Prüfungsschwerpunkte
(siehe auch Musterklausur einschließlich angefügter Befehlsliste)

Theoriefragen

Es ist nicht ausreichend die Fragestellungen nur verbal beantworten zu können, sondern man muß sie auch im Kontext mit kleinen Problemstellungen beantworten können.
(VERSTEHEN ! - NICHT AUSWENDIG LERNEN! ☺).

K0.1, K0.2

- Was versteht man allgemein unter einem Algorithmus und einem Programm?
- Wozu dient das Dual-, Oktal- und Hexadezimalsystem in der Informatik?
- Was versteht man unter einem Bit?
- Was ist der ASCII-Code?
- Was ist ein Datentyp und worin unterscheiden sich die Datentypen prinzipiell?
- Nennen und charakterisieren Sie mind. vier Grunddatentypen (einfache Datentypen).
- Was ist eine digitalisierte Information und wie gewinnt man Sie?

K1.1

- Wie unterscheiden sich compilerbasierte und interpreterbasierte Programmentwicklung?

K1.2, K2.4, K2.5

- Was bedeutet typgebundene und nicht-typgebundene (untypisierte) Programmierung?
- Was bedeutet Typkonvertierung und was ist dabei zu beachten?
- Wofür werden Datentypen in nicht-typgebundenen Sprachen wie MATLAB vorrangig eingesetzt?
- Was sind Verbund-Datentypen?

K1.4

- Was versteht man unter einer logischen Variablen bzw. einem Wahrheitswert?

K1.7

- Was versteht man unter einem syntaktischen und was unter einem semantischen Fehler?

K1.8,1.9

- Was sind wesentliche Unterschiede zwischen einem M-Skript und einer M-Funktion?
- Beschreiben Sie den Unterschied zwischen einer lokalen und einer globalen Variablen.

K1.10, K1.11

- Was unterscheidet eine skalare Variable von einer Vektorvariablen bzw. Matrixvariablen?

Anwendungsfähiges Wissen mit MATLAB

K1.2

- Regeln zur Namensgebung von Variablen
- Wertzuweisung an Variablen
- Kenntnis wesentlicher vordefinierter Variablen wie z.B. `pi, inf, NaN`
- Sichere Anwendung (Operator-Prioritäten, Auswertungsreihenfolge) der arithmetischen Operatoren und grundlegender mathematischer Funktionen wie round, fix, abs, sin, log, sqrt … in skalaren Ausdrücken (z.B. Formeln)
- Sichere Interpretation von Wokspace-Anzeigen mit dem Befehl `whos`

=> Mathematische Formeln mit skalaren Größen in einen MATLAB-Ausdruck überführen und umgekehrt.

K1.3

- Programmierung eigener kleiner Skripte mit interaktiver Eingabe (`input`) und Ausgabe (`disp`)
- Suche von Fehlern in vorgegebenen Skripten

K1.4

- Sichere Anwendung (Operator-Prioritäten, Auswertungsreihenfolge) der Vergleichsoperatoren und der logischen Operatoren & (AND), | (OR), ~ (NOT) in rein logischen Ausdrücken und in Kombination mit arithmetischen Ausdrücken

=> Rein logische und kombiniert logisch-arithmetische Ausdrücke aufstellen und a uswerten können.

K1.5, K1.6

- Sichere Beherrschung der Kontrollstrukturen
  - `if, elseif, else, end`
  - `for Schleife`
  - `while Schleife`
- Umsetzung von textuell beschriebenen oder in Form von Pseudocode oder Flussdiagrammen gegebenen Problemen mit den MATLAB-Kontrollstrukturen
- Analyse vorgegebener Kontrollstrukturen einschließlich Fehlersuche

K1.7

- Suche von syntaktischen und semantischen Fehlern in vorgegebenen Programmstücken

K1.8, 1.9

- Sichere Beherrschung des Konzepts von Funktionen!
  - Implementieren von Funktionen mit mehreren Eingangsparametern und Rückgabeparametern
  - Funktionsaufrufe; Übergabe und Rückgabe von Parametern
  - Lokale Variablen in Funktionen
  - Globale Variablen
- Eigenes Implementieren „kleiner" Funktionen
- Suche von Fehlern in vorgegebenen Funktionen bzw. erweitern vorgegebener Funktionsrümpfe
- ENTFÄLLT: Abfrage von Parameterlisten mit naegin, nargout, subfunctions

K1.10, K1.11

- Erzeugung (Definition) von Zeilenvektoren, Spaltenvektoren und Matrizen mit numerischen Werten bzw. mit ASCII-Zeichen
- Initialisieren von Vektoren und Matrizen mit `zeros` und `ones`
- Abfrage der Dimension von Vektoren mit `length` und von Matrizen mit `size`
- Zugriff auf Vektor- und Matrixelemente mit dem index-Operator (Klammeroperator)
- Verketten (horizontal und vertikal) von Vektoren und Matrizen
- Generierung einfacher Linienplots mit `plot` (ohne Beschriftungen)
- Programmierung mit Vektoren und Matrizen in Verbindung mit Kontrollstrukturen (gemäß dem Beispiel Schachbrettmuster in Kap.1.10 bzw. der Laboraufgabe „Transponieren einer Matrix" mit for-Schleifen
- Einlesen von Daten aus einer ASCII-Datei in einen Vektor bzw. eine Matrix und schreiben von Daten in eine ASCII-Datei (`load, save`)

=> Programmierung eigener M-Skripte und M-Funktionen mit skalaren Variablen, Vektoren und Matrizen unter Verwendung von arithmetischen und logischen Operatoren sowie Kontrollstrukturen. Suche von Fehlern in vorgegebenen Funktionen und Erweiterung von vorgegebenen Funktionen.

K2.1, K2.2

- Sichere Anwendung der auf Vektoren und Matrizen erweiterten arithmetischen und logischen Operatoren sowie mathematischen Funktionen (Prioritäten, Auswertungsreihenfolge).
  => Mathematische Formeln mit skalaren und vektoriellen Größen sowie Matrizen in einen MATLAB-Ausdruck überführen und umgekehrt.
- Sichere Anwendung der speziell für Vektoren und Matrizen eingeführten Operatoren: Transponierung ' , Indexierung ganzer Zeilen bzw. Spalten dem Colon-Operator (Doppelpunktoperator), Indexierung letzter Elemente mit `end`, Indexierung mehrerer Vektor- und Matrixelemente durch Vektoren als Index wie z.B. a([2,3]), a([2:2:end])
- Zusammenbauen und Zerlegen von Matrizen, auch im Kontext mit den Einlesen und Speichern von Daten aus/in ASCII-Dateien

=> Anwendung der erweiterten Matrixoperationen in kleinen Programmen.

K2.4, K2.5

- Beherrschung einfacher Typkonvertierungen in MATLAB wie z.B. double->int, double->int8, double->char
- Erzeugung einer Struktur mit `struct` bzw. dem Punktoperator (Punktnotation) mit verschiedenen Feldelementen von unterschiedlichem Datentyp
- Lesen und modifizieren der Inhalte von Struktur-Feldelementen mit der Punktnotation
- Vektoren mit Strukturvariablen des gleichen Datentyps (Vorlesungsbsp. Telefonbuch)