Fragenkatalog Schwerpunktthemen Informatik Q11/12 – Lösung

11/1 – Aufgabengebiet 1: Rekursive Funktionen und Methoden:

1. Erkläre den Begriff „Rekursive Funktion/Methode“!

   Eine Funktion/Methode ist rekursiv, wenn sie innerhalb ihres Funktionsterms den eigenen Funktions/Methodenbezeichner (evtl. sogar mehrfach) enthält.

2. Erstelle eine Methode, um die Fakultät / die n-te Potenz einer Zahl rekursiv zu berechnen.

```
int fakultaet(int n){
   if (n==1){
      return 1;
   }else
   return n*fakultaet(n-1);
}
```

3. Erkläre das Prinzip der Rekursion am Beispiel des „Turm von Hanoi“!

   Ein Turm einer bestimmten Höhe n wird von a mit Hilfe von b nach c gebracht, indem ein Turm mit Höhe n-1 von a nach b, dann eine einzelne Scheibe von a nach c und dann wieder ein Turm mit Höhe h-1 von b nach c gebracht wird. Man hat also ein Problem für eine Größe n zurückgeführt auf ein Problem der Größe n-1.

4. Was leistet die angegebene Methode? Forme sie um, dass sie rekursiv arbeitet!

   Die Methode berechnet die Potenz b^n.

```
int potenz(int n, int b){
   if (n==0){
      return 1;
   }else
   return n*potenz(n-1,b);
}
```

5. Erkläre das Vorgehen, um einen Knoten rekursiv in eine Liste am Ende einzufügen.

   in Liste:

```
void hintenEinfuegen(DATENELEMENT del){
   erster.hintenEinfuegen(del);
}
```

   in Listenelement:

```
LISTENELEMENT hintenEinfuegen(DATENELEMENT del){
   nachfolger = nachfolger.hintenEinfuegen(del);
   return this;
}
```

   in Abschluss:

```
LISTENELEMENT hintenEinfuegen(DATENELEMENT del){
   KNOTEN neuKnoten = new KNOTEN(this, del);
   return neuKnoten;
}
```

6. Formuliere eine rekursiv arbeitende Methode, um die Anzahl der Elemente eines Baums zu zählen.

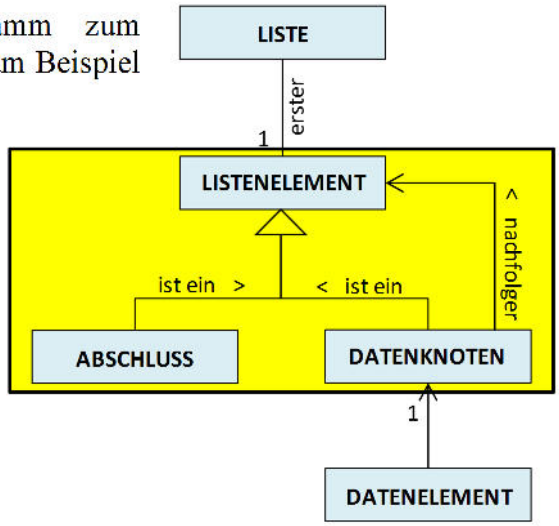
In Baum:
```
return wurzel.anK();
```
In Knoten:
```
int anzK(){
   return 1 + nachfolgLinks.anzK() + nachfolgRechts.anzK();
}
```
In Abschluss:
```
return 0;
```

7. Erkläre, inwiefern beim Graphendurchlauf Rekursion benötigt wird.

z.B. bei der Tiefensuche: hier wird, beginnend mit einem Startknoten, die Tiefensuche rekursiv durchgeführt, indem der Knoten markiert wird und dann die Methode für unbesuchte, direkt erreichbare Knoten rekursiv aufgerufen wird.

11/1 – Aufgabengebiet 2: Das Entwurfsmuster Kompositum:

8. Skizziere das Klassendiagramm zum Entwurfsmuster Kompositum am Beispiel der einfach verketteten Liste.



9. Erkläre das Entwurfsmuster Kompositum!

Es ist ein Entwurfsmuster in der Informatik, um eine verschachtelte, komlexe Struktur, z.B. einen Baum, einheitlich zu behandeln, unabhängig davon ob das betrachtete Element ein Abschluss/Blatt oder ein Behälter/Knoten für weitere Elemente ist.

10. Welche Methoden sollten bei der Implemetierung des Entwurfsmusters für die Knoten einer Liste zur Verfügung gestellt werden?

```
void inhaltSetzen(DATENELEMENT)
KNOTEN(LISTENELEMENT, DATENELEMENT)
LISTENELEMENT nachfolgerGeben()
DATENELEMENT inhaltGeben()
void nachfolgerSetzen(LISTENELEMENT)
```

11. Erkläre an zwei Beispielen, wie Methoden, je nachdem ob sie für ein LISTENELEMENT oder einen ABSCHLUSS definiert sind, unterschiedlich implementiert werden.

```
LISTENELEMENT nachfolgerGeben()
return this; / return nachfolger;
DATENELEMENT inhaltGeben()
return inhalt / return null;
```

12. Erkläre die Umsetzung des Entwurfsmusters Kompositum für die Methode `sortiertEntfernen`

In Knoten:

```
LISTENELEMENT sortiertEntf(DATENELEMENT suchinhalt) {
   if (inhalt.istGleich(suchinhalt)){
      return nachfolger;
   }
   else{
      if (inhalt.istKleiner(suchinhalt)){
         nachfolger = nachfolger.sortiertEntf(suchinhalt);
      }
      return this;
   }
}
```

In Abschluss:

```
LISTENELEMENT sortiertEntf(DATENELEMENT suchinhalt){
   return this;
}
```

13. Wie unterscheidet sich das Entwurfsmuster Kompositum bei den Strukturen BAUM und LISTE?

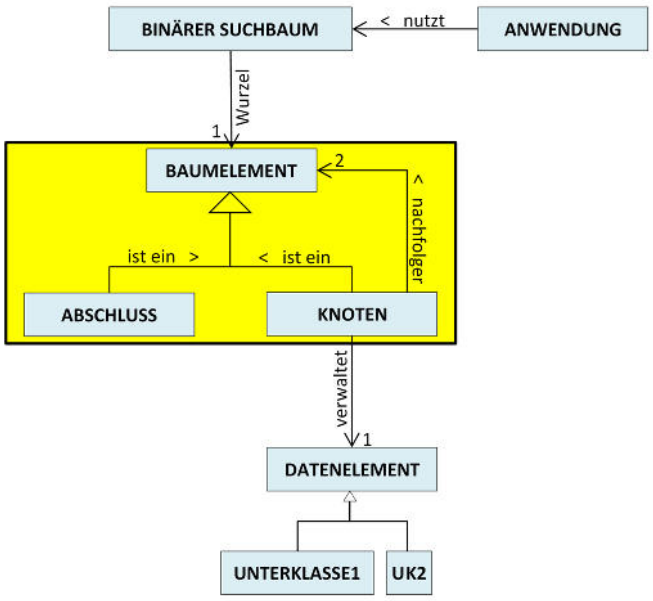
Entscheidender Unterschied ist, dass die Knoten beim Baum zwei Nachfolger haben (können), bei der Liste nur einen.

11/1 – Aufgabengebiet 3: Binärbäume:

14. Erkläre den Begriff „Binärer Suchbaum“.

Ein Binärbaum ist geordnet (=binärer Suchbaum), wenn gilt:
- Falls ein linker Teilbaum existiert, so ist jeder seiner Schlüsselwerte kleiner als der Schlüsselwert des aktuellen Knotens
- Falls ein rechter Teilbaum existiert, so ist jeder seiner Schlüsselwerte größer als der Schlüsselwert des aktuellen Knotens

15. Skizziere die Klassenstruktur des binären Suchbaums.



16. Erkläre unterschiedliche Traversierungsstrategien.

Preorder und Postorder entsprechend.

```
inorder (Knoten k)
   Falls k linken Kindknoten lk hat, inorder(lk)
   Datenwert von k bearbeiten
   Falls k rechten Kindknoten rk hat, inorder(rk)
```

17. Traversiere den vorliegenden Baum mit der Strategie inorder/preorder/postorder.

Preorder: k0 – k1 – k3 – k6 – k2 – k4 – k5
Inorder: k3 – k6 – k1 – k0 – k4 – k2 – k5
Postorder: k6 – k3 – k1 – k4 – k5 – k2 – k0

18. Formuliere eine Methode sortiertEinfuegen für einen sortierten Binärmbaum.

In Knoten:

```
KNOTEN sortiertEinfueg(DATENELEMENT inhaltNeu){
    if (inhalt.istKleiner(inhaltNeu)){
       nachfolgerR = nachfolgerR.sortiertEinfueg(inhaltNeu);
       return this;
    }else{
       nachfolgerL = nachfolgerL.sortiertEinfueg(inhaltNeu);
       return this;
    }
}
```

In Abschluss:

```
KNOTEN sortiertEinfuegen(DATENELEMENT inhaltNeu){
    return new KNOTEN(this, new ABSCHLUSS(), inhaltNeu);
}
```

19. Erkläre das Löschen eines Knotens aus einem Binärbaum.

Blatt wird einfach gelöscht. Innerer Knoten mit nur einem Nachfolger auch klar. Bei zwei Nachfolgern ersetz man zu löschenden Knoten durch linkesten Nachfolger des rechten Teilbaums. Dieser ist logischerweise Blatt oder hat nur einen Nachfolger.

20. Nimm das Löschen des Knotens X aus dem vorliegenden Binärbaum vor.

Je nach Vorgabe. Beispiel siehe Skript.

21. Beschreibe einen Binärbaum unter Zuhilfenahme der Begiffe Höhe, entartete Liste, vollständig.

Eine Baumstruktur liegt vor, wenn genau ein Objekt, die Wurzel, nicht referenziert wird. Von ihr aus kann jedes weitere Objekt der Baumstruktur durch sukzessives Verfolgen der Referenzen auf genau einem Weg erreicht werden kann, somit ist also jedes Objekt außer der Wurzel genau einmal referenziert wird. Binär bedeutet, dass jedes Objekt höchstens zwei Nachfolger hat. Gibt es jeweils nur einen Nachfolger, dann spricht man von einer entarteten Liste. Ist ein Baum vollständig, dann ist jede Ebene bis auf die letzte vollständig gefüllt. Die Höhe eines Baumes entspricht der Anzahl der Ebenen.

11/2 – Aufgabengebiet 1: Graphen, Wege durch Graphen:

22. Durch welche Bestandteile wird ein Graph festgelegt?

Ein Graph wird durch die Menge seiner Knoten und Kanten (Verbindungen zwischen Knoten) festgelegt.

23. Nenne verschiedene Fakten zu Kanten!

- Man unterscheidet zwischen gerichteten und ungerichteten Kanten.
- Kanten können gewichtet sein
- Zwischen zwei Knoten können mehrere Kanten auftreten (Scottland Yard)

24. Erkläre den Begriff Eulerkreis / Gibt es für den vorliegenden Graph einen Eulerkreis?

Weg durch den Graphen mit gleichem Start- und Endknoten, in dem jede Kante genau einmal vorkommt (Brückenproblem). Gibt es nur, wenn der Grad eines jeden Knotens gerade ist --> Graph anschauen und entscheiden

25. Erkläre den Begriff Isomorphie im Zusammenhang mit Graphen. Welche der angegebenen Graphen sind isomorph?

Graphen sind isomorph, wenn sie die gleiche Adjazenzmatrix darstellen. Hier sind das die Graphen 1, 2 und 4.

26. Wie unterscheiden sich unzusammenhängende, stark- und schwach zusammenhängende Graphen? (Anwenden können auf vorgelegte Graphen, bzw. hinzeichnen können)

Unzusammenhängende Graphen: Mindestens ein Knoten lässt sich von keinem anderen Knoten erreichen
Stark zusammenhängende Graphen: Jeder Knoten lässt sich von jedem anderen Knoten aus erreichen (im gerichteten Graphen)
Schwach zusammenhängende Graphen: Es gibt Knoten die zwar von anderen Knoten erreicht werden können, von denen aus man aber nicht alle anderen erreichen kann. (Einbahnstraßen-Sackgassen) Nur bei gerichteten Graphen möglich.

27. Was ist ein zyklischer Graph? Was ist ein Pfad? Was ist die Länge eines Pfades?

Ein Graph der mindestens einen Zyklus enthält. Ein Zyklus ist ein Pfad, bei dem Anfangs- und Endknoten identisch sind, und mindestens 3 Knoten enthalten sind. Die Länge eines Pfades ist die Anzahl der Knoten auf dem Weg bzw. die Summe der Kantengewichte auf dem Weg.

28. Ist der folgende Graph gerichtet, zyklisch, gewichtet?

Der Graph ist gerichtet, zyklisch und nicht gewichtet.

29. Erstelle eine Adjazenzmatrix für einen vorliegenden Graphen. Welchen Spezialfall für eine Adjazenzmatrix gibt es? (erkennen können)

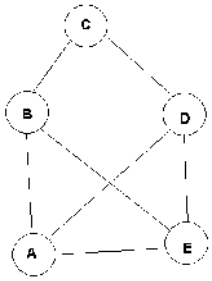
	A	B	C	D	E
A	W	W	W	F	F
B	W	W	F	F	W
C	W	W	W	W	F
D	F	F	W	W	F
E	F	F	W	F	W

Bei einem ungerichteten Graphen ist die Adjazenzmatrix symmetrisch.

30. Wie unterscheidet sich diese von einer Adjazenzmatrix eines Graphen mit gewichteten Kanten? Was könnte die Gewichtung bedeuten?

Bei einem Graphen mit gewichteten Kanten wird in die Matrix die Gewichtung eingetragen. Bei einer Straßenkarte z.B. Entfernung, Zeit, Art der Straße etc.

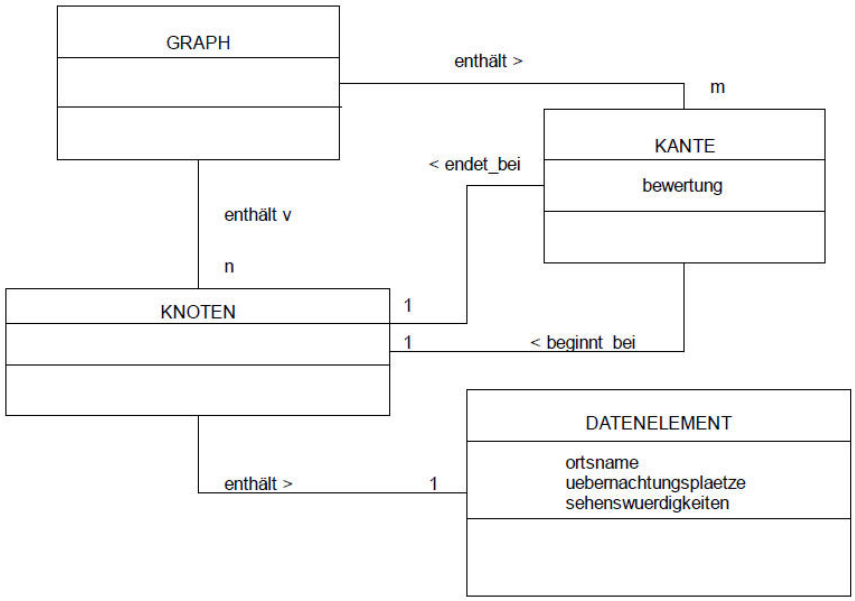
31. Erstelle einen Graphen zu folgender Adjazenzmatrix. Welche Figur entsteht dabei?



32. Wann ist ein Graph ein Baum?

ungerichteter Graph: es kommen keine Zyklen vor
gerichteter Graph: Zyklenfreiheit + zusätzlich hat jeder Knoten genau einen Vorgänger

33. Wie sieht das Klassendiagramm eines Graphen aus?



34. Erstelle die Klasse GRAPH mit dem dazugehörigen Konstruktor!

```
public class GRAPH {
   private KNOTEN[] knotenliste;
   private boolean[][] adjazenzmatrix;
   private int maxAnzahl;
   private int anzahl;
   public GRAPH(int m){
      maxAnzahl = m;
      anzahl = 0;
   knotenliste = new KNOTEN[m];
   adjazenzmatrix = new boolean[m][m];
}
```

35. Implementiere eine Methode, um in einen ungerichteten Graphen eine Kante einzufügen!

```
public void kanteEinfuegen(int i, int j){
   if (i == j || i >= anzahl || j >= anzahl){
      System.out.println("Knoten gibts nicht");}
   else{
      adjazenzmatrix[i][j] = true;
      afjazenzmatrix[j][i] = true;}
}
```

Es müssen zwei Kanten eingefügt werden, da Graph symmetrisch!

36. Implementiere die Klasse Knoten. Diese Klasse soll ein Attribut „inhalt“ vom Typ Datenelement enthalten, sowie eine Methode um den Inhalt zu setzen und auszugeben.

```
class KNOTEN{
   DATENELEMENT inhalt;
   KNOTEN(DATENELEMENT d){
      inhalt = d;
   }
   DATENELEMENT inhaltGeben(){
      return inhalt;
   }
}
```

11/2 – Aufgabengebiet 2: Verfahren für den Graphendurchlauf:

37. Erläutere kurz die Funktionsweise der Tiefensuche. Warum wird dabei jeder Knoten erreicht? (Rekursion erklären können!)

1. Suche einen beliebigen Startknoten aus
2. Markiere alle Knoten des Graphen als „unbesucht“.
3. Führe die Tiefensuche für einen Knoten K, beginnend mit dem gewählten Startknoten rekursiv durch:
   - Markiere K als besucht und gebe Inhalt des Knotens aus
   - Solange K einen noch unbesuchten, direkt erreichbaren Knoten T hat, führe die Tiefensuche (für diesen Knoten) rekursiv durch (Aufruf von Schritt 3)
   - Inhalt des Knotens ausgeben (für Rückweg)

Dabei Reihenfolge festlegen, z.B. durch Adjazenzmatrix oder Alphabet. Bei der Rückkehr aus einer tieferen Rekursionsebene wird der Pfad rückwärts gegangen.

38. Welchen Pfad durchläuft die Tiefensuche bei dem vorliegenden Graphen? Adjazenzmatrix dazu aufstellen!

A – T – G – H – S – H – X – H – G – T – N – T – A

		A	G	H	N	S	T	X
		0	1	2	3	4	5	6
A	0	F	F	F	F	F	W	F
G	1	F	F	W	F	F	W	F
H	2	F	W	F	F	W	F	W
N	3	F	F	F	F	F	W	F
S	4	F	F	W	F	F	F	F
T	5	W	W	F	W	F	F	F
X	6	F	F	W	F	F	F	F

39. Welchen Pfad durchläuft die Tiefensuche hier? Welches Problem stellt sich auf und warum?

A – T – G – H – S – H – X – H – G – T – A

Nicht alle Knoten werden besucht, da der Graph unzusammenhängend ist. Die Tiefensuche erreicht nur für zusammenhängende Graphen alle Knoten.

40. Ein Graph soll in Java implementiert werden. Dafür ist dieser Konstruktor gegeben. Erstelle davon ausgehend eine Methode um einen Knoten einzufügen.

```
public void knotenEinfuegen(KNOTEN k){
   if (anzahl >= maxAnzahl){
      System.out.println("Liste voll");}
   else{
      knotenliste[anzahl] = k;
      anzahl++;}
}
```

41. Welche zusätzlichen Methoden und Attribute sind in den Klassen KNOTEN und GRAPH nötig, um die Tiefensuche umzusetzen?

Attribute: markierung (Attribut vom Typ boolean)
Methoden: markierungSetzen, markierungGeben, alle markierungen auf false setzen

42. Wie inisialisiert man einen Graphen für die Tiefensuche in JAVA?

```
void graphInitialisieren(int startknoten){
   for (int i = 0; i<anz; i++){
      knotenliste[i].markeSetzen(false);
   }
   tiefensuche(startknoten);
}
```

43. Setze den Algorithmus der Tiefensuche in JAVA oder zumindest in Pseudocode um.

```
void tiefensuche(int index){
   knotenliste[index].markeSetzen(true);
   System.out.print("[");
   knotenliste[index].inhaltGeben().kuerzelGeben();
   for (int i=0; i<anz; i++){
      if ((knotenliste[i].markeGeben()==false)
           &&(admat[i][index] == true)){
         tiefensuche(i);
         knotenliste[index].inhaltGeben().kuerzelGeben();
         System.out.print("]");
      }
   }
}
```

44. Erläutere den Algorithmus von Dijkstra!

1. Weise allen Knoten die beiden Eigenschaften "Distanz" und "Vorgänger" zu. Initialisiere die Distanz im Startknoten mit 0 und in allen anderen Knoten mit ∞.
2. Solange es noch unbesuchte Knoten gibt, wähle darunter denjenigen mit minimaler Distanz aus und
   - Speichere, dass dieser Knoten schon besucht wurde
   - Berechne für alle noch unbesuchten Nachbarknoten die Summe des jeweili-gen Kantengewichtes und der Distanz im aktuellen Knoten
   - Ist dieser Wert für einen Knoten kleiner als die dort gespeicherte Distanz, aktualisiere sie und setze den aktuellen Knoten als Vorgänger. (Update oder Relaxieren)

45. Welche Attribute und Methoden benötigt man für die Umsetzung von Dijkstra?

In KNOTEN:
Attribute distanz, besucht, (und vorgaenger)
Methoden distanzSetzen, distanzGeben, besuchtSetzen, unbesuchtSetzen.

In GRAPH:
Adjazenzmatrix für gewichteten Graph
zweites Feld als Warteschlange für die Knoten.
Methode dijkstraStart für Startknoten, Initialisierung, Speichern der Knoten in korrekter Reihenfolge in der Warteschlange
Methode dijkstra führt den Algorithmus aus.

46. Erläutere den Algorithmus der Breitensuche!

1. Auswahl eines beliebigen Startknotens s
2. Initialisierung des Graphen:
   a) Markieren aller Knoten bis auf den Startknoten:
      - Bearbeitungszustand „unbesucht“
      - Entfernung = „unendlich“
      - Vorgängerknoten = null
   b) Markieren des Startknotens:
      - Bearbeitungszustand = „in Bearbeitung“
      - Entfernung = 0
      - Vorgängerknoten = null
3. Einfügen des Startknotens in eine Warteschlange
4. Entnahme eines Knotens „aktuell“ vom Beginn der Warteschlange, Ausführung folgender Schritte für jeden unbesuchten Nachbarn/Nachfolger v des gerade entnommenen Knotens:
   a) Markiere v als in Bearbeitung
   b) Setze die Entfernung von v um eins höher als die von Knoten „aktuell“
   c) Weise Knoten „aktuell“ als Vorgängerknoten zu
   d) Reihe v in die Warteschlange ein
   e) Markiere Knoten v als fertig bearbeitet
5. Wiederhole Schritt4 solange Warteschlange nicht leer ist

47. Führe den Algorithmus für vorliegenden Graphen durch!

Übungsblatt dazu auf www.oppelt-help.de , im Colloquium natürlich knapper

48. Gib den Graphendurchlauf für folgenden Graphen an, der als Breitensuche abgearbeitet werden soll.

S – A – D – C – E – G – L – P – F

49. Nenne drei verschiedene Verfahren für den Graphendurchlauf und unterscheide sie bezüglich ihres Einsatzbereiches.

Breitensuche (Auffinden von kürzesten Pfaden, z.B. möglichst wenige U-Bahn-Stationen), Tiefensuche (Besuchen aller Knoten - Handlungsreisendenproblem), Algorithmus von Dijkstra (Auffinden von kürzesten Wegen)

11/2 – Aufgabengebiet 3: Vorbereitung und Durchführung von Softwareprojekten:

50. Wodurch ist Projektarbeit allgemein gekennzeichnet?

Kennzeichen einer Projektarbeit sind ein verbindlicher Zeitrahmen, eine klare, überprüfbare Zielvorgabe, und begrenzte finanzielle, materielle und personelle Ressourcen. Dazu kommt eine ständige Überprüfung, Korrektur und Dokumentation des Verlaufs (DIN-Norm).

51. Nenne die Phasen eines geregelten Projektmanagements.

1. Projektdefinition
2. Projektplanung
3. Projektdurchführung
4. Projektabschluss

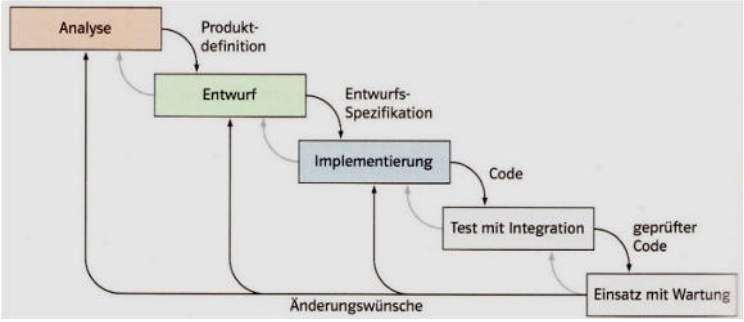
52. Welche Schritte sind bei der Vorbereitung (=Definition+Planung) von Projekten relevant?

1. Brainstorming (Ideen und Informationen sammeln, erste Vorschläge, Mindmap)
2. Zielsetzung (Sachziele, Kostenziele, Terminziele, Pflichtenheft)
3. Projektorganisation (Projektleiter und Arbeitsgruppen)

4. Abschnitte (Einteilung des Projekts in Abschnitte, Meilensteinplan)
5. Projektplanung (Strukturplan mit Arbeitsteilung und Schnittstellenbeschrei-bung)
6. Aufwandsschätzung (in Personenstunden)
7. Zeitplanung (vor allem parallele Abläufe planen, Netzplan erstellen)

53. Beschreibe das Wasserfallmodell als Modell zur Durchführung eines Softwareprojekts.

Vorgehensmodell zur Strukturierung, das die Vorgehensweise in eine sequentielle Folge von Phasen trennt.



54. Erkläre die einzelnen Phasen des Wasserfallmodells

1. Phase: Analyse
Anforderungsermittlung:
- Auftraggeber erstellt ein Lastenheft, daraus Erstellung eines detaillierten Pflichtenhefts durch den Auftragnehmer
- Funktionale Anforderungen (Funktionen des Programms)
- Nichtfunktionale Anforderungen (Sicherheit, Verfügbarkeit, Benutzerfreund-lichkeit, Kompatibilität)
- Aufwandsschätzung (siehe §4)

Systemanalyse:
- Modelle, um das geplante System zu verstehen
- Datenflussdiagramme, Zustandsmodelle, Klassenmodelle mit einem Systemmodell als detaillierte, strukturierte Produktdefinition

2. Phase: Entwurf
- Planung von Form und Struktur der Lösung durch Aufteilung in Teilsysteme (Subsysteme, Komponenten, Codebausteine, Klassen).
- genaue Festlegung der Schnittstellen
- für Teillösungen kann z.T. auf evtl. schon vorhandene Entwurfsmuster zurück gegriffen werden (z.B. Kompositum)

3. Phase: Implementierung
- Umsetzung im Programmcode meist einer Programmiersprache
- z.T. Einbindung von vorhandenen Bibliotheken
- lauffähige Software durch Übersetzung erstellen

4. Phase: Test mit Integration
- schrittweiser Einbau der Software in die Hard- und Softwareumgebung, jeweils Testen
- Tests mit bestimmten Eingabedaten um korrekte Verarbeitung zu prüfen
- Tests können aber keine Fehlerfreiheit garantieren

5. Phase: Einsatz mit Wartung
- „Kundendienst“
- Fehlerbeseitigung, Einbau neuer Funktionen
- Anpassungen an Hardware-, Softwareumgebungen
- Absicherung gegen Hacker oder Viren

55. Was fällt in den Bereich des Projektabschlusses bei einem Softwareprojekt?

Auftragsprojekte enden mit Abnahme durch Auftraggeber,
Gründe für Abweichungen festmachen
Analyse des Verlaufs und Dokumentation der Erkenntnisse für künftige Projekte

56. Nenne aus dem Wasserfallmodell heraus weiter entwickelte Modelle

- Spiralmodell, V-Modell, Extreme Programming

57. Nenne verschiedene Methoden zur Aufwandsschätzung. Wozu wird sie durchgeführt?

Analogie: Vergleich mit ähnlichen Projekten (müssen entsprechend dokumentiert sein)
Schätzklausur: Mittelwert aus unabhängigen Schätzungen mehrerer Personen mit evtl. zwei Durchgängen (nach Fehlerbeseitigung)
Aufwand pro Einheit: Aufteilung größerer Projekte in Teileinheiten und Schätzung dieser Einheiten mit anschließender Addition
Prozentsatzmethode: Hochrechnung von einzelnen Einheiten auf das Gesamtprojekt, dabei evtl. erste reale Ergebnisse abwarten
Wird durchgeführt, um zeitlichen und vor allem geldlichen Umfang eines Projekts abschätzen zu können.

58. Erkläre das COCOMO-Modell

Verfahren zur Aufwandschätzung bei Softwareprojekten, das die Codezeilen in leicht, mittel und schwer einteilt, damit und mit der Gesamtzahl der Codezeilen den Aufwand berechnet $A = m \cdot L^n$ und daraus eine Mindestpojektdauer errechnet $T = 2{,}5 \cdot A^p$. Bedeutung aber abnehmend.

59. Welche Modellierungstechniken kommen bei der Softwareentwicklung zum Einsatz?

Funktionale Modellierung:
Datenflussdiagramme beschreiben Subsysteme, Schnittstellen, Informationsflüsse. Subsysteme werden wiederum in Komponenten zerlegt
Objektorientierte Modellierung:
zeigt mit Objekt- und Klassendiagrammen Klassen, Objekte, Attribute, Methoden, Assoziationen und Klassenhierarchien. Klassenbeziehungen immer mit Kardinalität (hier: Multiplizität) ohne Methoden zur reinen Datenmodellierung geeignet (Datenbanken)
Zustandsmodellierung:
Abläufe als Folge von Objektzuständen durch Übergänge miteinander verbunden mit auslösenden (z.T. auch ausgelösten) Aktionen
Algorithmen:
Ablaufstruktur als Kombi von elementaren und zusammengesetzten Verarbeitungsschritten: Sequenz, Alternative, Wiederholung (mit Bedingung / fester Anzahl)
Szenariomodell, Sequenzdiagramm:
zum Verhalten von Objekten in einem Anwendungsszenarium, beschreiben den zeitlichen Ablauf der Kommunikation von Objekten, die durch den Aufruf von Methoden erfolgt. Informationsaustausch über Parameterwerte und Rückgabewerte

60. Erkläre eine der Modellierungstechniken genauer.

siehe oben.

61. Erkläre das Software-Muster Model-View-Controller.

Muster zur Strukturierung von Softwareentwicklung in drei Einheiten / Aufteilung des Softwaresystems in drei Komponenten, dadurch später leichter zu ändern / warten / erweitern.
Datenschicht – Modell:
eigentlicher funktionaler Kern des Systems, enthält die darzustellenden Daten. Unabhängig von Präsentation und Steuerung
Präsentationsschicht – View:
Sichten auf die Daten, Darstellung der benötigten Daten aus dem Modell für den Benutzer, ggf. Weiterleiten der Eingaben des Benutzers

Steuerungsschicht – Controller:
verwaltet eine oder mehrere Views, nimmt aus diesen Benutzeraktionen entgegen, wertet diese aus und leitet entsprechend Folgeaktionen ein.

12/1 – Aufgabengebiet 1: Notationsformen für formale Sprachen:

62. Was ist eine formale Sprache? Was bedeutet Syntax und Semantik?

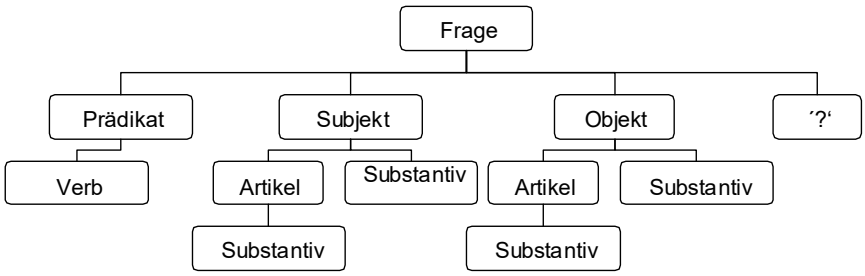
Eine formale Sprache ist die Menge aller zulässigen Zeichenketten über einem Alphabet, die durch formale Regeln zur Bildung von zulässigen Zeichenketten definiert ist (Syntax der Sprache) und zur eindeutigen Kommunikation dient. Semantik nennt man die Bedeutung eines Wortes.

63. Nenne Beispiele für formale Sprachen aus dem Alltag!

- Waschzettel
- Chemische Reaktionsgleichungen
- Backofensymbolik
- Autokennzeichen

64. Mit welchen Regeln könnte man einen deutschen Fragesatz der Art „Fährt Michael einen Ferrari?“ ableiten? Stelle die Ableitung als Baum dar.

R1: <Frage> → <Prädikat><Subjekt><Objekt> ´?'
R2: <Subjekt> → <Substantiv> | <Artikel> <Substantiv>
R3: <Prädikat> → <Verb>
R4: <Objekt> → <Substantiv> | <Artikel> <Substantiv>
R5: <Artikel> → „der“ | „die“ | „das“ | „den“ | „einen“



65. Wofür kann Rekursion in der formalen Sprache genutzt werden?

Um Zeichenketten darzustellen, die beliebige Länge haben können, z.B. Palindrome oder die Menge der natürlichen Zahlen.

66. Was ist der Unterschied zwischen einem Terminal und einem Nichtterminal?

Ein Terminal ist ein Symbol aus dem Alphabet Σ, welches im weiteren Ableitungsverlauf nicht mehr ersetzt wird. Ein Nichtterminal hingegen wird noch ersetzt.

67. Was bedeutet der Begriff Grammatik? Was ist eine Ableitung?

Jeder formalen Sprache liegt eine Grammatik zugrunde. Durch sie werden alle zulässigen Wörter der Sprache mit Hilfe von Ableitungsregeln beschrieben. Die Grammatik besteht aus einer Menge von syntaktischen Variablen, einem Alphabet, einer Startvariablen und einer Menge von Produktionen. Eine Ableitung ist die Erzeugung eines Wortes nach den Regeln der Grammatik.

68. Leite die Zahl 2011 mit Hilfe der folgenden Grammatik ab:

<zahl> → <zahl> <ziffer>

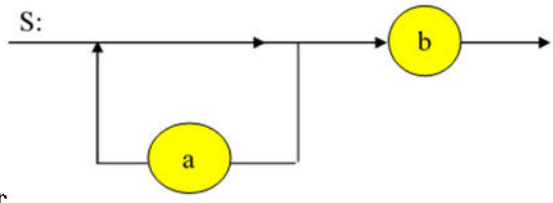
→ <zahl> <ziffer> <ziffer>
→ <zahl> <ziffer> <ziffer> <ziffer>
→ <z> <ziffer> <ziffer> 1
→ 2011

69. Gib die Produktionen einer formalen Sprache an, die die Menge aller natürlichen Zahlen darstellt.

Siehe Frage davor.

70. Welche Möglichkeiten der Darstellung einer Ableitung gibt es? Zeige diese am Beispiel dieser Grammatik.

(Erweiterte) Backus-Naur-Form: S = {‘a’}’b’;
Syntaxdiagramm:



71. Erkläre die Darstellungsweise der EBNF genauer.

- Statt Pfeil ein „=“
- Keine <>-Klammern bei Nicht-Terminalen
- Nach jeder Regel ein ;
- Zusammenfassung möglich: Statt ‘ac‘|‘bc‘ auch (‘a‘|‘b‘)‘c‘
- Wiederholungen in {}-Klammern (keinmal bis n-mal)
- Optionen durch []-Klammern (keinmal oder einmal)

72. Erkläre die Darstellungsweise des Syntaxdiagrammes genauer.

- Darstellung der Regeln einer Grammatik durch Graphen
- Kreisförmige Symbole als Terminale, Rechtecke als Nichtterminale
- Alternativen durch Verzweigungen
- Optionale Teile auch durch eine Verzweigung mit leerem Zweig
- Wiederholung durch rückwärts gerichteten Pfeil

73. Gehört ´DaDaLiMo‘ zur folgenden Sprache?

Ja: <S> → 'DaDa' <S> → 'DaDa' <L> 'Mo' → 'DaDa' 'Li' 'Mo'

12/1 – Aufgabengebiet 2: Endliche Automaten:

74. Erkläre allgemein die Arbeitsweise eines endlichen Automaten und den Unterschied zwischen deterministischen und nichtdeterministischen endlichen Automaten.

- Endliche Automaten nehmen eine Zeichenkette als Eingabe auf und arbeiten diese ab
- Endzustände sind speziell gekennzeichnet (doppelt umkringelt)
- Mehrere Eingabemöglichkeiten werden durch Kommas trennen
- Für **deterministische** endliche Automaten gilt, dass jedes Eingabezeichen **höchstens eine** Transition auslöst

75. Wie ist ein deterministischer endlicher Automat formal definiert?

1. Endliche Menge Z von Zuständen Z = {z0, z1, z2, …}
2. Endliches Eingabealphabet: Σ = {...}
3. Ein Startzustand, z.B. z0
4. Eine Menge E von Endzuständen
5. Eine zweistellige Übergangsfunktion die einem Paar Zustand/Eingabesymbol einen Folgezustand zuordnet z.B.: f(z0,’0’)=z1

76. Welche Sprachen können von endlichen Automaten erkannt werden?

Reguläre Sprachen: Nichtterminale werden in den Produktionsregeln zu **einem** Terminal oder zu **einem** Nichtterminal und **einem** Terminal.

77. Wie kann man einen Automaten durch eine Grammatik darstellen?

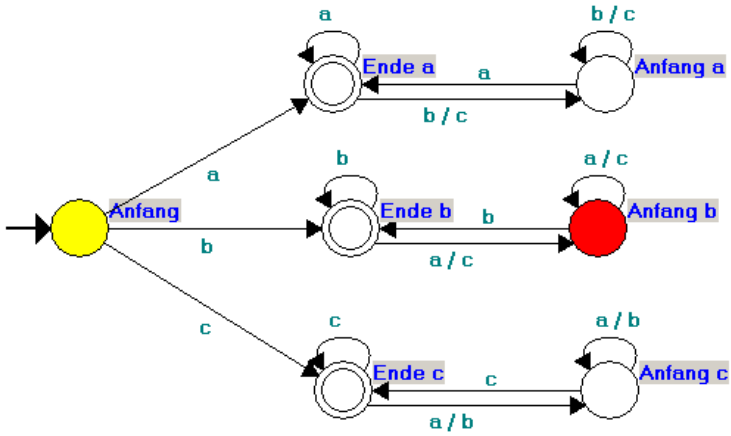
- Z (Zustände) als Menge der Nichtterminale
- Σ (Eingabealphabet) als Menge der Terminale
- Anfangszustand als Startsymbol S
- Pro Transition eine Produktionsregeln
- Pro Endzustand eine Transition vom Endzustand in ein leeres Wort

78. Wie unterscheiden sich Mealy-Automaten von anderen endlichen Automaten?

Bei Mealy-Automaten wird bei jedem Zustandsübergang eine Aktion durchgeführt. So wird z.B. für jede Eingabe auch ein Zeichen ausgegeben. Es kommen also hinzu:

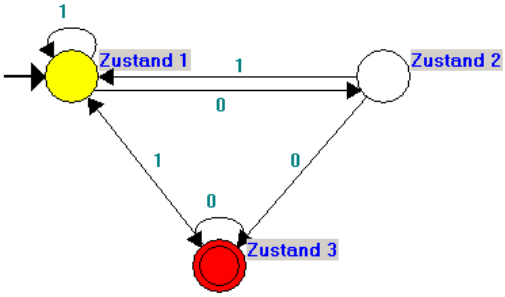
6. Ein endliches Ausgabealphabet Ω
7. Eine Ausgabefunktion die einem Eingabepaar Zustand/Eingabesymbol ein Ausgabezeichen aus Ω zuordnet

79. Skizziere einen Automaten der das Eingabealphabet Σ = {A,B,C} hat und erkennt ob das Eingabewort den gleichen Anfangs- und Endbuchstaben hat.



80. Welche Bedingung muss eine Binärzahl erfüllen um von folgendem Automaten akzeptiert zu werden?

Sie muss mit mindestens zwei 0 hintereinander enden, und ist somit durch 4 teilbar.



81. Erstelle eine Grammatik zu einem gegebenen Automaten (von voriger Frage).

Grammatik: G = (V, Σ,P,S)
Alphabet: Σ = {0,1}
Variablen: V = {<Zustand 1>, <Zustand 2>, <Zustand 3>}
Startvariable: S = <Zustand 1>
Produktionen P:
<Zustand 1> → ‚1' <Zustand 1> | ‚0' <Zustand 2>
<Zustand 2> → ‚1' <Zustand 1> | ‚0' <Zustand 3>
<Zustand 3> → ‚1' <Zustand 1> | ‚0' <Zustand 3>
<Zustand 3> → Ɛ

82. Wie können endliche Automaten mit JAVA simuliert werden?

Es werden zwei Methoden programmiert. In der einen wird die Übergangsfunktion implementiert. Es werden Fallunterscheidungen über die Zustände eine Ebene höher über Fallunterscheidungen über die Eingabesymbole geschachtelt und so jedem Eingabepaar aus Zustand und Eingabesymbol ein Folgezustand zugeordnet. Mit der anderen Methode werden die Wörter untersucht. Hier wird ein Wort Zeichen für Zeichen mithilfe der anderen Methode abgearbeitet. Wenn am Ende ein Endzustand erreicht ist, dann wird das Wort akzeptiert. (Auch aufschreiben können)

12/1 – Aufgabengebiet 3: Nebenläufige Prozesse:

83. Was bedeutet Nebenläufigkeit und Parallelität von Prozessen? Welchen Sinn macht Nebenläufigkeit?

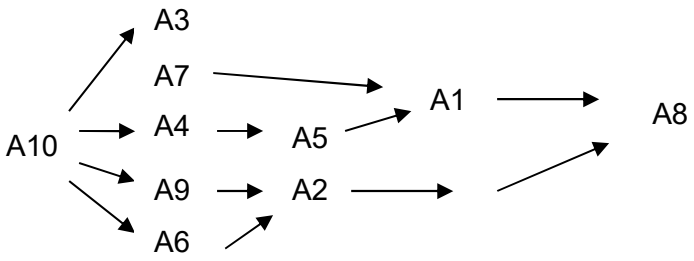
Wenn innerhalb eines Arbeitsvorgangs mehrere Prozesse gleichzeitig aktiv sind, so bezeichnet man diese aktiven Prozesse als nebenläufig, laufen sie tatsächlich gleichzeitig ab, dann als parallel. Gleichzeitig ablaufende Prozesse beschleunigen die Durchführung des Arbeitsvorgangs.

84. Ein Klassenzimmer soll renoviert werden. Es müssen folgende Arbeiten verrichtet werden:

A. A1 → A8
A2 → A8
A4 → A5
A5 → A1
A6 → A2
A7 → A5, A2, A1
A9 → A2
A10 → A1, A3, A6, A9

B. A1, A2, (A3) | A3, A4, A6, A7, A9 | A2, A5 | A7, A10

C.



85. Welche Bedingungen sind für das Eintreten eines Deadlocks nötig? Programmbeispiele!

1. Benötigte Ressourcen können nur im wechselseitigen Ausschluss benutzt werden.
2. Ein Prozess kann weiter Ressourcen anfordern, auch wenn er bereits Ressourcen angefordert hat.
3. Benötigte Ressourcen müssen ununterbrechbar zugeteilt sein.
4. Es gibt mehrere Prozesse, die jeweils auf Ressourcen warten, die von bereits anderen belegt sind.

Z.B. gleichzeitige Buchung von Plätzen im Flugzeug

86. Definiere die Begriffe „kritischer Abschnitt“ und „synchronisieren“ und „ununterbrechbare Ressource“

- kritischer Abschnitt: hier wollen mehrere Prozesse auf dieselben Ressourcen zugreifen.
- Falls dies der Fall ist, müssen sie synchronisiert, also zeitlich aufeinander abgestimmt werden

- die ununterbrechbare Ressource: kann erst nach Beendigung des kritischen Abschnittes dem benutzenden Prozess entzogen werden.

87. Wie kann man mit der Problematik von Deadlocks umgehen?

Behebung: einem Vorgang werden die Betriebsmittel entzogen oder er wird abgebrochen
Verhinderung: vorausschauendes Zuteilen von Betriebsmitteln
Vermeidung: z.B. bessere Speicherzuweisung, Ansprüche auf Betriebsmittel reduzieren

88. Was versteht man unter dem wechselseitigen Ausschluss? Nenne ein Beispiel für einen wechselseitigen Ausschluss.

Wechselseitiger Ausschluss bedeutet, dass ein Prozess nicht auf Ressourcen zugreifen kann, ohne die Nutzung durch einen anderen Prozess auszuschließen. Beispiele: Engstelle bei Straße, Kreuzung, Flussüberquerung auf Steinen, Philosophenproblem

89. Nenne und erläutere Konzepte zur Implementierung nebenläufiger Prozesse.

Semaphorkonzept: Dient der Bewachung des kritischen Abschnitts. Semaphore sind Objekte mit einem Zähler und einer Warteschlange und dienen dazu, die Betriebsmittel zu reservieren, also einen wechselseitigen Ausschluss zu realisieren.
Monitorkonzept: Als Monitore bezeichnet man eine Menge von Attributen und Methoden, die zusammengefasst wird und in ihrer Gesamtheit so bewacht wird, dass zu jedem Zeitpunkt nur ein Prozess auf dieser Menge arbeiten kann. Um Verklemmungen zu vermeiden bzw. aufzulösen, kann der Monitor blockieren bzw. wieder aufwecken.

90. Welche Methoden und Befehle sind bei der Implementierung von THREADS wichtig und welche Funktion haben sie jeweils?

`public static void main(String[] args)`	um Threads zu erzeugen
`public void run()`	was er nach dem Start leisten soll
`Thread(name).start()`	um den Thread zu starten
`Thread.sleep(zahl)`	um den Thread schlafen zu lassen
`notifyAll`	um Prozesse aufzuwecken
`synchronized`	um Programmabschnitte zu schützen

91. Erkläre das Programmierelement Try & Catch am Beispiel der Ganzzahldivsion! Erstelle die benötigte Methode.

Berechnunge, die einen Fehler aufwerfen können, umklammert man mit einem try, was dann bei einem Fehler gemacht werden soll, findet sich im catch-Teil.

```
public void ganzzahlAnteilGeben(){
   try{
      ganzzahlteil = zaehler / nenner;
      System.out.println(ganzzahlteil);
   }catch(ArithmeticException ae){
      System.out.println("Divison durch 0");
   }
}
```

12/2 – Aufgabengebiet 1: Von-Neumann-Rechner:

92. Beschreibe die Von-Neumann-Architektur:

Elementare Rechenoperationen werden im Rechenwerk durchgeführt. Zusammen mit dem Steuerwerk bildet es die CPU (Central Processing Unit). Das Steuerwerk ist für die Steuerung des Ablaufs sowie den Informati-onsaustausch zwischen Speicher- und Rechenwerk

verantwortlich. Im Speicherwerk sind Programm und Daten abgelegt. Das Ein- und Ausgabewerk ist für die Kommunikation mit außen zuständig. Das Bus-System ist ein zentral gesteuertes Übertragungssystem zwischen den Komponenten.Die CPU (Central Processing Unit) verbindet das Steuer- und das Rechenwerk. Über ein Bus System besteht die Verbindung zu dem Speicherwerk sowie zu den Ein- und Ausgabegeräten.

93. Beschreibe Rechenwerk, Speicherwerk und Steuerwerk genauer

Rechenwerk:
- Elektronische Komponenten zur Durchführung von Rechenoperationen
- Addition, Subtraktion, logische Operationen, Größenvergleiche, Sprungbefehle
- Dazu jeweils eine oder zwei Bitfolgen als Eingabe, Ausgabe ist dann eine einzige Bitfolge
- Ein Statusregister speichert z.B. „negativ“ oder „overflow“ ab

Speicherwerk:
- Fortlaufend nummerierte Zellen, die Befehle eines Programmes und Datenwerte speichern
- Jede Zelle bestehend aus elementaren Einheiten (Bit) – meistens 8 Bit (=1 Byte)

Steuerwerk:
- kontrolliert den Programmablauf, indem es je einen Befehl aus dem Speicherwerk liest und ihn interpretiert
- Anschließend spricht es die benötigten Komponenten des Rechenwerks an, und veranlasst die Befehlsausführung
- Gegebenenfalls Speicherung von Ergebnissen in der entsprechenden Speicherzelle
- Abschließend Berechnung der Adresse des nächsten auszuführenden Befehls

94. Erkläre die Binärdarstellung von natürlichen Zahlen.

Stellenwertsystem mit Basis 2. In Achterblöcke gegliedert.

95. Welche Arten von Bussen gibt es? Und wie kann man sie unterscheiden?

Arten:
- Datenbus zur Übertragung von Daten
- Steuerbus zur Steuerung der Kommunikation zwischen den einzelnen Komponenten
- Adressbus zur Übermittlung des Speicherorts von Befehlen und Daten

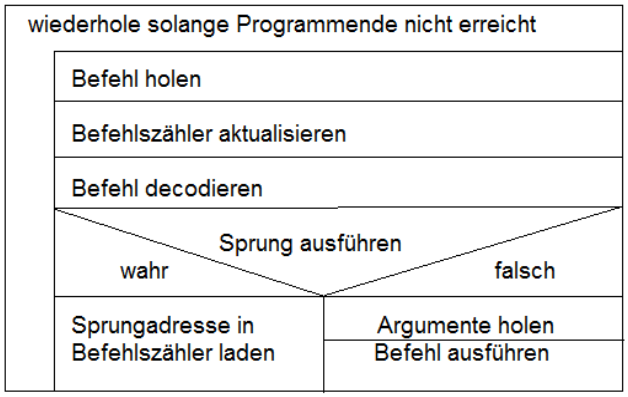
Unterscheidung:
- Paralleler Bus (viele einzelne Leitungen zur parallelen Übertragung von Signalen)
- Universeller Bus (z.B. USB; eine Leitung zur Übertragung von aufeinander folgenden Signalen)

96. Beschreibe den Befehlszyklus eines Befehls in einer Realen Maschine mit den Aufgaben der einzelnen Werke und Busse

- Die Speicheradresse des als nächsten auszuführenden Befehls die im Befehlszähler des Steuerwerks des Prozessors gespeichert ist wird der nächste zu bearbeitende Befehls ins Befehlsregister geladen
- Der Inhalt des Befehlszählers wird über den Adressbus und ein Signal für „Lesen“ über den Steuerbus an den Speicher übertragen.
- Der Befehl wird vom Speicher über den Datenbus ans Steuerwerk übertragen
- Befehlszähleraktualisierung indem der Befehlszähler um die Anzahl der vom Befehl belegten Speicherzellen erhöht wird
- Steuerwerk decodiert das Bitmuster des Befehls im Befehlsregister und führt wenn nötig einen Sprung aus und legt die Speicheradresse der Sprungmarke im Befehlszähler ab
- Steuerwerk aktiviert die zur Befehlsausführung notwendigen Komponenten im Rechenwerk. Erforderliche Daten werden aus dem Speicher in Register des Rechenwerk ablegt (über den Datenbus).

- Ziel und Speicheradressen für den Speicher werden vom Steuerwerk aus dem Inhalt des Befehlsregisters abgelesen und mit dem Adressbus übertragen
- Rechenwerk führt Befehl aus, Ergebnisse werden wenn nötig über den Datenbus in den Speicher übertragen.
- Wiederholung bis Programmende erreicht wurde

97. Beschreibe die Befehlsabarbeitung im Modell anhand eines Struktogramms!



98. Erkläre den Unterschied zwischen einem Von-Neumann-Rechner und einer Registermaschine!

Der Von-Neumann-Rechner ist eine Art Referenzmodell, bei dem Befehle und Daten in einem gemeinsamen Speicher liegen und bildet die Grundlage für die Arbeitsweise heutiger Computer
Die Registermaschine dagegen ist ein Rechnermodell der theoretschen Informatik. Unterschied ist vor allem, dass man von einem unbegrenzten Speicher ausgeht und außerdem jede Speicherstelle beliebig große Zahlen enthalten kann. Befehlsspeicher und Datenspeicher sind getrennt.

12/2 – Aufgabengebiet 2: Registermaschinen:

99. Nenne die Bestandteile der Registermaschine

- Befehlszähler BZ mit der Speicheradresse des als nächstes zu bearbeitenden Befehls
- Statusregister SR mit Informationen über das Ergebnis der letzten Operation
- Datenregister A, R0, R1… zur Ablage der Daten
- Datenregister A (Akkumulator) enthält einen Eingabewert für den folgenden Rechenbefehl und nimmt dessen Ergebnis auf
- (Entspricht der Arbeitsweise aktueller Mikroprozessoren)

100. Erkläre den Begriff Assemblersprache:

Spezielle Programmiersprachen, die nur sehr elementare Befehle zur Verfügung stellen:
- Transportbefehle zum Laden der Datenregister mit Werten (LOAD, DLOAD, STORE)
- Arithmetische Befehle (ADD, SUB, MULT…)
- Sprungbefehle mit und ohne Bedingung (JGE, JUMP)
- Logische Verknüpfungen (AND, OR, Negierung, Invertierung…)
- END zum Beenden des Programms

101. Welche Unterschiede gibt es zwischen der Registermaschine und der realen Maschine?

- Abspeicherung in beliebigen Registern möglich, nicht nur im Akkumulator
- Transportbefehle enthalten Herkunfts- und Zieladresse sowie den Datentyp

- Da verschiedene Datentypen unterschiedliche Bitmusterlängen haben, muss der Datentyp beim Transport angegeben werden (Byte B (8 Bit) - Halbwort H (16 Bit) - Wortform W / einfache Dezimalzahl F (32 Bit) - doppeltgenaue Dezimalzahl D (64 Bit)

102. Erstelle ein Programm in Assembler das den Betrag einer Zahl A berechnet. Beschreibe tabellarisch die Zustandsänderungen der Registermaschine; Setze hierfür A = - 15

```
1: DLOAD A
2: STORE 1
3: JGT 6
4: DLOAD -1
5: MULT 0
6: END
```

Befehl	A	BZ	R1	R2	R3
	0	1	0	0	0
DLOAD -15	-15	2	0	0	0
STORE 1	-15	3	-15	0	0
JGT 6	-15	4	-15	0	0
DLOAD -1	-1	5	-15	0	0
MULT 0	15	6	-15	0	0
END	15	7	-15	0	0

103. Simuliere die Berechnung des Volumens eines Quaders mit den Maßen 10*25*30

```
1: DLOAD 10
2: STORE 0
3: DLOAD 25
4: STORE 1
5: DLOAD 30
6: STORE 2
7: MULT 1
8: MULT 0
9: END
```

104. Beschreibe die Berechnung aus Aufgabe 7 allgemein in JAVA

```
public int berechnen (int l, int b, int h){
    int wert = l*b*h;
    return wert;
}
```

105. Was leistet folgendes Programm?

Es addiert alle Werte von der Zahl 6 bis zur Zahl 0. Also Summe der Zahlen von 1 bis n. Dabei dient R3 als Rückzähler, R2 als Speicher für aktuelle Summe, in Zeile 9 steht die Abbruchbedingung des Programms.

106. Erstelle einen Transportbefehl für eine reale Maschine, der die Register 4 und 6 addiert und das Ergebnis in Register 2 abspeichert. Hierfür soll die Wortform gewählt werden.

ADD W R4, R6, R2

107. Was sagt folgender Befehl aus: MULT W R1, R5, R8? Für welche Maschine ist das ein Transportbefehl?

Multipliziere Wortform in R1 mit R5 und speichere das Ergebnis in R8. Dies ist ein Transportbefehl für eine reale Maschine.

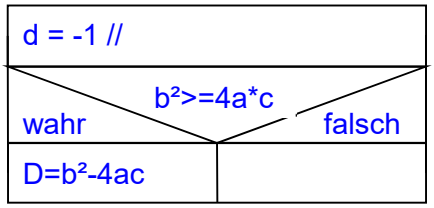
108. Verwandle die Zahl 76 in eine Binärzahl, die in einem Byte gespeichert ist.

$72 = 2^6+12 = 2^6+2^3+4 = 2^6+2^3+2^2$
Binärzahl: 01001100

109. Nenne ein algorithmisches Strukturelement, stelle es graphisch dar und beschreibe die Umsetzung durch eine Registermaschine sowie in JAVA

Bedingte Anweisung // Alternativen // Wiederholung mit Bed. / fester Anzahl Durchläufe

Bedingte Anweisung:




```
public int Diskriminante (int a, int b, int c){
   int d = -1;
   if (b*b >= 4*a*c) {
      d = b*b-4*a*c;
   }
   return d;
}
```

```
1:  DLOAD -1 -- Fehlerwert (spaeter Ergebnis)
2:  STORE 1  -- in R1 speichern
3:  Berechnen von b*b-4*a*c in R2
4:  LOAD 2   -- Ergebnis laden
5:  JGE 7    -- wenn groessergleich 0 überspringen
6:  LOAD 1   -- Fehlerwert laden
7:  END
```

110. Wie kann eine Registermaschine mit JAVA simuliert werden?

Lösung…

12/2 – Aufgabengebiet 3: Laufzeitberechnungen:

111. Wie stellt man das Laufzeitverhalten eines Algorithmus grafisch dar und wie kann man das Ergebnis deuten?

- Für das Aufzeigen exponentiellen Wachstums verwendet man eine halblogarithmische Darstellung der Werte im Koordinatensystem, dann linearer Verlauf.
- Für das Aufzeigen polynomialer Laufzeiten verwendet man eine doppeltlogarithmische Darstellung. Dann linearer Verlauf. Mit Hilfe der Steigung des Graphen lässt sich der Grad des Polynoms bestimmen.

112. Erkläre Bubblesort, Insertionsort, Mergesort, Selectionsort, Quicksort umgangssprachlich

http://www.oppelt-help.de/infodownloads/info12/12_Inf_SkriptSortieralgorithmen_Opp.pdf

113. Formuliere Bubblesort, Insertionsort, Mergesort, Selectionsort, Quicksort in Pseudocode

http://www.oppelt-help.de/infodownloads/info12/12_Inf_SkriptSortieralgorithmen_Opp.pdf

114. Nenne und erläutere drei Verschlüsselungsmechanismen.

Transpositionsalgorithmus: Verschiebung der relativen Position der Zeichen → Spartanische Methode der Skytale

Monoalphabetische Verschlüsselung: jedes Zeichen des Klartextes wird durch ein spezielles andere Zeichen ersetzt, z.B. Cäsar-Code. Besser ist die homophone Verschlüsselung (spezielle, häufig vorkommende Zeichen z.B. „e“ durch verschiedene Zeichen ersetzbar)
Polyalphabetische Verschlüsselung: jeder Buchstabe wird mit einem anderen Schlüssel verschlüsselt, z.B. ENIGMA-Maschine, Vigenère-Verschlüsselung (mit Vigenère-Quadrat)

115. Unterscheide die Begriffe symmetrische und asymmetrische Verschlüsselung.

Symmetrische Verschlüsselung: Sender und Empfänger verwenden den gleichen Schlüssel.
→ Spartaner: Stab mit gleichem Durchmesser
Asymmetrische Verschlüsselung: „Einwegfunktionen“ als Idee der asymmetrischen Verschlüsselung (Beispiel Telefonbuch). Mit dem öffentlichen Schlüssel wird die Nachricht verschlüsselt, was jeder machen kann, mit dem privaten Schlüssel wird sie entschlüsselt

116. Worin liegen die Vorteile der asymmetrischen Verschlüsselung?

- Funktionen zur asymmetrische Verschlüsselung sind Operationen, die in eine Richtung schnell durchführbar sind, in die andere aufgrund exponentiellem Laufzeitverhaltens jedoch nicht, weshalb so eine Verschlüsselung schwer zu knacken ist.
- Es ist keine geheime Schlüsselübergabe nötig

117. Vergleiche das Suchen in einer Liste mit dem Suchen in einem Binärbaum.

Liste: im Schnitt n/2 Vergleiche nötig bis man den richtigen Knoten gefunden hat, also **Laufzeit n.** Baum: im Schnitt nur log n Vergleiche nötig, also **Laufzeit log n**.

118. Welche Entschlüsselungstechniken, mit jeweiliger Laufzeit, gibt es?

- Entschlüsselung durch Häufigkeitsanalysen bei monoalphabetischen Verschlüsselungen, dann nur polynomiale Laufzeit
- Brute-Force-Entschlüsselung durch zufälliges Ausprobieren der einzelnen Verschlüsselungsmöglichkeiten z.B. bei Cäsar-Code, RSA-Methode exponentiell

119. Welche Funktionen verwendet man zur asymmetrischen Verschlüsselung?

- Multiplikation zweier sehr großer Primzahlen, da Primzahlzerlegung sehr aufwendig
- RSA-Verschlüsselung verwendet modulares Potenzieren mit ca. 100 Rechenschritten, wobei die Umkehrung etwa 10^{100} Schritte benötigt