6. Programmieraufgabe Computerorientierte Mathematik II

Abgabe: 4.6.2021 über den ComaJudge bis 17 Uhr

Aufgabe

Auf ISIS steht zum Download eine Datei `intVektor` bereit. In dieser finden Sie die Klasse `IntVektor`, welche die Operationen der Vektorklasse aus der Übung auf der Menge $\mathbb{Z}^3$ realisiert. Es seien nun zwei Vektoren $b_1, b_2 \in \mathbb{Z}^3$ wiefolgt gegeben:

$$b_1 := \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} \qquad b_2 := \begin{pmatrix} 1 \\ 0 \\ 5 \end{pmatrix} .$$

Mittels dieser Definieren wir die Teilmenge $G \subset \mathbb{Z}^3$:

$$G := \{k_1 \cdot b_1 + k_2 \cdot b_2 : k_1, k_2 \in \mathbb{Z}\} .$$

G ist ein *Teilgitter* von $\mathbb{Z}^3$. Schreiben Sie eine Klasse `Teilgitter`, die `IntVektor` erbt. `Teilgitter` hat gegenüber `IntVektor` die zusätzlichen Attribute `Koordinate_1` und `Koordinate_2`, welche die Koordinaten des gegebenen Vektors bezüglich der geordneten Basis $\{b_1, b_2\}$ von G darstellen. Die Klasse `Teilgitter` soll folgende Methoden bereitstellen:

- `__init__(self, x, y, z)` Wirft den Fehler `"Vektor liegt nicht im Teilgitter."` falls der durch `x,y,z` gegebene ganzzahlige Vektor nicht in G liegt. Ansonsten werden die Attribute `x,y,z` mit den übergebenen Parametern initialisiert. `Koordinate_1` und `Koordinate_2` werden mit den Koordinaten des gegebenen Vektors bezüglich der geordneten Basis $\{b_1, b_2\}$ initialisiert.
  Hinweis: Überlegen Sie sich, warum es bei der speziellen Eintragsstruktur von b_1 und b_2 einfach ist, die Zugehörigkeit zu G prüfen und gegebenenfalls die Koordinaten zur Basis $\{b_1, b_2\}$ zu berechnen.

- `__str__(self)` Gibt die Informationen zur Klasseninstanz in folgendem Format aus:

  ```
  (x,y,z); Koordinate 1: Koordinate_1, Koordinate 2: Koordinate_2
  ```

- `__add__(self,other)` Wie bei `IntVektor`, lediglich der Rückgabewert ist ein Objekt von Typ `Teilgitter`.

- `__mul__(self,other)` Wie bei `IntVektor`, lediglich der Rückgabewert der skalaren Multiplikation ist ein Objekt von Typ `Teilgitter`.

- `__rmul__(self,other)` Siehe Angaben zu `__mul__(self,other)`.

- `copy(self)` Siehe Angaben zu `__add__(self,other)`.

Wichtige Hinweise

- Bitte kopieren Sie die Klasse `IntVektor` in Ihre Abgabedatei, der Comajudge kann sonst Ihren Code nicht interpretieren.
- Programme, bei denen alle Funktionen komplett neu geschrieben wurden, ohne Funktionalitäten der ererbten Klasse zu nutzen, werden bei der Abnahme nicht akzeptiert.

Beispielaufrufe

```
>>> A = Teilgitter(10, 3, 23)
>>> print(A)
(10,3,23); Koordinate 1: 3, Koordinate 2: 4
>>> B = Teilgitter(14,4,34)
>>> print(B)
(14,4,34); Koordinate 1: 4, Koordinate 2: 6
>>> print(A+B)
(24,7,57); Koordinate 1: 7, Koordinate 2: 10
>>> print(3*A)
(30,9,69); Koordinate 1: 9, Koordinate 2: 12
>>> print(-3*A)
(-30,-9,-69); Koordinate 1: -9, Koordinate 2: -12
>>> print(B*7)
(98,28,238); Koordinate 1: 28, Koordinate 2: 42
>>> print(A*B)
934
>>> print(A.copy())
(10,3,23); Koordinate 1: 3, Koordinate 2: 4
>>> print(Teilgitter(9,5,0))
(9,5,0); Koordinate 1: 5, Koordinate 2: -1
```