



2019

Oppgave 1

Et program håndterer matretter som kan bestå av ulike ingredienser. Man kan tenke seg at ingrediensene behandles som objekter i programmet.

Skriv koden for klassen **Ingrediens** .

- Start med å definere følgende 5 variabler i klassen: navn på ingrediens, måleenhet, mengde på lager, innkjøpspris og holdbarhetsdato.
- Klassen skal inneholde en konstruktør som fyller alle disse variablene med verdier når det opprettes et nytt objekt av klassen.
- Holdbarhetsdato er et valgfritt argument i konstruktøren siden det kan være ukjent ved registreringen.
- Lag en funksjon som henter ut variablene og returnerer tekststrenger på formen: «Det er 15 kg mel på lager med holdbarhet til 15. juni 2014».
- Lag også en funksjon som returner utsalgspris ved å legge på 25% på innkjøpsprisen.
- Skriv til slutt en rutine/kodesnutt i skjemaet som bruker konstruktøren du har laget i klassen til å opprette et objekt for deretter å bruke en av funksjonene du har laget for å skrive ut resultatet på en webside.

```
class Ingrediens {
    navn: string;
    måleenhet: string;
    lagerMengde: number;
    innkjøpspris: number;
    holdbarhetsdato?: Date;

    constructor(navn: string, måleenhet: string, lagerMengde: number,
        innkjøpspris: number, holdbarhetsdato?: Date){
        this.navn = navn;
        this.måleenhet = måleenhet;
        this.lagerMengde = lagerMengde;
        this.innkjøpspris = innkjøpspris;
        this.holdbarhetsdato = holdbarhetsdato;
    }

    skrivUt(){
```

```

        let result: string = `Det er ${this.lagerMengde} ${this.måleenhet} på lager`;

        if(this.holdbarhetsdato){
            result += ` med holdbarhet til ${this.holdbarhetsdato}`;
        } else result += ".";

        return result;
    }

    utsalgspris(){
        let prosentTillegg: number = 1.25;
        let utsalgspris: number = this.innkjøpspris * prosentTillegg;

        return utsalgspris;

        //evt bare return this.innkjøpspris*1.25 direkte
    }
}

let mel = new Ingrediens("mel", "kg", 8, 20, new Date("07.11.2023"));
let sukker = new Ingrediens("sukker", "dl", 10, 40);

console.log(mel.skrivUt(), mel.utsalgspris());
console.log(sukker.skrivUt(), sukker.utsalgspris());

```

OUTPUT (i konsoll)

```

Det er 8 kg på lager med holdbarhet 2019_1.html:45
til Tue Jul 11 2023 00:00:00 GMT+0200
(sentraleuropeisk sommertid) 25
Det er 10 dl på lager. 50 2019_1.html:46
> |

```

Oppgave 2

Du skal skrive kildekoden til en applikasjon som hjelper familien med å planlegge og utføre dagligvarehandler. Varene som skal kjøpes skal lagres i en database, og i tillegg skal det kunne registreres om en vare har blitt plukket opp i butikken eller ikke.

Databasetabellen er allerede opprettet med følgende SQL-setning:

```

CREATE TABLE ShoppingList (
    id INT NOT NULL AUTO_INCREMENT,
    name TEXT,
    count INT,
    collected BOOLEAN,
    PRIMARY KEY(id)
);

```

Index

```
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { Component } from 'react-simplified';
import { NavLink, HashRouter, Route } from 'react-router-dom';
import { ShoppingList, shoppingService } from './services';
import { Alert, Card, Row, Column, Button, Form, NavBar } from './widgets';

class VareHandel extends Component {
  items: ShoppingList[] = [];
  newItem = new ShoppingList();

  render(){
    return(
      <div>
        <Card>
          <Row>
            <Column width={4}>
              <h4>Varenavn</h4>
            </Column>
            <Column width={2}>
              <h4>Antall</h4>
            </Column>
            <Column width={4}>
              <h4>Plukket opp</h4>
            </Column>
          </Row>

          { this.items?.map((item) => {
            return <Row key={item.id}>
              <Column width={4}>
                {item.name}
              </Column>
              <Column width={2}>
                {item.count}
              </Column>
              <Column width={3}>
                { item.collected ? "x"
                  : <Button.Light onClick={() => this.collectItem(item.id)}>
                    Collect
                  </Button.Light> }
              </Column>
            </Row>
          }) }

        </Card>

        <Card title="Ny vare:">
          <Row>
            <Column width={4}>
              <Form.Label>Navn</Form.Label>
              <Form.Input
                type='text'
                placeholder="varenavn">
```

```

        value={this.newItem.name}
        onChange={(e: any) => this.newItem.name = e.target.value}
      />
    </Column>
    <Column>
      <Form.Label>Antall</Form.Label>
      <Form.Input
        type='number'
        placeholder="antall av vare"
        value={this.newItem.count == 0 ? "" : this.newItem.count}
        onChange={(e: any) => this.newItem.count = Number(e.target.value)}
      />
    </Column>
    <Column width={4}>
      <Button.Light onClick={this.addItem}>
        Legg til vare
      </Button.Light>
    </Column>
  </Row>
</Card>
<Button.Danger onClick={() => this.reset}>Nullstill</Button.Danger>
</div>
)
}

mounted(){
  shoppingService.getItems((items) => {
    this.items = items;
  });
}

reset(){
  shoppingService.resetList(() => this.mounted());
}

addItem(){
  shoppingService.addItem(this.newItem.name, this.newItem.count, () => {
    this.newItem = new ShoppingList();
  });
  this.mounted();
}

collectItem(id: number){
  shoppingService.collectItem(id, () => this.mounted());
}
}

let root = document.getElementById('root');
if (root)
  createRoot(root).render(
    <div>
      <Alert />
      <HashRouter>
        <VareHandel />
      </HashRouter>
    </div>
  )

```

Services

```
import { pool } from './mysql-pool';
import type { RowDataPacket, ResultSetHeader } from 'mysql2';

export class ShoppingList {
  id!: number;
  name: string = "";
  count!: number;
  collected: boolean = false;
}

class ShoppingService {
  getItems(success: (items: ShoppingList[]) => void){
    pool.query(
      'SELECT * FROM ShoppingList', (error: Error, results: any) => {
        if(error) return console.error(error);

        success(results);
      })
  }

  resetList(success: () => void) {
    pool.query('DELETE FROM ShoppingList', null, (error: any) => {
      if (error) return console.error(error);

      success();
    });
  }

  addItem(name: string, count: number, success: () => void){
    pool.query(
      'INSERT INTO ShoppingList(name, count, collected) VALUES(?,?,false)', [name, count],
      (error: any) => {
        if(error) return console.error(error);

        success();
      }
    )
  }

  collectItem(id: number, success: () => void){
    pool.query(
      'UPDATE ShoppingList SET collected = true WHERE id =?', [id], (error: any) => {
        if(error) return console.error(error);

        success();
      }
    )
  }
}
```

```
export let shoppingService = new ShoppingService();
```

OUTPUT

Varenavn	Antall	Plukket opp
mel	12	x
sukker	6	x
banan	24	x
egg	4	Collect
RedBull	200	x
brød	3	Collect
brokkoli	14	x
gulrot	7	Collect
agurk	1	Collect

Ny vare:

Navn	Antall	Legg til vare
<input type="text"/>	<input type="text"/>	

Nullstill