**Zusammenfassung Theoretische Informatik**

**1. Unified Modeling Language**

**UML:** Die Unified Modeling Language ist eine Sprache und Notation zur Spezifikation, Konstruktion, Visualisierung und Dokumentation von Modellen für Softwaresysteme.

**Klassendiagramm:** Klassendiagramme dienen dazu eine oder mehrere Klassen unabhängig von der Sprache abzubilden. Hierbei sollen alle Eigenschaften, Methoden und Beziehungen zu anderen Klassen dargestellt werden. Bsp.:

Ein Bild, das Text, Screenshot, Diagramm, Schrift enthält.

Automatisch generierte Beschreibung

**Objektdiagramm:** Ein Objektdiagramm ist ähnlich wie ein Klassendiagramm aufgebaut, bildet allerdings einzelne Objekte mit ihren Attributen ab. Bsp.:

Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.

Automatisch generierte Beschreibung

**4. Abstrakte Datentypen**

**ADT:** Ein abstrakter Datentyp ist eine Datenstruktur zusammen mit darauf definierten Operationen. In Java ist eine Implementation abstrakter Datentypen durch Klassen und Interfaces möglich. Interfaces geben dabei vor, welche Methoden notwendig sind. In Klassen werden die Methoden dann implementiert.

**5. Liste**

**Def.:** Eine Liste ist eine Folge von Elementen zusammen mit einem sogenannten (ggf. undefinierten) aktuellen Element.

**Methoden:**

* Empty: Liefert true, falls Liste leer ist
* Endpos: Liefert true, wenn das aktuelle Element das letzte ist
* Reset: Das erste Listenelement wird zum aktuellen
* Advance: Der Nachfolger des aktuellen Elements wird zum aktuellen
* Elem: Liefert das aktuelle Element
* Insert: Fügt vor das aktuelle Element ein Element ein und macht das neue Element zum aktuellen
* Delete: Löscht das aktuelle Element und macht den Nachfolger zum aktuellen

**Konzept:**

**Ein Bild, das Diagramm, Entwurf, technische Zeichnung, Reihe enthält.

Automatisch generierte Beschreibung**

**Arten von Listen:**

* Verweisliste: Jeder Eintrag verweist auf das Objekt nach ihm. Das letzte Objekt verweist auf null. Über die Verweise und das aktuelle Element kann auf alle Elemente zugegriffen werden.
* Arrayliste: Alle Inhalte werden in einem zentralen Array gespeichert. Es gibt Variablen, welche den Index der Position und die Anzahl der Elemente speichern.
* Singly-Linked-List: Es wird an das Ende der Liste ein sogenannter Wächter gehangen, auf welchen von dem letzten Element verwiesen wird und welcher wieder auf das erste Element verweist. Die Position zeigt auf das Element vor dem aktuellen Element.
* Doubly-Linked-List: Es gibt wieder ein Wächterelement. Jedes Element zeigt allerdings nicht nur auf seinen Nachfolger, sondern auch auf das Element, welches vor ihm steht.

**Iterator:** Iterator in Java haben folgende drei wichtigen Methoden:

* Hasnext: Liefert true, wenn der Iterator ein nächstes Element hat
* Next: Liefert das nächste Element
* Remove: Löscht das letzte Element, welches zurückgegeben wurde

**6. Keller (Stapel)**

**Def.:** Ein Keller (engl. Stack) ist eine Folge von Elementen zusammen mit einem so genannten Top-Element. Man kann es sich wie einen echten Stapel vorstellen. Kommt ein neues Element hinzu wird es obendrauf gelegt. Wird ein Element entfernt, ist es das oberste.

**Methoden:**

* Empty: Liefert true, wenn der Keller leer ist
* Push: Legt ein Element auf den Keller
* Top: Liefert oberstes Element
* Pop: Entfernt oberstes Element

**Konzept:**

Ein Bild, das Screenshot, Text, Diagramm, Design enthält.

Automatisch generierte BeschreibungEin Bild, das Diagramm, Reihe, technische Zeichnung, Screenshot enthält.

Automatisch generierte Beschreibung

**Type – Casting:** Um in Java zum Beispiel eine Variable des Typen Object als untergeordneten Typen z.b. Schüler zu verwenden muss dieses Object gecastet werden. Dies macht man mit der Klasse, in welche man casten möchte, in Klammern vor dem Object.

**Generic – Types:** Generic Types sind Variable Objekttypen, welche es erlauben Code für mehrere möglichen Eingangsklassen zu definieren.

**Shunting – Yard – Algorithmus:** Unter dem Shunting Yard Algorithmus versteht man einen Algorithmus, welcher eine menschlich verständliche Formel (Infix) in eine maschinell verarbeitbare Formel (Postfix) umwandelt. Bsp.:

* Infix: ((a + b) \* c) / (e – f)
* Postfix: a b + c \* e f - /

Man geht bei dem Algorithmus von Zeichen zu Zeichen und macht folgendes:

* Öffnende Klammer: Man legt die Klammer auf den Stack
* Schließende Klammer: Jedes Element im Stack ausgeben, bis zu einer öffnenden Klammer und beide Klammern löschen
* + oder -: Alle Elemente vom Stack ausgeben, bis zur öffnenden Klammer und das + oder – auf den Stack
* \* oder /: Falls oben auf dem Stack bereits \* oder / liegt dieses ausgeben und dann das neue auf den Stack
* Operand: Sofort ausgeben
* Ausdruck beendet: Den Rest vom Stack ausgeben

**7. Schlange**

**Def.:** Eine Schlange (engl.: Queue) ist eine Folge von Elementen zusammen mit einem so genannten Front-Element.

**Methoden:**

* Empty: Liefert true, wenn die Schlange leer ist
* Enqueue: Fügt ein Element hinten ein
* Front: Liefert vorderstes Element
* Dequeue: Entfernt vorderstes Element

**Konzept:**

**Ein Bild, das Reihe, Text, Quittung enthält.

Automatisch generierte Beschreibung**

**8. Baum**

**Def.:** Ein binärer Baum (engl. Binary tree) ist entweder leer oder besteht aus einem Knoten (engl. Node), dem ein Element und zwei binäre Bäume zugeordnet sind.

**Methoden:**

* Empty: Liefert true, wenn der Baum leer ist
* Value: Liefert Wurzelelement
* Left: Liefert den linken Teilbaum
* Right: Liefert den rechten Teilbaum

**Konzept:**

Ein Bild, das Diagramm, Reihe, Text, Plan enthält.

Automatisch generierte Beschreibung