```
> restart:
```

Aufgabe 4

Modul-Programmierung

Maple's factor Kommando faktorisiert Polynome, ganzzahlige Koeffizienten dagegen nicht:

```
> factor(6*x^2+6*x-12);
```

$$6\,(x+2)\,(x-1) \tag{1}$$

Für die Faktorisierung ganzer Zahlen (integer) wird das ifactor Kommando benötigt:

```
> ifactor(6);
```

$$(2)\,(3) \tag{2}$$

Beide Faktorisierungsschritte lassen sich mittels map und einer geeigneten anonymen Funktion zusammenfassen:

```
> map(t->if t::integer then ifactor(t) else t end if, factor(6*
  x^2+6*x-12));
```

$$(2)\,(3)\,(x+2)\,(x-1) \tag{3}$$

Überschreiben Sie Maple's `factor` Kommando so, das Polynome mit ganzzahligen Koeffizienten vollständig faktorisiert werden. Verwenden Sie die Modul-Programmiertechnik.

Hinweis: Definieren Sie eine neue Prozedur `factor`, die über ein Modul exportiert wird (module,export). Der Zugriff auf die "globale" `factor`-Funktion (außerhalb des Moduls) erfolgt mit dem :- Operator (colondash).

```
> module_factor := module()
    export factor:
    option package:
    factor := proc(func)
      map(t->if t::integer then ifactor(t) else t end if, :-factor
  (func));
    end proc:
  end module:
> with(module_factor):
  factor(6*x^2+6*x-12);
```

$$(2)\,(3)\,(x+2)\,(x-1) \tag{4}$$