

Uitleg over de Student Klasse in TypeScript

Deze TypeScript-code definieert een Student klasse met eigenschappen en methoden om studentgegevens te beheren,

waaronder de naam, het studentnummer en de cijfers. Hieronder leg ik het stap voor stap uit.

1. DE STUDENT KLASSE

Eigenschappen (velden):

```
private _naam: string;
```

```
private _studentnummer: string;
```

```
private _cijfers: number[] = [];
```

- `_naam` en `_studentnummer`: Privé eigenschappen die de naam en het studentnummer van de student opslaan.

- `_cijfers`: Een array waarin de cijfers van de student worden opgeslagen.

Door deze eigenschappen `private` te maken, kunnen ze alleen binnen de klasse worden gebruikt en niet direct buiten

de klasse worden gewijzigd.

Constructor:

```
constructor(naam: string, studentnummer: string) {
```

```
    this._naam = naam;
```

```
    this._studentnummer = studentnummer;
```

```
}
```

- De constructor zorgt ervoor dat wanneer een nieuw Student object wordt aangemaakt, de naam en het studentnummer worden ingesteld.

Voorbeeld:

```
const student1 = new Student("Pieter", "12345");
```

Dit maakt een Student object met naam "Pieter" en studentnummer "12345".

2. GETTERS EN SETTERS

Getter: getNaam

```
public get getNaam(): string {  
    return this._naam;  
}
```

- Dit is een getter methode die de naam van de student ophaalt.

Getter: getGemiddelde

```
public get getGemiddelde(): number {  
    if (this._cijfers.length === 0) return 0;  
    const totaal = this._cijfers.reduce((acc, curr) => acc + curr, 0);  
    return Number((totaal / this._cijfers.length).toFixed(2));  
}
```

- Berekent het gemiddelde van de cijfers.

Setter: setNaam

```
public set setNaam(value: string) {  
  if (value.trim() !== "") {  
    this._naam = value;  
  } else {  
    console.log("De naam mag niet leeg zijn.");  
  }  
}
```

- Dit is een setter methode waarmee de naam van de student gewijzigd kan worden.

3. METHODEN

Methode: voegCijferToe

```
public voegCijferToe(cijfer: number): void {  
  this._cijfers.push(cijfer);  
}
```

- Deze methode voegt een cijfer toe aan de _cijfers array.

4. STUDENT OBJECTEN MAKEN EN GEBRUIKEN

```
const student1 = new Student("Pieter", "12345");  
const student2 = new Student("Anna", "54321");  
const student3 = new Student("Sofie", "67890");
```

We voegen cijfers toe:

```
student1.voegCijferToe(8);
```

```
student1.voegCijferToe(7.5);
```

```
student1.voegCijferToe(6);
```

```
student2.voegCijferToe(7);
```

```
student2.voegCijferToe(7.5);
```

```
student2.voegCijferToe(8);
```

5. RESULTATEN

```
console.log(`${student1.getNaam}'s gemiddelde is ${student1.getGemiddelde}`);
```

```
// Pieter's gemiddelde is 7.17
```

```
console.log(`${student2.getNaam}'s gemiddelde is ${student2.getGemiddelde}`);
```

```
// Anna's gemiddelde is 7.5
```

```
console.log(`${student3.getNaam}'s gemiddelde is ${student3.getGemiddelde}`);
```

```
// Sofie's gemiddelde is 0 (want Sofie heeft nog geen cijfers)
```

6. NAAM WIJZIGEN

```
student1.setNaam = "Peter";
```

```
console.log(`De nieuwe naam van student1 is: ${student1.getNaam}`);
```

```
// De nieuwe naam van student1 is: Peter
```

SAMENVATTING

- Encapsulatie: De gegevens (`_naam`, `_studentnummer`, `_cijfers`) zijn privé en kunnen niet direct worden aangepast.
- Getters & Setters: Toegang tot gegevens via `getNaam`, `getGemiddelde`, `setNaam`.
- Methoden: `voegCijferToe()` voegt cijfers toe en `getGemiddelde` berekent het gemiddelde.
- Gebruik van TypeScript: Type safety (`string`, `number[]`, `void`), moderne ES6+ methoden (`reduce`, `.toFixed(2)`).

Wil je nog iets verder verduidelijkt hebben?