

Oefeningen Javascript

(1) Vertrek van voorbeeld1 in webpack_babel_jest_skeleton.zip op Blackboard. Maak de klasse Line. Een Line bestaat uit 2 Point-objekten (point1 & point2).

Maak een constructor die 2 argumenten heeft. Als een van deze argumenten geen Point is wordt een Error opgeworpen Anders worden de argumenten toegekend aan point1 en point2.

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/instanceof>

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/throw>

Voorzie getters voor point1 en point2 en toString.

Maak gebruik van Jest om constructor, getters en toString te testen.

(2) Maak de klasse WereldObject in de map src/js/game. Deze klasse heeft eigenschap location van datatype Point (deze klasse kopieer je vanuit oefening1 naar game). Voorzie getX, setX, getY, setY in de klasse WereldObject.

Maak de klasse Hero. Hero is afgeleid van de klasse WereldObject. Maak de methode moveDown, moveUp, moveLeft, moveRight. Via deze methoden wordt de waarde van y verlaagd met 1, verhoogd met 1 of de waarde van x verlaagd met 1, verhoogd met 1.

Voorzie de unittests voor de methoden moveLeft, moveRight.

(3) Maak de klasse Datum in de map (src/js/accounting). Datum heeft eigenschappen day, month, year. Voorzie een constructor om een datum aan te maken, er worden waarden voor day, month en year meegegeven.

De datum moet in het formaat 1/1/2008 teruggegeven worden via de methode toString.

Voorzie een extra methode toStringMonth die de datum als volgt afdruckt: 1/jan/2008. Maak hiervoor gebruik van de static array MONTHS.

Voorzie setters en getters voor day, month en year.

Maak een static methode make om een datum aan te maken.

```
let date1=Datum .make(1,2,2001);
```

Voorzie de unittests voor toString en toStringMonth.

Hint: een static variabele voor de array MONTHS kan je maken via een static getter:

```
class Datum{
  constructor(day, month, year){
    this.day=day;
    this.month=month;
    this.year=year;
  }
  static get MONTHS() {
    return ['jan', 'feb', 'mar', 'apr', 'may', 'jun', 'jul', 'aug', 'sep', 'oct', 'nov', 'dec'];
  }

  getLongMonth(){
    return Date.MONTHS[this.month-1];
  }
}
```

```
let d= Datum Date(1,1,1977);
console.log(d.getLongMonth());
```

Je kan ook werken met een private static class field (ES2019):

```
class Datum{
  static #MONTHS=['jan', 'feb', 'mar', 'apr', 'may', 'jun', 'jul', 'aug', 'sep', 'oct', 'nov', 'dec'];
  #day;
  #month;
  #year
  constructor(day, month, year){
    this.#day=day;
    this.#month=month;
    this.#year=year;
  }

  getLongMonth(){
    return Datum.#MONTHS[this.#month-1];
  }
}
```

```
let d=new Datum(1,1,1977);
console.log(d.getLongMonth());
```

(4) De startcode kan je op Blackboard terugvinden.

In het spel regenwormen gooit een speler een aantal keren met een aantal zeszijdige dobbelstenen. De dobbelstenen hebben zijden '1', '2', '3', '4', '5' en 'R'. Zijden '1' tot '5' hebben numerieke waarde 1 tot 5. 'R' heeft numerieke waarde 5.



src/js/regenwormen/Dobbelsteen.js

Vul de klasse Dobbelsteen in de map regenwormen aan. De klasse Dobbelsteen heeft eigenschap _zijde. Binnen de constructor komt zijde binnen als argument. Er wordt gecontroleerd of zijde gelijk is aan '1', '2', '3', '4', '5' of 'R' indien niet wordt een Error opgeworpen.

Maak de methode getZijde die de eigenschap _zijde teruggeeft.

Maak de methode getNumeriekeWaarde die de numerieke waarde van de dobbelsteen teruggeeft. (_zijde '1' heeft waarde 1, _zijde '2' heeft waarde 2, _zijde '3' heeft waarde 3, _zijde '4' heeft waarde 4, _zijde '5' heeft waarde 5, _zijde 'R' heeft waarde 5)

test/js/regenwormen/Dobbelsteen.test.js

Maak een test waarmee gecontroleerd wordt dat de constructor een Dobbelsteen object maakt voor zijde '1'. Maak een test waarmee gecontroleerd wordt dat een Error opgeworpen wordt voor zijde 'S'.

Per worp maakt de speler een **selectie** van een aantal gelijke dobbelstenen. De speler kan er bijvoorbeeld voor kiezen om 2 dobbelstenen met zijde '3' te selecteren. De numerieke waarde van deze selectie is 6 (2 maal 3).

src/js/regenwormen/Selectie.js

De klasse Selectie heeft eigenschappen _aantal en _dobbelsteen.

De constructor krijgt een waarde voor aantal en dobbelsteen binnen. Als dobbelsteen geen object is van de klasse Dobbelsteen wordt een Error opgeworpen. aantal en dobbelsteen worden toegekend aan _aantal en _dobbelsteen.

De methode getAantal geeft _aantal terug.

De methode getDobbelsteen geeft _dobbelsteen terug.

De methode getNumeriekeWaarde geeft _aantal maal de numerieke waarde van _dobbelsteen terug.

Elke selectie wordt bij op de **stapel** gelegd. De speler mag elke soort dobbelstenen maar één keer selecteren. Dus als de speler in een worp dobbelstenen met zijde '1' geselecteerd heeft, mag hij in de volgende worpen de dobbelstenen met zijde '1' niet meer selecteren.

De numerieke waarde van de stapel is de som van de numerieke waarden van alle selecties op voorwaarde dat er dobbelsten met zijde 'R' in een van de selecties zit. Als dit niet zo is dan is de numerieke waarde 0.

src/js/regenwormen/Stapel.js

De klasse Stapel heeft een array `_selecties` als eigenschap.

De constructor heeft geen argumenten. Binnen de constructor wordt `_selecties` geïnitieerd als zijnde een lege array.

De methode `voegSelectieToe` heeft `selectie` als argument. Controleer dat `selectie` een object van de klasse `Selectie` is. Indien dit niet zo is wordt een `Error` opgeworpen. Ook wordt er gecontroleerd dat de zijde van de dobbelsteen in de selectie nog niet voor komt als zijde in de array `_selecties`. Als dit wel zo is wordt een `Error` opgeworpen. Elke selectie die wel voldoet aan bovenstaande regels wordt toegevoegd aan `_selecties`.

Via de methode `getNumeriekWaarde` wordt de numerieke waarde van de stapel berekend en teruggekeerd. De numerieke waarde is 0 als de dobbelsteen met zijde 'R' nog niet geselecteerd is. De numeriek waarde is de som van alle numerieke waarden van de selecties anders.

test/js/regenwormen/Stapel.test.js

Maak een test die controleert dat er een `Error` opgeworpen wordt wanneer via de methode `voegSelectieToe` eerst de selectie bestaande uit 3 maal de dobbelsteen met zijde '1' en dan de selectie bestaande uit 2 maal de dobbelsteen met zijde '1' toegevoegd wordt.

Maak een test die controleert dat `getNumeriekeWaarde` 0 teruggeeft voor de opeenvolgende selecties:

- 3 maal de dobbelsteen met zijde '1'
- 2 maal de dobbelsteen met zijde '2'

Extra oefening 1

Breid voorbeeld3 uit. Maak de methode find in PersonRepository. Alle personen worden opgezocht en teruggegeven als array. Maak de methode findNames in PersonService. De methode find wordt aangeroepen en een array met namen wordt gesorteerd teruggegeven.

Maak een test waarbij je de methode findNames test.