Aufgabenblatt 6

Aufgabe 1)

a)

Erstelle eine Klasse Student. Ein Student erhält bei seiner Erstellung einen Namen, Geschlecht, Matrikelnummer und ein Semester.

Ein Student hat eine Funktion, welche alle personenbezogenen Informationen ausgibt. Erstellen Sie drei unterschiedliche Studenten und lassen Sie deren Informationen auf der Konsole ausgeben.

Bsp.

```
Name: Timon Hellerich, Geschlecht: maennlich, MatNr: 123456, Semester: 3
Name: Maria Mustermann, Geschlecht: weiblich, MatNr: 123423, Semester: 3
Name: Max Mustermann, Geschlecht: maennlich, MatNr: 800331, Semester: 1
```

b)

Die Matrikelnummer muss genau 6 Ziffern betragen. Wenn dies nicht der Fall ist, soll eine Warnung ausgegeben werden und nach einer erneuten Eingabe gefragt werden. Wenn diese erneut falsch ist wird eine Exception geworfen.

c)

Erstellen Sie einen zweiten Konstruktor, welcher für die Matrikelnummer automatisch eine zufällige Zahl vergibt. Zudem soll in diesem die Semesterzahl automatisch auf 1 gesetzt werden.

d)

Erstellen Sie eine neue Klasse namens Studiengang. Diese Klasse erhält eine Bezeichnung und eine beliebige Anzahl an Studenten. Erstellen Sie eine Funktion, welche die Infos aller Studenten ausgibt.

Hinweis:

Speichern Sie alle Studenten in einem Studentenarray. Für den Übergabeparameter benötigen sie params.

e)

Studenten können exmatrikuliert werden, wenn Sie ihr Studium abschließen. Erstellen Sie für diesen Vorgang eine Funktion, welche die Matrikelnummer empfängt und den Studenten löscht. An den alten Platz im Array wird der Wert null gespeichert.

Bsp:

```
Studenten im Studiengang Informatik:
Name: Timon Hellerich, Geschlecht: maennlich, MatNr: 123456, Semester: 3
Name: Maria Mustermann, Geschlecht: weiblich, MatNr: 123423, Semester: 3
Name: Max Mustermann, Geschlecht: maennlich, MatNr: 895512, Semester: 1
123456 wird entfernt...
Studenten im Studiengang Informatik:
Name: Maria Mustermann, Geschlecht: weiblich, MatNr: 123423, Semester: 3
Name: Max Mustermann, Geschlecht: maennlich, MatNr: 895512, Semester: 1
```

f)

Wenden Sie diese Funktion in einer neuen Funktion „neuesSemester()“ an in dieser wird die Zahl des Semesters von jedem Studenten um 1 erhöht. Wenn die Zahl des Semesters größer als 9 ist wird der Student exmatrikuliert.

```
Studenten im Studiengang Informatik:
Name: Timon Hellerich, Geschlecht: maennlich, MatNr: 123456, Semester: 3
Name: Maria Mustermann, Geschlecht: weiblich, MatNr: 123423, Semester: 9
Name: Max Mustermann, Geschlecht: maennlich, MatNr: 997746, Semester: 1
123423 wird entfernt...
Studenten im Studiengang Informatik:
Name: Timon Hellerich, Geschlecht: maennlich, MatNr: 123456, Semester: 4
Name: Max Mustermann, Geschlecht: maennlich, MatNr: 997746, Semester: 2
```

Aufgabe 9.1. Klasse Vector (basierend auf Praktikumsaufgabe 6/2022)

Definieren Sie eine Klasse Vector für das Rechnen mit Vektoren. Ein Vektor soll intern durch ein privates, n-dimensionales Double-Array repräsentiert werden.

Hier zunächst das Hauptprogramm mit Testfällen, dass genau so ausgeführt werden können soll, wie es hier abgedruckt ist. Übernehmen Sie den Code aus Moodle oder hier aus dem PDF.

```
// Beispielvektoren definieren
Vector v = new Vector(5, 4, -8, 2, 6);
Vector w = new Vector(3, -4, -1, 5, -2);

// Vektoren ausgeben
Console.WriteLine($"v = {v}");
Console.WriteLine($"v ist {v.GetDim()}-dimensional.");
Console.WriteLine($"w = {w}");
Console.WriteLine();

// Norm berechnen
Console.WriteLine("L2-Norm(v) = " + v.Norm());
Console.WriteLine();

// Summe
Console.WriteLine("v + w = " + v.Add(w));
Console.WriteLine();

// Normalisierung des Vektors
v.Normalize();
Console.WriteLine("v_normalisiert = " + v);
Console.WriteLine("L2-Norm(v_normalisiert) = " + v.Norm());
```

Implementieren Sie folgende **öffentliche, nicht-statische** Funktionen:

a) Konstruktoren: Erstellen Sie zwei öffentliche Konstruktoren.
Der erste Konstruktor soll zwei Parameter erhalten: Einen Integer-Parameter für die Dimensionalität n und einen optionalen Wert für die Initialisierung des Arrays (Standard 0).
Der zweite Konstruktor soll beliebig viele Double-Parameter erhalten und daraus ein Vector-Objekt generieren.

b) ToString(): Erzeugt einen String in der Form "(5, 4, -8, 2, 6)" erzeugt und ihn zurück.

c) GetDim(): Soll die Dimensionalität des Vektors zurückgeben.

d) Add(): Addiert zwei Vektoren. Was wird zurückgegeben?

e) Norm(): Berechnet die Euklidische Norm des Vektors: $\|v\|_2 = \sqrt{\sum_{i=1}^{n} v_i^2}$

f) Normalize(): Normalisiert den Vektor, d.h. ändert ihn auf $v' = \frac{v}{\|v\|_2}$.

g) Optional: DotProduct(): Berechnet das Skalarprodukt $v \circ w = \sum_{i=1}^{n} v_i \cdot w_i$.