramm.

2.) Liste alle Vorlesungen, bei denen der Schüler "Max Mustermann" Sprecher ist.

Antwort(en):**1. Aufgabe:****2. Aufgabe:**

1.)

2. Aufgabe:

2.)

3. Aufgabe:

1)

3. Aufgabe:

2)

Frage 8 DB: Normalformen I

Gegeben sind folg. Funktionale Abhängigkeiten

a,b	=>	c,d	e,g,h	=>	f,j
a,c	=>	b,d	p,q	=>	r,s
e,f,g	=>	h,i	s	=>	t
f,g	=>	j	q	=>	u
g,h	=>	i			

1. Aufgabe:

Welche/s der folgenden Schematas könnte das Ergebnis der Normalisierung von $R(\underline{s},q,t,u)$ sein ?

1. Das Schema $R1(\underline{s},q) R2(\underline{s},t) R3(\underline{q},u)$
2. Das Schema $R1(\underline{s},q) R2(q,t) R3(t,u)$
3. Das Schema $R1(\underline{s},q) R2(\underline{s},t) R3(q,u)$
4. Das Schema $R1(\underline{s},q,t) R2(\underline{s},q,u)$
5. Das Schema $R(\underline{s},q,t,u)$

Antwort(en):

Frage 9 DB: Normalformen II

Gegeben sei das Relationenschema

$R = \{A, B, C, D, E, F\}$
mit den funktionalen Abhängigkeiten
 $BC \rightarrow C$
 $C \rightarrow AD$
 $D \rightarrow CE$
 $E \rightarrow BC$
 $F \rightarrow D$

F ist der Schlüssel und alle Attribute A bis F sind einelementig

Aufgabe 1:

R ist in der 1. Normalform. Ist R auch in der 2. Normalform? Begründen Sie ihre Antwort.

Aufgabe 2:

Warum ist R nicht in der 3. Normalform?

Antwort(en):

Aufgabe 1:**Aufgabe 2:**

Frage 10 DB: Normalform PKW

Für eine Autovermietung wurde ein ER-Diagramm erstellt und von einem Mitarbeiter wurde dieses dann in die folgenden Tabellen übertragen.

1. Aufgabe:

Prüfen Sie die Qualität dieser Vorlage und gehen Sie dabei auf die Begriffe: Normalformen, Anomalien, Redundanzfreiheit, Funktionale Abhängigkeit, Minimalität des Schlüssels ein.

Kunden

Knr	Name	Straße	PLZ	Ort	Geb_Datum
100	Mustermann	Waldweg 9	1234	Wien	12.12.1970
101	Musterfrau	Wiesenweg 9	4321	Linz	10.11.1997

PKW

Kenn-zeichen	Erst-zulassung	Klima-anlage	Modell-name	Leistung	Länge	Hersteller-name	Hersteller-adresse
W-123 AB	10.09.2002	Ja	Opel-Clio	78	390	Opel	4444, Opel-Str. 7
L-123 BB	11.06.2005	Nein	Opel-Clio	88	490	Opel	4444, Opel-Str. 7

Ausleihe

Kenn-zeichen	Knr	Ausleihtag	Modell-name	Start_km	Rueckgabe-tag	Ende_km
W-123- AB	100	18.03.2010	Opel-Clio	95000	20.03.2010	96500
L-123 BB	101	12.02.2010	Opel-Clio	44500	12.03.2010	55000

Antwort(en):
Aufgabe 1:

3. Programmiersprachen

Frage 11 PROG-C: Pointer und Array

1. ZEIGER u. Arrays: Gegeben sei:

```
int a[5]; int *pa; int x;
a[0] liege an der Speicherstelle 1000
a[1] liege an der Speicherstelle 1002
a[2] liege an der Speicherstelle 1004
...
```

1.1.Aufgabe: Wie sieht das Speicherbild nach den folgenden Anweisungen aus ?

Tragen Sie die Werte in das Speicherbild1 unten ein.

```
pa= &a[0];
for (i=0; i < 5; i++) {
    *pa= i+100;
    pa++;
}
```

1000	1002	1004	1006	1008	1010	1012	(= Adressen dez.)
a[0]	a[1]	a[2]	a[3]	a[4]	pa	x	(= Variablennamen)
-----	-----	-----	-----	-----	-----	-----	
							(= Inhalt)
-----	-----	-----	-----	-----	-----	-----	

Abb: Speicherbild1

1.2.Aufgabe: Gegeben sind die Werte aus dem obigen Speicherbild1.

Wie sieht das Speicherbild dann nach den folgenden Anweisungen aus ?

Tragen Sie die Werte in das Speicherbild2 unten ein.

```
pa= a;
x=*pa;
a[0]= *(pa+1);
pa++;
a[1]= *pa+1;
```

1000	1002	1004	1006	1008	1010	1012	(= Adressen dez.)
a[0]	a[1]	a[2]	a[3]	a[4]	pa	x	(= Variablennamen)
-----	-----	-----	-----	-----	-----	-----	
							(= Inhalt)
-----	-----	-----	-----	-----	-----	-----	

Abb: Speicherbild2

Antwort(en):

Bitte oben einsetzen

Frage 12 PROG-C: Speicherklassen

Aufgabe: Was gibt das folgende Programm aus ?

1.Datei: <pre>int i=1; int main(){ int i,j; i=fu(); for (j=1;j<2;j++) { printf("%i%i", i,j); printf("%i", first()); printf("%i", next()); printf("%i", last(i+j)); } }</pre>	2.Datei: <pre>static int i=10; int first(){ i+=1; return i; } int next(){ i-=1; return i; } int last(int i){ static int j=5; j++; return i+j ; }</pre>	3.Datei: <pre>extern int i; int fu(){ return i; }</pre>
---	---	--

Antwort(en):

Frage 13 PROG-C: maxi

Gegeben sei folgender Funktionsprototyp :

```
?????? find_maxi(int *arr, int anzahl, ???? maxi);
```

- Der Funktion wird ein Array und dessen Anzahl der Elemente übergeben.
- Im Parameter *maxi* soll von der Funktion find_maxi() das Maximum an die aufrufende Funktion übergeben werden.(=Call by Referenz)
- Die Funktion soll auch die Adresse des Arrayelementes, das das Maximum enthält mittles return zurückgeben.

Aufgabe: Programmieren Sie find_maxi() und ersetzen Sie die Fragezeichen

```
_____ find_maxi(int* arr, int anzahl, _____ maxi){

}
}
```

Frage 14 PROG-C: Ausgabe

1. **Aufgabe:** Was gibt das folgende Programm aus? (Gib die ersten 5 Ausgabenwerte an)

```
int main(){
    int f[500]; //Felddeklaration
    int i,j;

    for (i=1;i<500;i++)
        f[i]=i; //Feld füllen

    j=2;
    while(j<250){
        for(i=2*j;i<500;i=i+j)
            f[i]=0;
        j++;
        while(f[j]==0)
            j++;
    }

    for(i=2;i<500;i++)
        if(f[i])
            printf("%i,",i);

    printf("\n");
    return 0;
}
```

Antwort(en):

Frage 15 PROG-C: Stack

Gegeben ist der folg. Programmcode für einen Stack:

Datei: doubleStack.h

```
/**
 * @brief to init the internal values of Module Stack
 *
 * sets stackpointer to 0 and uses a static double Array with 1024 Elements
 *
 * @param void
 * @return void
 */
void init(void);

/**
 * @brief puts a value on top of stack
 *
 * if stack is already full an error message is written to stderr
 *
 * @param value The value to be pushed on top of stack
 * @return void
 */
void push(double value);

/**
```

```
* @brief return the element on top of stack
*
* if stack is empty an error message is written to stderr
*
* @param void
* @return the element on top of stack
*/
double  top(void);

/**
* @brief delete the element on top of stack
*
* if stack is already empty an error message is written to stderr
*
* @param void
* @return void
*/
void  pop(void);
```

Aufgabe:

Erstellen Sie die Datei doubleStack.c

Antworten:

Frage 16 PROG-C++: out

1. **Aufgabe:** Ergänzen Sie das folgende Programm (beachten Sie die Hinweise) und
2. **Aufgabe:** geben Sie die genaue Ausgabe an.

```
#include <iostream>

_____ std;

// HINWEIS 1: Globale Variablen
int x = 255, y = 254;

// HINWEIS 2: Ausgabe von x und y als Hexadezimal- und dann als Dezimalzahl
// HINWEIS 3: x und y dürfen in der Funktion nicht geändert werden

void out(_____ int x, _____ int y = -1){
    cout _____ x << "    " _____ x << endl;
    if (y >= 0) { cout _____ y << "    " _____ y << endl;}
    cout << endl;
}

// HINWEIS 4: Der Funktionsaufruf wird durch den Programmcode inline ersetzt
_____ void out() { cout << x << endl << endl; }

// HINWEIS 5: a und b werden getauscht

void swap(_____ a, _____ b) { int h = a; a = b; b = h; }

int main(){
// HINWEIS 6: Lokale Variablen
    int x=127, y=128;

    cout << "Ausgabe:"<< endl;
    out(); out(x); out(::x, y); swap(x, y); out(::x, y);
    return 0;
}
```

Antwort(en):**1. Aufgabe:** Bitte oben die Lücken ausfüllen.**2. Aufgabe:****Ausgabe:**

Frage 17 PROG-C++: polygon

Gegeben ist die Klassenhierarchie:

CPolygon

CTriangle

CRectangle

1. Aufgabe: Füllen Sie die Lücken im Programm unten aus, sodass die Klasse CPolygon eine abstrakte Klasse wird und die Ausgabe des Programmes lautet:

20

10

2. Aufgabe: Beantworten Sie die folgenden Fragen:

1. Frage: Reine virtuelle Methoden müssen von der jeweiligen
o Unterklasse oder o Klasse, in der sie deklariert werden implementiert werden.

2. Frage: Der Destruktor der Basis-Klasse muss virtuell sein, weil _____

```
class CPolygon {  
    _____ :  
        int width, height;  
public:  
    void set_values (int a, int b) { width=a; height=b; }  
  
    _____ ;  
  
    void printarea (void) { cout << this->area() << endl; }  
};  
  
class CRectangle: public CPolygon {  
public:  
    int area (void) { return (width * height); }  
};  
  
class CTriangle: public CPolygon {  
public:  
    int area (void){ return (width * height / 2); }  
};  
  
int main () {  
    CPolygon * ppoly1 = new CRectangle();  
    CPolygon * ppoly2 = new CTriangle();  
  
    ppoly1->set_values (4,5);  
    ppoly2->set_values (4,5);  
    ppoly1->printarea();  
    ppoly2->printarea();  
    delete ppoly1;  
    delete ppoly2;  
    return 0;  
}
```

Antwort(en):

Oben bitte direkt ausfüllen.

Frage 18 PROG-C++: cipher

Gegeben ist die Klasse Cipher:

```
class Cipher {  
public:  
    string encrypt(string s, int key);  
    string decrypt(string s, int key);  
  
    virtual char encrypt(char ch, int key)=0;  
    virtual char decrypt(char ch, int key)=0;  
};
```

Die Klasse XORSubstitution als Unterklasse von Cipher:

Bei der XOR Substitution wird der Code jedes Zeichens durch XOR mit dem Key verküpft.

Sehen Sie hier die Verwendung der Klasse:

```
string plainText="Hallo, Welt!";  
int key= 15;  
Cipher* cipher = new XORSubstitution();  
string cryptText = cipher->encrypt(plainText, key);
```

1. **Aufgabe:** Programmieren Sie die CPP-Klassen Cipher.cpp und XORSubstitution, die von Cipher (siehe oben) abgeleitet ist, indem Sie die folgenden Dateien programmieren:
 1. Die Datei cipher.cpp
 2. Die Datei xorsubstitution.h
 3. Die Datei xorsubstitution.cpp

Antwort(en):

Die Datei cipher.cpp:

Die Datei xorsubstitution.h:

Die Datei xorsubstitution.cpp:

Frage 19 PROG-C++: mystring

Gegeben ist das folgende Programm, bei dem Sie 2 Problemfälle analysieren und lösen sollen.

Hier zunächst das Programm:

```
#include <iostream>
#include <cstring>
using namespace std;

class MyString{
private:
    char* s;
    int size;
public:
    MyString(const char* s){
        this->size= strlen(s);
        this->s= new char[size+1];
        strcpy(this->s, s);
    }
    friend ostream& operator<<(ostream& o, const MyString& e){
        o << e.s;
        return o;
    }
};

//globals
MyString sglobal1("SGLOBAL1");
MyString sglobal2("SGLOBAL2");

void dowhat(void){
    // PROBLEM1:
    MyString slocal1(sglobal1);
    cout << "slocal1= " << slocal1 << endl;

    // PROBLEM2:
    sglobal2= slocal1;
}

int main(){
    dowhat();
    cout << "sglobal1= " <<sglobal1 << endl;
    cout << "sglobal2= " <<sglobal2 << endl;
    return 0;
}
```

2. **Aufgabe 1:** Programmieren Sie den Destruktor.
3. **Aufgabe 2:** PROBLEM1
 1. Warum passiert hier ein Fehler?
 2. Lösen Sie das Problem, indem Sie die notwendige Member-Funktion programmieren.
4. **Aufgabe 3:** PROBLEM2
 1. Warum passiert hier ein Fehler?
 2. Lösen Sie das Problem, indem Sie die notwendige Member-Funktion programmieren.

Antwort(en):

Aufgabe 1:**Aufgabe 2:****Aufgabe 3:**

Frage 20 PROG-C++: Matrix

Aufgabe 1: Programmieren Sie die Klasse Matrix, sodass folgendes funktioniert:

```
#include "matrix.h"
#include <iostream>
using namespace std;

int main(){
    Matrix a(3,3); // 3x3 double Werte mit 0.0 initialisiert
    Matrix b(3,3);
    Matrix c(3,3);

    for (int i=0; i<3; i++)
        for (int j=0; j<3; j++)
            a(i,j)= 1.0;
    for (int i=0; i<3; i++)
        for (int j=0; j<3; j++)
            b(i,j)= 2.0;
    c= a + b;
    return 0;
}
```

Antwort(en):

Frage 21 PROG-C++: EAN

Programmieren in C++ - Prüfwertberechnung

Zur Kennzeichnung von Waren verwendet man den sogenannten EAN-Code (Europäische Artikel-Nummerierung mit 13 Ziffern). Bei Büchern ist die ISBN-Nummer üblich. Bei der letzten Ziffer der Nummer handelt es sich um eine sog. Prüfwert, sodass z.B. einfache Eingabefehler erkannt werden können. Die Prüfwert berechnet sich aus den übrigen 12 Ziffern.

ISBN-13

Zur Berechnung der Prüfwert bei der ISBN-13 werden alle zwölf Ziffern der noch unvollständigen ISBN addiert, wobei die Ziffern mit gerader Position (also die 2., 4. usw.) dreifachen Wert erhalten. (Beispiel: Eine 5 an 6. Stelle fließt als 15 in die Addition ein.)

Von dem Ergebnis der Addition der 12 Ziffern wird die letzte Stelle ('Einer-Stelle') bestimmt, die dann von 10 subtrahiert werden muß. (Beispiel: Wenn das Additionsergebnis 124 ist, ist die 'Einer-Stelle' also 4. Diese Zahl wird dann von 10 subtrahiert. Dieses Ergebnis ist dann wiederum die Prüfwert der ISBN-13. In diesem Beispiel also: $10 - 4 = 6$. Die Prüfwert ist 6.)

Ist das Endergebnis der Addition der ersten 12 Ziffern indessen 10, so ist die Prüfwert 0.

Formel zur Berechnung der Prüfwert: $z_{13} = 10 - \left(\sum_{i=1}^{n=12} z_i \cdot 3^{(i+1) \bmod 2} \right) \bmod 10$

Das $(i+1) \bmod 2$ sorgt für die wechselnde Gewichtung von 1 und 3.

Beispiel: Berechne die Prüfwert

978-3-7657-2781-?

Berechnung:

$9+8+7+5+2+8 + 3 * (7+3+6+7+7+1) = 39 + 3*31 = 39 + 93 = 132$

$132 \bmod 10 = 2$

$10 - 2 \bmod 10 = 8$ d.h. Die Prüfwert ist 8

Aufgabe:

Erstellen Sie das Programm pruefwert.cpp (siehe unten), das die Korrektheit der angegebenen ISBN-Nummer berechnen soll.

Ein Beispiel für die Verwendung:

```
int main() {
    string isbn= string("978-3-89771-040-5");
    string isValid;

    cout << "ISBN: " << isbn << endl;
    isValid= Pruefwert::isValid_ISBN(isbn) ? " true" : " false";
    cout << "Pruefwert ist " << isValid << endl;

    return 0;
}
```

Antwort: pruefwert.cpp (Ersetzen Sie die Fragezeichen mit dem richtigen CPP-Sourcecode.)

```
class Pruefwert{
public:
    /**
     * Gibt an, ob die Prüfwert der ISBN-Nummer gültig ist.
     * verwendet: calcPruefwert_ISBN(string nummerISBN)
     * @param string nummerISBN: Die ISBN Nummer (inkl. '-')
     * @return bool: true, wenn die errechnete Prüfwert
```

```
        *   gleich der letzten Ziffer der ISBN Nummer ist
        */
static bool isValid_ISBN(string nummerISBN){
    ???????????

}

/**
 * Berechnet aus der ISBN-13 Nummer die Prüfziffer
 * @param string nummerISBN: Die ISBN Nummer (inkl. '-')
 * @return char: errechnete Prüfziffer */
static char calcPruefziffer_ISBN(string nummerISBN){
    ???????????
    // löscht alle '-' Zeichen
    string nummer= getDigitsOnly(nummerISBN);

    // berechnet die Prüfziffer und gibt diese zurück
    ???????????

}

private:
    // lokale Hilfsfunktion
    // gibt einen string, der nur aus Ziffern besteht zurück
    static string getDigitsOnly(string nummer){
        ???????????

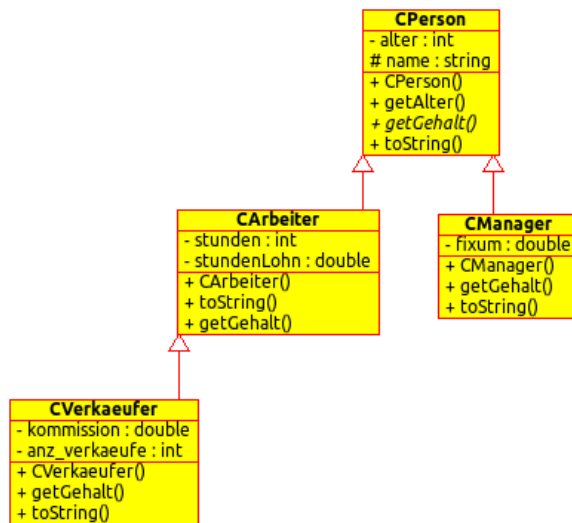
    }
};
```

Hinweis:

```
int isdigit(int);
```


4. Softwareentwicklung

Frage 22 SENG: UML-Personen



Gegeben ist ein UML Diagramm.

Aufgabe 1:

Es sollen noch Programmierer hinzugefügt werden. Programmierer werden -wie Arbeiter- entlohnt und erhalten für ihren jeweiligen LinesOfCode einen Zusatz.

- Fügen Sie die Klasse Programmierer in das UML-Diagramm.
- Erstellen Sie die Klasse Programmierer in der Sprache C++.

Aufgabe 2:

Es soll nun die Klasse Firma erstellt werden, die alle Arbeiter, Verkäufer, ... speichert und das monatliche Gehaltsaufkommen (`double` `gesamtGehalt()`) errechnen lässt.

- Fügen Sie die Klasse Firma in das UML-Diagramm.
- Erstellen Sie für die Klasse Firma in der Sprache C++ nur die Header-Datei: `firma.h`

Aufgabe 3:

Erklären Sie die Begriffe Polymorphismus und spätes Binden.

Antwort(en):

Aufgabe 1:

Aufgabe 2:**Aufgabe 3:**

Frage 23 SENG: Rekursion I

Die Fibonacci-Folge (f_0, f_1, \dots) ist durch das Bildungsgesetz

$$f_n = f_{n-1} + f_{n-2} \text{ für } n \geq 2$$

mit den Anfangswerten

$$f_0 = 0 \text{ und } f_1 = 1$$

definiert. Das bedeutet in Worten:

- a) Für die beiden ersten Zahlen werden die Werte 0 und 1 vorgegeben.
- b) Jede weitere Zahl ist die Summe ihrer beiden Vorgänger.

Daraus ergibt sich die Folge zu:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, usw.

Aufgabe 1:

Schreiben Sie ein Programm, mit dem Namen Fibonacci, mit dem die ersten n Zahlen der Fibonacci-Folge ausgegeben werden. Die einzelnen Zahlen sollen in einer Zeile ausgegeben werden. Als Trennzeichen (Delimiter) sollte ein Beistrich-Zeichen verwendet werden.

Antwort(en):

Frage 24 SENG: Rekursion II

Geben ist folgende Funktion (Setzen Sie für den ersten Aufruf $x > y > 0$ voraus)

```
long methodeZ(long x, long y) {  
    if (x >= y)  
        return (x+methodeZ(x-1,y));  
    else  
        return 0;  
}
```

Aufgabe 1: Was lässt sich mit dieser Funktion berechnen?

Aufgabe 2: Wie lässt sich das Ergebnis mit Hilfe einer Schleife ermitteln?

Aufgabe 3: Wo in der Informatik verwendet man die Methode der Rekursion?

Antwort(en):

Aufgabe 1:

Aufgabe 2:

Aufgabe 3:

Frage 25 SENG: quicksort

Aufgabe 1:

Im folg. Algorithmus sind 4 Fehler eingebaut. Korrigieren Sie diese.

Quelle: <https://de.wikipedia.org/wiki/Quicksort>

```
funktion quicksort(links, rechts)
    wiederhole solange links < rechts dann
        teiler:= teile(links, rechts)
        quicksort(links, teiler+1)
        quicksort(teiler-1, rechts)
    ende
ende

funktion teile(links, rechts)
    i:= links
    // Starte mit j links vom Pivotelement
    j:= rechts - 1
    pivot:= daten[rechts]

    wiederhole
        // Suche von links ein Element, das größer als das
        // Pivotelement ist
        wiederhole solange daten[i] <= pivot und i < rechts
            i:= i + 1
        ende

        // Suche von rechts ein Element, das kleiner als das
        // Pivotelem. ist
        wiederhole solange daten[j] <= pivot und j > links
            j:= j - 1
        ende

        falls i < j dann
            tausche daten[i] mit daten[j]
        ende
    solange i < j // solange i an j nicht vorbeigelaufen ist

    // Tausche Pivotelem.(daten[rechts]) mit endgült. Position
    // (daten[i])

    falls daten[i] > pivot dann
        tausche daten[i] mit daten[rechts]
    ende

    // gib die Position des Pivotelements zurück

    antworte i
ende
```

Antwort(en):

Frage 26 SENG: Performance Sort

Aufgabe 1:

Sortieralgorithmen spielen in der Informationsverarbeitung eine wesentliche Rolle.

1. Beschreiben Sie in diesem Zusammenhang die Arbeitsweise des **Quicksort**-Algorithmus.
2. Stellen Sie einen Vergleich mit dem Bubblesort-Algorithmus an.
3. Wenn die zu sortierenden Datenmenge bereits sortiert sein sollte, welcher Algorithmus ist dann schneller?

Antwort(en):

Frage 27 SENG:Search

Gegeben ist folgender Algorithmus:

```
int myBinSearch (int key, int *a, int left, int right) {  
    while (left <= right) {  
        int m= (left+right)/2;  
        if (key==a[m]) return m;  
        else if(key < a[m]) right= m-1;  
        else  
            ???????  
    }  
    return -1;  
}
```

Aufgabe 1: Vervollständigen Sie das obige Programm.

Aufgabe 2: Transformieren Sie diesen Algorithmus in eine **rekursive** Version.

Antwort(en):**Aufgabe 1:**

Aufgabe 2:

Frage 28 Listen I

Gegeben sind folg. Datenstrukturen:

```
typedef struct {char* data; SLIST* next;} SLIST;
typedef struct {SLIST* first; int len; SLIST* last;} SLIST HEADER;
```

Aufgabe 1: Um den Text des 2. Listenknotens auszugeben:

```
void printIt(SLIST_HEADER* aList){
    printf("Text= %s", _____)
}
```

Aufgabe 2: Gesucht ist die Funktion, die den längsten Text in der Liste ausgibt.

```
void printLongest(SLIST_HEADER* aList) {????????????????????????????????????????}
```

Antwort(en):

Aufgabe 1: oben eintragen

Aufgabe 2:

```
void printLongest(SLIST_HEADER* aList) {
```

}

Frage 29 Bäume I

Gegeben ist folg. rekursive Funktion.

Aufgabe 1: Gesucht ist die iterative Version, also kein rekursiver Aufruf mehr.

```
BinSearchTree* search (int key, BinSearchTree* root){  
    if (root == NULL)  
        return NULL;  
    else if (key == root->key)  
        return root;  
    else if (key < root->key)  
        return search (key, root->Left);  
    else  
        return search (key, root->Right);  
}
```

Antwort(en):

Aufgabe 1:

Frage 30 Bäume II

Folgende Ausgabe wurde durch postorder() in einem binären Baum erzeugt.

1 2 / 3 * 4 * 5 +

Aufgabe 1: Zeichnen Sie den Baum

Aufgabe 2: Vervollständigen Sie folg. Programm:

void postorder(BNODE* ptr){????????????????}

Antwort(en):

Aufgabe1:

Aufgabe2:

```
void postorder(BNODE* ptr){
```

```
}
```

Frage 31 Bäume III

Gegeben sei die Datenstruktur:

```
typedef struct tree {int data; struct tree* left; struct tree* right;} TREE;
```

Aufgabe 1: Ausgehend von einem beliebigen Knoten in einem binären Suchbaum, geht man 1 Knoten nach links und dann immer nach rechts bis zum Blatt. Um welchen Knoten handelt es sich bei diesem Blatt in Bezug auf den Ausgangsknoten?

Antwort: _____

Aufgabe 2: Erstellen Sie die Funktion `sum()`, um die Werte aller Knoten eines binären Suchbaumes aufzusummieren.

```
int sum(TREE* tree){  
    if (_____) {
```

```

        return _____ + tree->data + _____;
    }
}

```

Aufgabe 3: Welche Funktion gibt einen binären Suchbaum sortiert aus?

- ☐ inorder
- ☐ postorder
- ☐ preorder

Antwort(en):

oben eingeben

Frage 32 SENG: Singleton (+)

Aufgabe 1: Füllen Sie die Lücken im folg. Programm.

```

class Singleton {
    _____ :
    Singleton() {} // you cannot create an object
    Singleton(const _____ ) {} // you cannot make a copy

    // you cannot make a copy by assign-operator
    Singleton& operator=(const ??????????) { return *this; }
    ~Singleton() {}

    _____ :
    static Singleton& getInstance(){
        _____ Singleton instance;

        return _____ ;
    }

    void log(int level, string msg){
        time_t second;
        struct tm *atime;
        char sTime[80];
        time(&second);
        atime= localtime(&second);
        strftime(sTime, 80, "%c", atime);
        cout << sTime << ":" <<level << ":" << msg<<endl;
    }
};

int main() {
    // create the one and only one singleton object.
    // its created within getInstance(), that returns
    // reference to the singleton object

    Singleton logger= Singleton::getInstance();
}

```

```
// Addresses are all the same, because of there is
// only one and only one singleton object

cout << "\ndemonstration of singleton pattern: " << endl;
cout << "    3 addresses should have the same value:" << endl;
cout << "    "<< hex << &logger << endl;
cout << "    "<< hex << &Singleton::getInstance() << endl;
cout << "    "<< hex << &Singleton::getInstance() << endl;

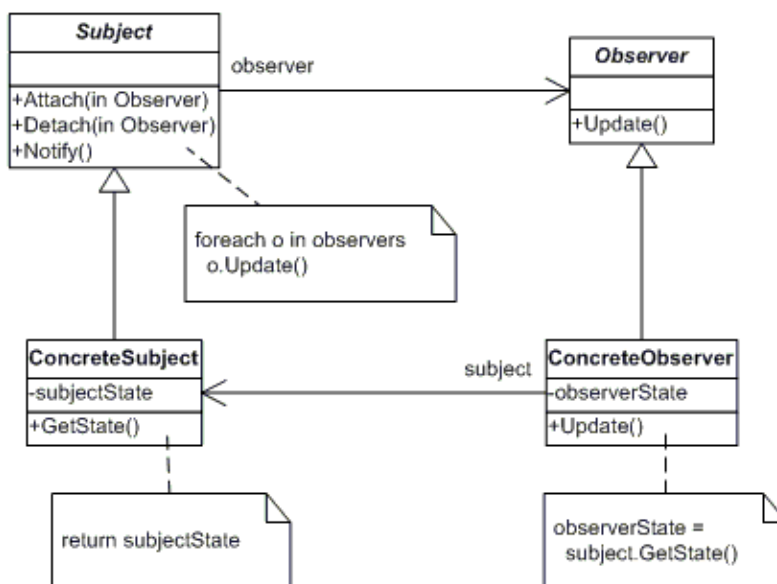
// you can use/reference the singleton object
Singleton::getInstance().log(0, "this is my first log entry");

logger.log(1, "here is my second log entry");

return 0;
}
```

Antwort(en):
Lücken bitte oben ausfüllen.

Frage 33 SENG: Observer (+)



Aufgabe 1:

Gegeben ist das oben befindliche Klassendiagramm. Gesucht sind folgende Teilaufgaben:

1. Erklären Sie das Observer-Pattern
2. Implementierung in C++ (Pseudocode)
3. Wo wird dieses Pattern gerne eingesetzt?

Antwort(en):

5. Web- und Netzwerkprogrammierung / Datensicherheit

Frage 34 NW-PROG: Socket-API I

Aufgabe 3: Welche Funktion wird wo verwendet?

socket:	<input type="radio"/> clientside	<input type="radio"/> serverside	<input type="radio"/> bothside
accept:	<input type="radio"/> clientside	<input type="radio"/> serverside	<input type="radio"/> bothside
bind:	<input type="radio"/> clientside	<input type="radio"/> serverside	<input type="radio"/> bothside
listen:	<input type="radio"/> clientside	<input type="radio"/> serverside	<input type="radio"/> bothside
send:	<input type="radio"/> clientside	<input type="radio"/> serverside	<input type="radio"/> bothside
recv:	<input type="radio"/> clientside	<input type="radio"/> serverside	<input type="radio"/> bothside
close:	<input type="radio"/> clientside	<input type="radio"/> serverside	<input type="radio"/> bothside

Frage 35 NW-PROG: Socket-API II

Aufgabe1: Beantworten Sie die hier gestellten Fragen:

Frage 1: Ordne folgende Begriffe den 7 Schichten des OSI-Modells zu

- ___3___ <- ICMP
- ___ <- Three-Way-Handshake/TCP
- ___ <- htons()
- ___ <- IP-Adresse
- ___ <- FTP
- ___ <- Port-Adresse
- ___ <- Big Endian (Byte-Reihenfolge)

Frage 2: Welche Komponenten enthält die Socket-Adress Struktur?

Frage 3: Wie heisst die Struktur zur Namensauflösung? _____

Frage 4: Welche Komponenten enthält sie?

Frage 5: Nenne Funktionen zur (DNS) Namensauflösung

Frage 6: Wie lautet die IP-Adresse von localhost? _____

Frage 7: Bringen Sie die Funktionen in die richtige Reihenfolge:
connect(),listen(),socket(), accept(),write(),read(),close()

Client:

- 1.) _____ 2.) _____
3.) _____ 4.) _____
5.) _____ 6.) _____

Server:

- 1.) _____ 2.) _____
3.) _____ 4.) _____
5.) _____ 6.) _____

Frage 8: Welche Funktion fehlt oben? _____

Frage 9: wahr oder falsch

int sd= socket(AF_INET,SOCK_STREAM,0);

- ☐ wahr
☐ falsch

int ret= read(sd, buf, sizeof(buf));

- ☐ wahr
☐ falsch

read() ist blockierend

- ☐ wahr
☐ falsch

wähle: accept() liefert den

- ☐ connection socket
☐ well-known-port socket als return-Wert

Frage 10: Sie lesen von einem Socket. Wie erkennen sie, dass der Partner die Verbindung beendet hat?

Frage 11: Geben Sie die 7 Schichten des OSI-Modelles an.

Frage 12: Die Eigenschaften von TCP:

Frage 13: RFC ist die Abkürzung für _____

Frage 14: Erklären Sie die folg. Begriffe und geben Sie je ein Beispiel
a) http-header

b) html-header

Frage 15: Das http-Protokoll ist in einem sogenannten _____-Dokument spezifiziert.

Frage 16: Das http-Protokoll ist ein zustandsloses Protokoll. Das bedeutet, dass

Frage 17: Das http-Protokoll befindet sich im ISO/OSI-Schichten Modell im Layer _____

Frage 18: Beim http-Protokoll trennt eine Leerzeile _____

Frage 19: Gegeben:

```
char buf[512], filename[512];  
strcpy(buf, "GET /index.html HTTP/1.1\r\n");
```

Gesucht: Zeigen Sie den C-Programmcode um in das Array filename den Dateinamen index.html zu bringen. (Der Dateiname index.html ist hier beliebig gewählt)

Frage 20: Gegeben:

```
typedef uint32_t in_addr_t;
struct in_addr { in_addr_t s_addr; };

in_addr_t inet_addr(const char *cp);
char *inet_ntoa(struct in_addr in);

uint32_t htonl(uint32_t hostlong);
uint16_t htons(uint16_t hostshort);
uint32_t ntohl(uint32_t netlong);
uint16_t ntohs(uint16_t netshort);

char buf[512];
strcpy (buf, "172.16.48.122");
```

Gesucht: Zeigen Sie den C-Programmcode:

Die IP-Adresse in buf soll in die Variable curr im sog. Host-Format gespeichert werden.

unsigned long curr= _____

Frage 21: Gegeben:

```
typedef uint32_t in_addr_t;
struct in_addr { in_addr_t s_addr; };

in_addr_t inet_addr(const char *cp);
char *inet_ntoa(struct in_addr in);

uint32_t htonl(uint32_t hostlong);
uint16_t htons(uint16_t hostshort);
uint32_t ntohl(uint32_t netlong);
uint16_t ntohs(uint16_t netshort);

char buf[512];
strcpy (buf, "172.16.48.122");
```

Gesucht: Zeigen Sie den C-Programmcode:

Die Variable unsigned long curr; hält eine IP-Adresse im sog. Host-Format.
Wie lautet der C-Programmcode, um diese IP-Adresse als Nummernnotation auf die Konsole auszugeben. (zB: 127.0.0.1)

Frage 22: Die struct sockaddr_in hat folg. Komponenten:

Frage 23: http verwendet den Port _____ und https verwendet den Port _____

Frage 24: Gegeben sind Auszüge aus C-Programmen, die das Socket-API nutzen. Füllen Sie die richtigen Funktionsnamen bzw. Parameternamen ein.

Auszug:

```
....  
int sd, ld;  
struct sockaddr_in name;  
char buf[MSGSIZE];  
  
ld = socket ( _____ );  
  
name.sin_family = AF_INET;  
name.sin_port = htons(0);  
name.sin_addr.s_addr= INADDR_ANY;  
  
_____ (ld, (struct sockaddr*)&name, sizeof (name));
```

Auszug:

```
_____ (ld, 5);  
  
while ( _____ ){  
    sd= accept (ld, &addr, &addrlen);  
    nbytes= read ( _____ );  
  
    _____ fp= fopen(buf, "rt");  
  
    while (fgets( _____ )){  
        write ( _____ );  
    }  
    fclose ( _____ );  
    close ( _____ );
```

Auszug:

```
name.sin_family = AF_INET;  
name.sin_port = htons(80);  
name.sin_addr.s_addr= inet_addr("172.16.48.122");  
  
_____ (sd, (struct sockaddr*)&name, sizeof(name));
```

Frage 36 NW-PROG: SSL/Zertifikate

1. **Aufgabe:** Erklären Sie die folgenden Begriffe:
 1. PRIVATE_KEY,
 2. ECHTHEIT(=AUTHENTIZITÄT),
 3. ECHTHEIT des PUB-Keys,

4. DIGITALE ZERTIFIKATE,
5. SERVER-ZERTIFIKAT
6. SELF-Signed Zertifikate
7. CA
8. CSR
9. Signieren

2. Aufgabe:

1. Ein Zertifikat beinhaltet u.a. folgendes:

1. _____
2. _____
3. _____

2. Man signiert, um den Sicherheits-Dienst
o Verschlüsselung oder **o Echtheit** zu garantieren.

☒ **Frage:** Woran erkennt man, dass es sich um ein selbst-signiertes Zertifikat handelt?

☐ **Antwort: o stimmt o stimmt nicht**

Da **Subject:** (zertifizierter Gegenstand) und **Issuer:** (Zertifikats-Aussteller) identisch sind, handelt es sich um ein **selbst-signiertes Zertifikat**.

☒ **Frage:** Woran erkennt man, dass es sich um ein CA Zertifikat handelt?

☐ **Antwort: o stimmt o stimmt nicht**

Im Bereich **x509v3 Basic Constraints:** ist anhand von **CA:TRUE** auch erkennbar, dass es sich um ein CA-Zertifikat handelt.

☒ **Frage:** Wie kann nun der Browser erkennen, ob das Server-Zertifikat (also der public-key) wirklich vom Server stammt? (=Echtheit)

☐ **Antwort:: o stimmt o stimmt nicht**

Die CA installiert ihr eigenes CA-Stamm-Zertifikat (also seinen eigenen public-key) beim Browser.

☒ **Frage:** Wie kann ein Browser feststellen, dass das **Server-Zertifikat wirklich von der jeweiligen CA** signiert wurde?

☐ **Antwort: o stimmt o stimmt nicht**

Es wird der public-key der CA verwendet. Er 'passt' zum private-key der CA, der ja für das Signieren des Server-Zertifikates verwendet wurde.

Antworten:

Aufgabe 1: Erklären Sie die folgenden Begriffe:

1. PRIVATE_KEY,

2. ECHTHEIT(=AUTHENTIZITÄT),

3. ECHTHEIT des PUB-Keys,

4. DIGITALE ZERTIFIKATE,

5. SERVER-ZERTIFIKAT

6. SELF-Signed Zertifikate

7. CA

8. CSR

9. Signieren

Die Antworten für die anderen Aufgaben tragen Sie bitte oben direkt ein.

Frage 37 NW-PROG: Diffie-Hellman

Bei der symmetrischen Verschlüsselung kann der Algorithmus von Diffie-Hellman verwendet werden, der ein sogenanntes Trustcenter verwendet. Dieses speichert N (Modulo genannt und ist prim) und G (Generator genannt). Sowohl der Sender -im Beispiel oft Alice genannt- ,als auch der Empfänger – oft Bob genannt – lesen diese Werte. Diffie-Hellman ist deswegen interessant, weil der Schlüssel für die symmetrische Verschlüsselung nicht übertragen werden muss.

Gegeben seien die folgenden Abkürzungen, die Sie in den folgenden Lückentext eintragen:

TC	→ Trustcenter	N	→ Modulo (prim)	G	→ Generator
BPK	→ Bob's Public Key	BSN	→ Bobs Secret Number		
APK	→ Alice's Public Key	ASN	→ Alice's Secret Number		
KEY	→ symm. Key f. Encryption	%	→ the modulo Operator		
M	→ the message (plainText)	C	→ the message (cryptText)		
C=encrypt (M,KEY)		M=decrypt (C,KEY)			

Aufgabe 1:

Given: use the following values to calculate APK,BPK, symKey.

Alice and Bob use $n = 13$ and $g = 2$. Alice uses $ASN = 5$. Bob uses $BSN = 8$.

Question 1: Alice:

Show the calculation of APK

APK= _____

Question 2: Alice:

Show the calculation of symKey

symKey= _____

Question 3: Bob:

Show the calculation of BPK

BPK= _____

Question 4: Bob:

Show the calculation of symKey

symKey= _____

6. Referenzlisten

```
CREATE TABLE TabellenName (Spalte1 Datentyp Constraint, ...);
ALTER TABLE TabellenName ADD CONSTRAINT ConstraintName
    FOREIGN KEY Name
    REFERENCES TabellenName(SpaltenName,...) ON ??????????????????;
CREATE INDEX IndexName ON TabellenName(Spaltenname, ...);
SELECT Spalten FROM Tabellen WHERE Join-Bedingung [AND|OR einfache Bedingung ]
    ORDER BY Spalten;
SELECT Spalte1, SUM(Spalte2) FROM Tabellen GROUP BY Spalte1 HAVING Bedingung;

DROP TABLE Tabelle;
INSERT INTO Tabelle (Spalten) VALUES (Werte);
UPDATE Tabelle SET Spalte= Wert WHERE Bedingung;
DELETE FROM Tabelle WHERE Bedingung;
```

Java ArrayList:

```
import java.util.ArrayList
void add(int index, E element)
    Inserts the specified element at the specified position in this list.
E get(int index)
    Returns the element at the specified position in this list.

E remove(int index)
    Removes the element at the specified position in this list.
```

Aufgabenverzeichnis

Frage 1 OS: Begriffe.....	3
Frage 2 OS: Semaphore.....	4
Frage 3 OS: FIFO threadsafe.....	5
Frage 4 DB: Check.....	7
Frage 5 DB: Entity-Relationship-Diagramm.....	7
Frage 6 DB: SQL I.....	9
Frage 7 DB: is_uni.....	10
Frage 8 DB: Normalformen I.....	12
Frage 9 DB: Normalformen II.....	12
Frage 10 DB: Normalform PKW.....	13
Frage 11 PROG-C: Pointer und Array.....	14
Frage 12 PROG-C: Speicherklassen.....	15
Frage 13 PROG-C: maxi.....	15
Frage 14 PROG-C: Ausgabe.....	16
Frage 15 PROG-C: Stack.....	16
Frage 16 PROG-C++: out.....	18
Frage 17 PROG-C++: polygon.....	19
Frage 18 PROG-C++: cipher.....	20
Frage 19 PROG-C++: mystring.....	21
Frage 20 PROG-C++: Matrix.....	23
Frage 21 PROG-C++: EAN.....	24
Frage 22 SENG: UML-Personen.....	26
Frage 23 SENG: Rekursion I.....	28
Frage 24 SENG: Rekursion II.....	29
Frage 25 SENG: quicksort.....	30
Frage 26 SENG: Performance Sort.....	31
Frage 27 SENG: Search.....	31
Frage 28 Listen I.....	32
Frage 29 Bäume I.....	33
Frage 30 Bäume II.....	33
Frage 31 Bäume III.....	34
Frage 32 SENG: Singleton (+).....	35
Frage 33 SENG: Observer (+).....	36
Frage 34 NW-PROG: Socket-API I.....	37
Frage 35 NW-PROG: Socket-API II.....	37
Frage 36 NW-PROG: SSL/Zertifikate.....	41
Frage 37 NW-PROG: Diffie-Hellman.....	43