

**Algorithmen und Datenstrukturen**  
**Versuch 4**

**VORBEREITUNGSAUFGABEN**

- Beantworten Sie vorab die theoretische Frage.
- Erstellen Sie das Struktogramm (Nassi-Shneiderman) für das Hauptprogramm der Aufgabe 4.

**AUFGABENSTELLUNG**

**Mondlandung simuliert die Landung einer Landefähre auf dem Mond. Aufgabe des Spielers ist die Steuerung des durch die Schubdüsen erzeugten Schubs. Dabei kann zwischen 10 verschiedenen Schubstufen (0-9) ausgewählt werden und ist jeweils für eine Sekunde gültig. Das Spiel ist gewonnen, wenn die Landefähre mit einer Geschwindigkeit unter 1 m/s aufsetzt.**

**Implementieren Sie das Spiel in ANSI-C unter Berücksichtigung folgender Hinweise:**

- Initialisieren Sie das Spiel mit einer Anfangshöhe von 50 m und einer Anfangsgeschwindigkeit von -10 m/s (fallend).
- Die wiederholte Abfrage des Spieler nach der Schubstufe erfolgt über `scanf()`. Überprüfen Sie, ob die Eingabe der ganzen Zahl zwischen 0 und 9 liegt.
- Aus der Schubstufe ist der Schub in N zu ermitteln. Dazu multiplizieren Sie die Stufe mit einem Schubfaktor von 500 N, so dass ein Schub zwischen 0 N und 4500 N kommandiert werden kann.
- Berechnen Sie die sich durch den Schub ergebende Beschleunigung  $a$  der Fähre mittels der Formel  $F = ma$ . Die Masse der Landefähre betrage 1000 kg. Ziehen Sie von dieser Beschleunigung noch die Fallbeschleunigung des Mondes ( $1.6 \text{ m/s}^2$ ) ab.
- Ermitteln Sie aus der Beschleunigung  $a$  die aktuelle Geschwindigkeit  $v$  und Höhe  $h$ . Wegen

$$\dot{v} = a \text{ und } \dot{h} = v$$

können wir als Integrationsformeln

$$v_{\text{neu}} = v_{\text{alt}} + a dt \text{ und } h_{\text{neu}} = h_{\text{alt}} + v_{\text{neu}} dt$$

nutzen. Dabei ist  $dt$  gerade 1s.

- Sollte die Höhe größer als 100 m werden, dann ist das Programm mit einer geeigneten Meldung zu beenden.
- Die Landefähre gilt als aufgesetzt, wenn sie einen Abstand von weniger als 10 cm zum Boden hat. In diesem Fall überprüfen Sie, ob die Fähre sanft gelandet oder gecrasht ist und geben eine entsprechende Erfolgs- oder Katastrophenmeldung (inkl. letzten Schub, Höhe und Geschwindigkeit) auf dem Bildschirm aus.
- Folgender Screenshot zeigt einen exemplarischen Programmablauf:

```
user@user-vm: ~/versuch4
user@user-vm:~/versuch4$ ./versuch4
Mondlandung
Bitte geben Sie jeweils eine Schubstufe von 0-9 an:
Schub: 0 -> akt. Hoehe: 50.0 m, Geschwindigkeit: -10.0 m/s: 4
Schub: 4 -> akt. Hoehe: 40.4 m, Geschwindigkeit: -9.6 m/s:
Schub:  -> akt. Hoehe:  m, Geschwindigkeit:  m/s:
Schub:  -> akt. Hoehe:  m, Geschwindigkeit:  m/s:
Schub:  -> akt. Hoehe:  m, Geschwindigkeit:  m/s:
Schub:  -> akt. Hoehe:  m, Geschwindigkeit:  m/s:
Schub:  -> akt. Hoehe:  m, Geschwindigkeit:  m/s:
Schub:  -> akt. Hoehe:  m, Geschwindigkeit:  m/s:
Schub:  -> akt. Hoehe:  m, Geschwindigkeit:  m/s:
Schub:  -> akt. Hoehe:  m, Geschwindigkeit:  m/s:
Schub:  -> akt. Hoehe: 0.0 m, Geschwindigkeit: -0.5 m/s ->
Gewonnen!
user@user-vm:~/versuch4$
```

**Algorithmen und Datenstrukturen**  
**Versuch 4**

**THEORETISCHE FRAGE**

Welche Fehlerfälle können bei der Eingabe des Schubs auftreten und wie sollten diese in Ihrem Programm sinnvollerweise behandelt werden?

**VERSUCHSDURCHFÜHRUNG**

- Melden Sie sich mit Ihrem Account an und öffnen Sie ein Terminal-Fenster.
- Erstellen Sie zuerst das Verzeichnis `versuch4` in Ihrem Heimatverzeichnis (`/home/Benutzername`).
- Verwenden Sie den Editor `gedit` zum Erstellen der Datei `versuch4.c` im eben erstellten Verzeichnis.
- Die im Programm vorkommenden Konstanten (Naturkonstanten und Initialisierungswerte) sind über Präprozessoranweisungen festzulegen.
- Verwenden Sie den Rückgabewert von `scanf()` zum Abfangen der Fehlerfälle.
- Falls Ihre Umsetzung (mit den Compiler-Schaltern "`-Wall -pedantic -ansi`") Warnungen oder Fehler erzeugt, korrigieren Sie Ihren Quellcode.
- Landen Sie Ihre Fähre sicher auf dem Mond.
- Schaffen Sie die Landung mit einer kürzeren Landungszeit (also mit weniger Steuerkommandos) als Ihre Kommilitonen?
- Lassen Sie sich das lauffähige Programm und Ihr Struktogramm von Ihrem Betreuer abnehmen.

**OPTIONAL**

- Das Spiel wird viel interessanter, wenn der zur Verfügung stehende Treibstoff (oder dazu gleichwertig → der zur Verfügung stehende Schub) begrenzt ist. Steht nicht mehr genug Treibstoff zur Verfügung, muss in diesem Fall das Programm den gewünschten Schub entsprechend reduzieren. Ermitteln Sie durch ein Experiment eine sinnvolle Treibstoffmenge.
- Realistischer (und schwieriger) wird die Situation, wenn sich die Masse der Fähre mit der Menge des ausgestoßenen Treibstoffs verringert. Wie könnte dies implementiert werden?