Java Befehlsliste

Konsolen Programmierung Befehlsliste

Befehl	Aufgabe
System.out.println("Hello World");	Textausgabe auf der Konsole
int ersteZahl;	Erstellt int Variable mit dem Namen ersteZahl
double ergebnis;	Erstellt double Variable mit dem Namen ergebnis
char einZeichen;	Erstellt char Variable mit dem Namen einZeichen
String vieleZeichen;	Erstellt String Variable mit dem Namen einZeichen
boolean Gesetzt;	Erstellt Boolean Variable mit Namen Gesetzt.
double Kommazahl = (int)intZahl;	Wandelt inZahl in Double Zahl um.
char Zeichen = Stringvariable.charAt(0);	Speichert ersten Buchstaben vom String in der Variablen char Zeichen.
Stringvariable.length();	Zeigt die Länge des String an
ergebnis=zahl1+zahl2;	Addiert zahl1, 2 und speichert es in der Variablen ergebnis
BufferedReader input=new BufferedReader (new InputStreamReader (System.in));	Muss ihr in euer Programm einfügen damit ihr Usereingaben einlesen könnt.
strInput = input.readLine();	Damit wird eine Usereingabe eingelesen und in der Variablen strInput gespeichert.
try { // Die Befehlskette welche ausgeführt werden soll. } *catch (Exception ex) { // Wenn irgend etwas schief geht wird diese Operation ausgeführt. }*	Hier wird der try Block ausgeführt wenn das Programm normal läuft. Wenn es in einen Fehler auftritt wird der Inhalt des catch Bereichs ausgeführt.
if (Bedingung) *{ Anweisungsblock 1; }* *else* *{ Anweisungsblock 2; }*	Mit so einer if Abfrage kann man z.B. prüfen ob der Inhalt einer Variablen kleiner, größer, gleichgroß usw. ist wie eine bestimmte Zahl. Wenn das zutrifft wird der Anweisungsblock 1 ausgeführt wenn nicht der Anweisungsblock 2.
while (Bedingung) *{* *Anweisung 1;* *Anweisung 2;* *}*	Die While Schleife wird so lange ausgeführt wie die Bedingung zutrifft. Während dieser Zeit werden die Anweisungen in der Schleife abgearbeitet. Dies ist eine Kopfgesteuerte Schleife da die Bedingung am Schleifenanfang also im Kopf steht. Es wird zuerst die Bedingung überprüft wenn diese zu trifft wird der Code in der Schleife ausgeführt.
do { *Anweisung 1;* *Anweisung 2;* *} while (Bedingung);*	Bei einer do while Schleife werden die Anweisungen auch so lange ausgeführt wie die Bedingung zutrifft. Dies ist aber eine Fußgesteuerte Schleife da die Bedingung sich am Ende befindet. Der Programmcode der Schleife wird mindestens einmal ausgeführt da die Bedingung erst am Ende der Schleife steht.
echteZahlint = Integer.parseInt(zahlAlsString);	String Variable in eine int Variable umwandeln.
echteZahldouble = Double.parseDouble(zahlAlsString);	String Variable in eine double Variable umwandeln.

String zahlAlsString = String.valueOf(eineZahl); *for (int i=1; i<=5; i++)* *{* *System.out.println ("Hello World");* *}*	Damit kann man verschiedene Typen in einen String umwandeln. Eine for Schleife wird auch so lange ausgeführt wie die Bedingungen zutreffen. Bei dem Beispiel wird die Laufvariable i bei jedem Schleifendurchlauf um die Zahl 1 erhöht. Damit kann man dann einstellen wie oft eine Anweisung ausgeführt wird.
switch (Testvairable){ *case 1: Anweisung 1;* *break;* *case 2: Anweisung 2;* *break;* default: Anweisungdefault; }	Es wird geprüft ob der Wert in der Variablen Testvariable 1 entspricht. Wenn das so ist wird Anweisung 1 ausgeführt, wenn der Wert 2 ist wird Anweisung 2 ausgeführt. Wenn etwas anderes in der Vairablen steht wird Anweisungdefault ausgeführt.
<Datentyp>[] <variablenname> = new <Datentyp> [<anzahl>] ; *int[] artikelnummern = new int [100] ;*	Ein Array wird benötigt wenn man mehre Variablen vom gleichen Datentyp benötigt. Hier werden z.B. 100 Variablen vom Datentyp int erstellt.
artikelnummern[14]=4;	Hier wird in das 15. Element vom Array die Zahl 4 gespeichert. Bei einem Array muss man beachten dass das erste Element mit der Zahl 0 beginnt.
import java.util.ArrayList;	Import welcher für die Arraylist benötigt wird
ArrayList testlist = new ArrayList();	Anlegen einer neuen Arraylist mit dem Namen test.
ArrayList<String> testlist = new ArrayList<String>();	Anlegen einer neuen Arraylist mit dem Namen Test vom Typ String.
testlist.add("Hallo");	Befüllt Arraylist test mit dem Wort Hallo
testlist.remove("Hallo");	Entfernt das Hallo wieder aus der Arraylist.
testlist.remove(2);	Entfernt den Eintrag an der 3 Stelle aus dem Array.
testlist.contains ("Hallo");	Prüft ob “Hallo” in der Liste enthalten ist.
testlist.get(2);	Gibt den Eintrag an Stelle 3 der Arraylist aus.
testlist.size();	Gibt die Anzahl der Elemente des Arrays aus.
String[] splittArray = Variable.split("\\;");	Splittet String bei einem ; und speichert in Array.
Variable.replaceAll("ü", "ue");	Sucht in einem String nach ü ersetzt durch ue.
BufferedReader inFile = new BufferedReader (new FileReader ("D:\\Logfile.txt")	Muss man im Programmcode einfügen damit man eine Datei auslesen kann. Hier z.B. die Datei Logfile.txt
aktline = inFile.readLine();	Hiermit wird eine Zeile aus einer Datei ausgelesen und in der Variablen aktline gespeichert.
inFile.close();	Damit wird die Datei welche man ausließt wieder geschlossen.
Test.startsWith ("Hallo");	Damit kann geprüft werden ob die Varibale Test mit dem Wort Hallo beginnt.
BufferedWriter inFile2 = new BufferedWriter (new FileWriter ("D:\\Logfile2.txt"));	Dieser Codeteil wird benötigt wenn man etwas in eine Textdatei schreiben möchte. Hier wäre es z.B. die Datei Logfile2.txt in welche etwas geschrieben wird.
inFile2.write(Text);	Damit wird der Inhalt der Variablen Text in der vordefinierten Datei gespeichert.

GUI Programmierung Befehlsliste

Befehl	Aufgabe
Test = jTextfeld.getText();	Eingabe aus dem Textfeld jTextfeld einlesen und in der Variablen Test Speichern.
jLabel.setText (Test);	Inhalt der Variablen Test auf dem Label jLabel ausgeben.
JOptionPane.showMessageDialog(null, "Test Messagebox", "Test Titel", JOptionPane.OK_CANCEL_OPTION);	Messagebox erstellen welche den Titel “Test Titel” hat. In der Textbox wird Test Messagebox angezeigt. Außerdem hat die Box einen OK und einen X Button
btnHdd.setEnabled(true);	Button einblenden / ausblenden true/false
lblHdd.setVisible(false);	Label einblenden / ausgrauen true/false
public double quadrat(double zahl) *{* *double ergebnis = zahl * zahl;* *return ergebnis;* *}*	Damit erstellt man eine Methode mit dem Namen quadrat. Eine Methode ist dafür da wenn man eine bestimmte Funktion öfters in einem Programm benötigt. Denn so kann man einfach die gewünschte Methode aufrufen und es wird der ganzen Code darin abgearbeitet.
quadratzahl= this.quadrat(2);	Methode wird aufgerufen und als Parameter die Zahl 2 übergeben. Der Rückgabewert der Methode wird in der Variablen quadratzahl gespeichert.
CKlasse Verwaltung = new CKlasse(200);	Erstellt ein Objekt mit dem Namen Verwaltung vom Typ CKlasse und übergibt den Wert 200.

Styleguide

Art	Schreibweiße	Beispiel
Klassennamen	Groß schreiben (evtl. mit „C“ davor)	CPC
Variablen	Klein Schreiben	int hdd
Button	btnName	btnHdd
Label	lblName	lblHdd
Textfeld	txtName	txtHdd
Textareea	taName	taHdd
Panel	paName	paHdd