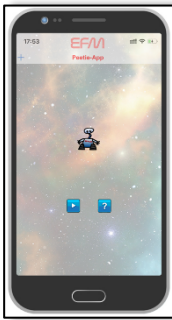


# ESCAPE FROM MARS: EINGABE/AUSGABE



## Mission 01: Reset P.T.98

Die Programmierer von General Spirin haben sich Zugriff auf **PEETIE** verschafft und ihn gehackt. Gleichzeitig stürmt ein SWAT-Team die Zentrale, um Dr. White festzunehmen. Doch in dieser schweren Stunde zeigt sich, dass er sich auf Dragan verlassen kann.

### Aufgabe

**EFM**  
escape-from-mars.de

**Job für Dr. White**

Peetie muss mit neuen Werten versorgt werden. Das bedeutet, dass seine Variablen neu deklariert und initialisiert werden müssen. Dazu verwenden wir Werte, die vom User eingegeben werden.  
Implementiere das Struktogramm bei Peetie!

**NEUE BEFEHLE:**  
textAufDisplay(text) - Gibt Text auf Smartphone aus  
variable = textEinlesen() - Liest Zeichenkette ein und weist sie der Variablen zu  
variable = ganzzahlEinlesen() - Ganzzahl einlesen  
variable = kommazahlEinlesen() - Dezimalzahl einlesen

```
Peetie neu initialisieren -- EG M01
// Anmerkung: In Java verwenden wir für ...
// - Zeichenkette --> String
// - Kommazahl --> double
// - Ganzzahl --> int
// Achtung: Großschreibung bei String!

// Einlesen: variable = textEinlesen(), oder
// variable = ganzzahlEinlesen(), oder
// variable = kommazahlEinlesen()
// Ausgabe: textAufDisplay("Du bist " + alter + " Jahre alt.")

Deklaration: name als Zeichenkette
Deklaration: modellbezeichnung als Zeichenkette
Deklaration: gewicht als Kommazahl
Deklaration: gewichtTriebwerk als Kommazahl
Deklaration: anzahlRaketen als Ganzzahl
Ausgabe: "Namen eingeben: [Zeichenkette]"
Einlesen: name
Ausgabe: "Modellbezeichnung: [Zeichenkette]"
Einlesen: modellbezeichnung
Ausgabe: "Gewicht eingeben: [Kommazahl]"
Einlesen: gewicht
Ausgabe: "Triebwerksgewicht? [Kommazahl]"
Einlesen: gewichtTriebwerk
Ausgabe: "Anzahl Raketen? [Ganzzahl]"
Einlesen: anzahlRaketen
Ausgabe: "--- IHR INPUT: ---"
Ausgabe: "Name: " + name
Ausgabe: "Modellbezeichnung: " + modellbezeichnung
Ausgabe: "Gewicht: " + gewicht
Ausgabe: "Triebwerksgewicht: " + gewichtTriebwerk
Ausgabe: "Anzahl Raketen: " + anzahlRaketen
```

## Mission 2: Die Peetie-App

Endlich - **PEETIE** ist startklar! Doch es müssen einige letzte Berechnungen vorgenommen werden: Ist **PEETIE** nicht zu schwer? Wird er wie ein Stein auf die Marsoberfläche zurückfallen? Zum Glück kommt Dragan die zündende Idee, wie man Gewicht einsparen kann.



## Aufgabe

P

### Job für Dragan

1. **Gesamtgewicht Peetie** ausrechnen (gewichtPeetie + gewichtTriebwerk) und ausgeben. Dazu neue Variable gesamtgewichtPeetie deklarieren und verwenden.
2. Anzahl Raketen einlesen
3. **Gesamtgewicht Raketen** ausrechnen (gewichtRakete \* anzahlRaketen) - neue Variable verwenden (z.B. gewichtAlleRaketen)
4. **Gesamtgewicht Peetie ohne Raketen** ausrechnen - neue Variable verwenden (z.B. gesamtgewichtPeetieOhneRaketen)
5. Alle Ergebnisse ausgeben