***Лабораторна робота №2***

***Тема: Рядки. Масиви. Об’єкти***

***Мета роботи***: навчитися працювати з рядками, масивами та об’єктами та методами роботи з ними

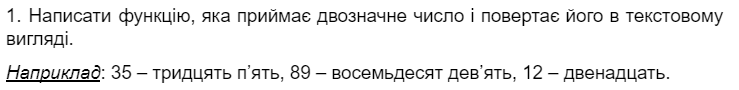
***Хід роботи:***

***Завдання на лабораторну роботу:***

1. ***Ознайомитись з теоретичними відомостями:***

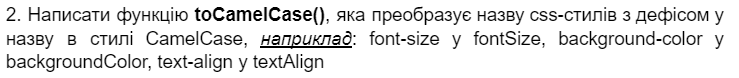
* <https://uk.javascript.info/string>
* <https://uk.javascript.info/object>
* <https://uk.javascript.info/object-copy>
* <https://uk.javascript.info/object-methods>
* <https://uk.javascript.info/constructor-new>
* <https://uk.javascript.info/array>
* <https://uk.javascript.info/array-methods>

1. ***Рядки(5 завдань):***

******

***Виконання:***

'use strict';  
  
*const checkTwoDigitsNumber* = (*twoDigitsNumber*) => {  
 *if* (*isNaN*(*twoDigitsNumber*) || *twoDigitsNumber* < 10 || *twoDigitsNumber* > 99) {  
 *alert*("Двозначне число введено не правильно!");  
 *return true*;  
 }  
  
 *return false*;  
};  
  
*const enterTwoDigitsNumber* = () => {  
 *let* twoDigitsNumber  
 *do* {  
 twoDigitsNumber = *parseInt*(*prompt*('Введіть двозначне у грн.: '));  
 } *while* (*checkTwoDigitsNumber*(twoDigitsNumber))  
  
 *return* twoDigitsNumber;  
};  
  
*const getNumberName* = (*number*) => {  
 *let* stringNumber = *number*.toString();  
 *let* unitsString = ['один ', 'два ', 'три ', 'чотири ', "п'ять ", 'шість ', 'сім ', 'вісім ', "дев'ять"];  
 *let* tenthString = ['двадцять ', 'тридцять ', 'сорок ', "п'ятдесят ", 'шістдесят ', 'сімдесят ', 'вісімдесят ', "дев'яносто "];  
 *let* tenthNumbers = [2, 3, 4, 5, 6, 7, 8, 9];  
 *let* numberName = '';  
  
 *if* (*number* === 10) *return* 'десять';  
 *if* (*number* === 11) *return* 'одинадцять';  
 *if* (*number* === 12) *return* 'дванадцять';  
 *if* (*number* === 13) *return* 'тринадцять';  
 *if* (*number* === 14) *return* 'чотирнадцять';  
 *if* (*number* === 15) *return* "п'ятнадцять";  
 *if* (*number* === 16) *return* 'шістнадцять';  
 *if* (*number* === 17) *return* 'сімнадцять';  
 *if* (*number* === 18) *return* 'вісімнадцять';  
 *if* (*number* === 19) *return* "дев'ятнадцять";  
  
 *for* (*let* i = 0; i < tenthString.length; i++) {  
 *if* (*parseInt*(stringNumber[0]) === tenthNumbers[i]) {  
 numberName = numberName.concat(tenthString[i]);  
 console.log(tenthString[i]);  
 *break*;  
 }  
 }  
  
 *for* (*let* i = 0; i < unitsString.length; i++) {  
 *if* (*parseInt*(stringNumber[1]) === i + 1) {  
 numberName += unitsString[i];  
 console.log(unitsString[i]);  
 *break*;  
 }  
 }  
 console.log(numberName);  
 *return* numberName;  
};  
  
  
*let* number = *enterTwoDigitsNumber*();  
*let* numberName = *getNumberName*(number);  
*alert*(`Ім'я двозначного числа ${number}: ${numberName}`);

******

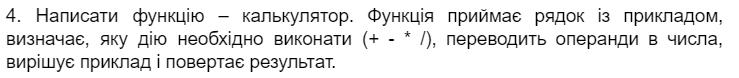
***Виконання:***

'use strict';  
  
*const toCamelCase* = (*cssStyle*) => {  
 *for* (*let* i = 0; i < *cssStyle*.length; i++) {  
 *if* (*cssStyle*[i] === '-') {  
 *cssStyle* = *cssStyle*.substring(0, i) + *cssStyle*[i + 1].toUpperCase() + *cssStyle*.substring(i + 2);  
 *break*;  
 }  
 }  
 *return cssStyle*;  
}  
  
*let* cssStyleString = *prompt*('Введіть css стиль: ').toString();  
*alert*(`Сконвертований css стиль за допомогою toCamelCase(): ${*toCamelCase*(cssStyleString)}`);

******

***Виконання:***

'use strict'  
  
*const toKebabCase* = (*cssStyle*) => {  
 *for* (*let* i = 0; i < *cssStyle*.length; i++) {  
 *if* (*cssStyle*[i] === *cssStyle*[i].toUpperCase()) {  
 *cssStyle* = *cssStyle*.substring(0, i) + '-' + *cssStyle*[i].toLowerCase() + *cssStyle*.substring(i + 1);  
 *break*;  
 }  
 }  
 *return cssStyle*;  
};  
  
*const checkUpperLetters* = (*string*) => {  
 *for* (*let* i = 0; i < *string*.length; i++) {  
 *if*(*string*[i] === *string*[i].toUpperCase()){  
 *return true*;  
 }  
 }  
 *return false*;  
};  
  
*let* cssStyleString = *prompt*('Введіть css стиль в CamelCase:').toString();  
*if*(*checkUpperLetters*(cssStyleString)){  
 *alert*(`Сконвертований css стиль за допомогою toKebabCase(): ${*toKebabCase*(cssStyleString)}`);  
} *else*{  
 *alert*('Введений рядок не є в стилі CamelCase!');  
}

**

***Виконання:***

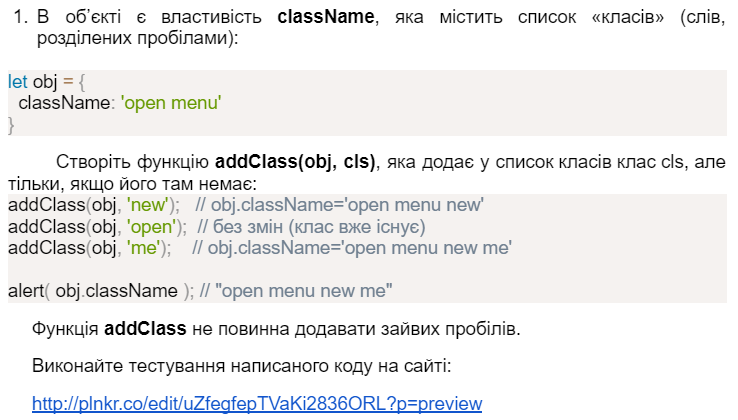
'use strict';  
  
*const isNum* = (*char*) => !*isNaN*(*char*);  
  
*const solveExpression* = (*expression*) => {  
 *let* nums = [];  
 *let* action;  
 *let* splitExpression = *expression*.split(' ', 3);  
  
 *for* (*let* i = 0; i < splitExpression.length; i++) {  
 *if* (*isNum*(splitExpression[i])) {  
 nums.push(+splitExpression[i]);  
 }  
 *if* (splitExpression[i] === '+' || splitExpression[i] === '-' || splitExpression[i] === '\*' || splitExpression[i] === '/') {  
 action = splitExpression[i];  
 }  
 }  
  
 *switch* (action) {  
 *case* '+':  
 *return* nums[0] + nums[1];  
 *case* '-':  
 *return* nums[0] - nums[1];  
 *case* '\*':  
 *return* nums[0] \* nums[1];  
 *case* '/':  
 *return* nums[0] / nums[1];  
 }  
};  
  
*const checkExpression* = (*expression*) => *isNum*(+*expression*[0]) && *expression*[1] === ' ' && *expression*[3] === ' ' && *isNum*(+*expression*[4]);  
  
  
*let* calculatorExpression = *prompt*("Введіть якийсь вираз для обчислення.\nОперації, які підтримуються(+, -, \*, /)\nПриклад: 2 + 2");  
*if* (*checkExpression*(calculatorExpression)) {  
 *alert*(`Результат обчислення виразу ${calculatorExpression} = ${*solveExpression*(calculatorExpression)}`);  
} *else* {  
 *alert*("Вираз введено не правильно!");  
}

******

***Виконання:***

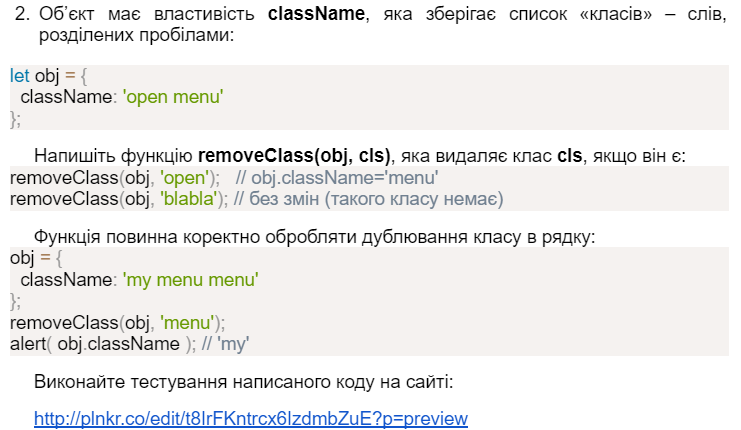
'use strict';  
  
*// text for check:  
// Bla bla bla 2017.12.12 bla bla bla lkasjdfklsad 2011.12.3 fdsdf 23.03.2001  
  
const formatTextDates* = (*textWithDates*) => {  
 *let* splitTextWithDates = *textWithDates*.split(' ');  
 *let* dateRegex = *new* RegExp('\\d{4}\\.(0?[1-9]|1[012])\\.(0?[1-9]|[12]\\d|3[01])\*');  
 console.log(splitTextWithDates);  
 *for* (*let* i = 0; i < splitTextWithDates.length; i++) {  
 *let* searchIndexDate = splitTextWithDates[i].search(dateRegex)  
 *if* (searchIndexDate !== -1) {  
 console.log(`${i} Рядок ${splitTextWithDates[i]} знайдено в ${searchIndexDate}`);  
 splitTextWithDates[i] = splitTextWithDates[i].slice(8, 10) + '.'  
 + splitTextWithDates[i].slice(6, 7) + '.'  
 + splitTextWithDates[i].slice(0, 4);  
 console.log(`Рядок змінено на ${splitTextWithDates[i]}`);  
 }  
 }  
  
 *return* splitTextWithDates.join(' ');  
};  
  
*let* textWithDates = prompt('Введіть який завгодно текст з датами типу рррр.мм.дд:');  
*let* newTextWithDates = formatTextDates(textWithDates);  
alert(`Новий текст з відформатованими датами:\n\n${newTextWithDates}`);

1. ***Об’єкти(3 завдання):***

******

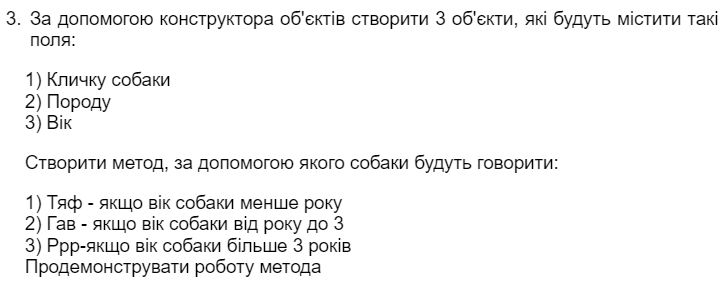
***Виконання:***

'use strict';  
  
*const checkClassNameInput* = (*enteredClassName*) => *enteredClassName*.search(' ') === -1;  
  
*const checkRepeatedClassNames* = (*className*, *classNames*) => {  
 *for* (*let* i = 0; i < *classNames*.length; i++) {  
 *if* (*classNames*[i] === *className*) {  
 *return false*;  
 }  
 }  
 *return true*;  
};  
  
*const addClass* = (*obj*, *newClass*) => {  
 *let* classNames = *obj*.className;  
 *if* (*checkRepeatedClassNames*(*newClass*, classNames.split(' '))) {  
 *obj*.className += ` ${*newClass*}`;  
 *return obj*;  
 }  
 *return obj*;  
};  
  
  
*let* obj = {  
 className: "open menu",  
};  
  
*let* newClass = *prompt*("Введіть ім'я класу для додавання:");  
*if*(*checkClassNameInput*(newClass)){  
 *let* newObj = *addClass*(obj, newClass);  
 *alert*(`Новий об'єкт: ${newObj.className}`);  
} *else* {  
 *alert*('Назву класу введено не правильно!')  
}

******

***Виконання:***

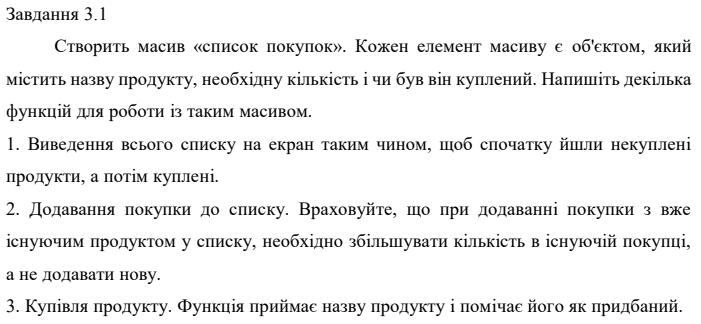
'use strict';  
  
*const checkClassNameInput* = (*enteredClassName*) => *enteredClassName*.search(' ') === -1;  
  
*const removeClass* = (*obj*, *existedClass*) => {  
 *let* classNames = *obj*.className;  
 *let* splitClassNames = classNames.split(' ');  
  
 *for* (*let* i = 0; i < splitClassNames.length; i++) {  
 console.log(splitClassNames[i]);  
 *if* (splitClassNames[i] === existedClass) {  
 *delete* splitClassNames[i];  
 obj.className = splitClassNames.join(' ').split(' ').join(' ');  
 *return* obj;  
 }  
 }  
 *return* obj;  
};  
  
  
*let* obj = {  
 className: "open menu new my",  
};  
*let* newClass = prompt("Введіть ім'я класу для видалення:");  
*if*(checkClassNameInput(newClass)){  
 *let* newObj = removeClass(obj, newClass);  
 alert(`Новий об'єкт: ${newObj.className}`);  
} *else* {  
 alert('Назву класу введено не правильно!')  
}



***Виконання:***

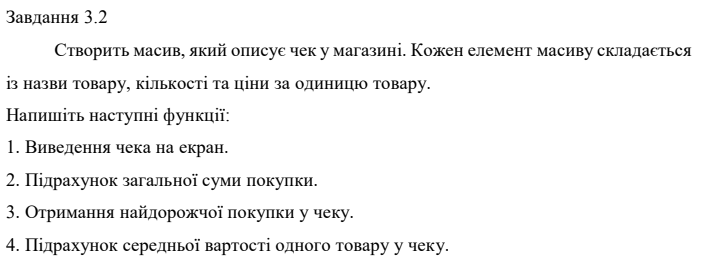
'use strict'  
  
*function Dog*(*dogName*, *dogBreed*, *dogAge*) {  
 *if* (!*new*.target) {  
 *return new Dog*(name);  
 }  
  
 *this*.name = *dogName*;  
 *this*.breed = *dogBreed*;  
 *this*.age = *dogAge*;  
  
 *this*.dogTalking = () => {  
 *if* (*this*.age < 1) *return* 'Тяф'  
 *if* (*this*.age >= 1 && *this*.age < 3) *return* 'Гав';  
 *if* (*this*.age >= 3) *return* 'Ррр';  
 }  
}  
  
*let* bulldog = *new Dog*("Archy", "bulldog", 4);  
*let* germanShepherd = *new Dog*("Ralf", "German shepherd", 2);  
*let* shibaInu = *new Dog*("Josh", "Shiba inu", 0.5);  
  
*alert*(`Собака породою ${bulldog.breed} з ім'ям ${bulldog.name} і віком ${bulldog.age} говорить: ${bulldog.dogTalking()}`);  
*alert*(`Собака породою ${germanShepherd.breed} з ім'ям ${germanShepherd.name} і віком ${germanShepherd.age} говорить: ${germanShepherd.dogTalking()}`);  
*alert*(`Собака породою ${shibaInu.breed} з ім'ям ${shibaInu.name} і віком ${shibaInu.age} говорить: ${shibaInu.dogTalking()}`);

1. ***Масиви(2 завдання):***



***Виконання:***

* HTML:
* <!doctype *html*>  
  <html *lang*="en">  
  <head>  
   <meta *charset*="UTF-8">  
   <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0, minimum-scale=1.0">  
   <meta *http-equiv*="X-UA-Compatible" *content*="ie=edge">  
   <title>Task 4.1</title>  
   <link *rel*="stylesheet" *href*="../styles/css/task4\_1.css">  
  </head>  
  <body>  
  <div *class*="wrapper">  
   <div *class*="container">  
   <section *class*="button\_\_section">  
   <label *for*="buttonCreateList" *class*="createList">Створити список</label>  
   <input *type*="button" *value*="Створити список" *id*="buttonCreateList">  
   <div *id*=outputCreateListField"></div>  
   </section>  
   <section *class*="button\_\_section">  
   <label *for*="buttonOutputList">Вивести список</label>  
   <input *type*="button" *value*="Вивести список" *id*="buttonOutputList">  
   <div *id*="outputListField"></div>  
   </section>  
   <section *class*="button\_\_section">  
   <table>  
   <tr>  
   <td> <label *for*="buttonAddProductToList">Додати покупку до списку</label></td>  
   <td><input *type*="button" *value*="Додати" *id*="buttonAddProductToList"></td>  
   </tr>  
   <tr>  
   <td><label *for*="productName">Назва продукту:</label></td>  
   <td><input *type*="text" *name*="productName" *id*="productName"></td>  
   </tr>  
   <tr>  
   <td><label *for*="productCount">Кількість для покупки:</label></td>  
   <td><input *type*="text" *name*="productCount" *id*="productCount"></td>  
   </tr>  
   <tr>  
   <td><label *for*="productStatus">Статус продукту(+, -):</label></td>  
   <td><input *type*="text" *name*="productStatus" *id*="productStatus"></td>  
   </tr>  
   </table>  
   <div *id*="outputAddElemToListField"></div>  
   </section>  
   <section *class*="button\_\_section">  
   <table>  
   <tr>  
   <td><label *for*="buttonMarkProductInList">Помітити покупку як придбану</label></td>  
   <td><input *type*="button" *value*="Помітити" *id*="buttonMarkProductInList"></td>  
   </tr>  
   <tr>  
   <td><label *for*="markProductInList">Введіть назву продукту:</label></td>  
   <td><input *type*="text" *name*="markProductInList" *id*="markProductInList"></td>  
   </tr>  
   </table>  
   <div *id*="outputMarkProductInListField"></div>  
   </section>  
   </div>  
  </div>  
    
   <script *src*="../js/task4\_1.js"></script>  
  </body>  
  </html>
* SASS:
* \*  
   margin: 0  
   padding: 0  
    
  .*wrapper* width: 100%  
   height: 100vh  
   display: flex  
   align-items: center  
   flex-direction: column  
    
  .*button\_\_section* border: 2px solid black  
   margin: 10px 0  
   padding: 10px  
   max-width: 30vw  
    
  .*outputSection* border: 1px solid black  
   margin: 10px 0  
   p  
   text-align: center  
   padding-bottom: 5px
* JS:
* 'use strict';  
    
  *function getRandomInt*(*fromNumber*, *toNumber*) {  
   *return* Math.round(Math.random() \* (*toNumber* - *fromNumber*) + *fromNumber*);  
  }  
    
  *function ShoppingList*(*productName*, *productCount*, *productBoughtStatus*) {  
   *this*.productName = *productName*;  
   *this*.productCount = *productCount*;  
   *this*.productBoughtStatus = *productBoughtStatus*;  
  }  
    
  *function createShoppingList*() {  
   *let* shoppingList = [];  
   *let* someProductNames = ['Помідори', 'Кетчуп', 'Гречка', 'Гарбуз', 'Печиво',  
   'Вода', 'Вино', 'Виноград', 'Манго', 'Соняшникова олія'];  
   *let* someProductBoughtStatus = [*true*, *false*];  
    
   *for* (*let* i = 0; i < 10; i++) {  
   shoppingList.push(*new ShoppingList*(someProductNames[i], *getRandomInt*(1, 10), someProductBoughtStatus[*getRandomInt*(0, 1)]));  
   }  
    
   *return* shoppingList;  
  }  
    
  *function compareObjStatus*(*obj1*, *obj2*) {  
   *if*(*obj1*.productBoughtStatus > *obj2*.productBoughtStatus)  
   *return* 1;  
    
   *if*(*obj1*.productBoughtStatus < *obj2*.productBoughtStatus)  
   *return* -1;  
    
   *return* 0;  
  }  
    
  *function sortShoppingListByBoughtStatus*(*objArray*) {  
   *return objArray*.sort(*compareObjStatus*);  
  }  
    
  *function getShoppingListOutPut*(*shoppingList*) {  
   *let* outputString = '';  
    
   *for* (*let* i = 0; i < *shoppingList*.length; i++) {  
   outputString += `<section class="outputSection"><p class="id">№${i + 1}</p>`;  
   outputString += `<p>Назва продукту: <strong>${*shoppingList*[i].productName}</strong></p>`;  
   outputString += `<p>Кількість для покупки: <strong>${*shoppingList*[i].productCount}</strong></p>`;  
   outputString += `<p>Статус покупки: <strong>${*shoppingList*[i].productBoughtStatus}</strong></p></section>`;  
   }  
    
   *return* outputString;  
  }  
    
  *function addProductToShoppingList*(*objs*, *productName*, *productCount*, *productStatus*){  
   *for* (*let* i = 0; i < *objs*.length; i++) {  
   *if*(*objs*[i].productName === *productName*) {  
   *return objs*;  
   }  
   }  
    
   *if*(*productStatus* === '+') *productStatus* = *true*;  
   *if*(*productStatus* === '-') *productStatus* = *false*;  
    
   *objs*.push(*new ShoppingList*(*productName*, *productCount*, *productStatus*));  
    
   *return objs*;  
  }  
    
  *function checkProductAddInputs*(*productName*, *productCount*, *productStatus*) {  
   *return* !(*isNaN*(*productCount*) || *productCount* <= 0 || (*productStatus* !== '+' && *productStatus* !== '-'));  
  }  
    
  *function markProductStatus*(*objs*, *productName*) {  
   *for* (*let* i = 0; i < *objs*.length; i++) {  
   *if*(*objs*[i].productName === *productName*){  
   *objs*[i].productBoughtStatus = *true*;  
   *return objs*;  
   }  
   }  
    
   *return objs*;  
  }  
    
  *function checkExistProductName*(*objs*, *productName*){  
   *for* (*let* i = 0; i < *objs*.length; i++) {  
   *if*(*objs*[i].productName === *productName*){  
   *return true*;  
   }  
   }  
    
   *return false*;  
  }  
    
  *let* shoppingList = *null*;  
    
  *let* buttonCreateList = document.querySelector('#buttonCreateList');  
  buttonCreateList.addEventListener('click', () => {  
   shoppingList = *createShoppingList*();  
   document.getElementById('outputCreateListField"').innerHTML = '<p><strong>Список покупок створено!</strong></p>';  
  });  
    
  *let* buttonOutputList = document.querySelector('#buttonOutputList');  
  buttonOutputList.addEventListener('click', () => {  
   *if*(shoppingList === *null*) *alert*('Список покупок пустий!');  
   *else*{  
   document.getElementById('outputListField').innerHTML = *getShoppingListOutPut*(*sortShoppingListByBoughtStatus*(shoppingList));  
   }  
  });  
    
  *let* buttonAddProductToList = document.querySelector('#buttonAddProductToList');  
  buttonAddProductToList.addEventListener('click', () => {  
   *if*(shoppingList === *null*) {  
   document.getElementById('outputAddElemToListField').innerHTML = '<p><strong>Список покупок пустий!</strong></p>';  
   }  
   *else*{  
   *let* productName = document.getElementById('productName').value;  
   *let* productCount = document.getElementById('productCount').value;  
   *let* productStatus = document.getElementById('productStatus').value;  
   *if*(productName === *null* || productCount === *null* || productStatus === *null*)  
   document.getElementById('outputAddElemToListField').innerHTML = '<p><strong>Поле(я) для введення пусті!</strong></p>';  
   *else*{  
   *if*(*checkProductAddInputs*(productName, productCount, productStatus)){  
   shoppingList = *addProductToShoppingList*(shoppingList, productName, productCount, productStatus);  
   document.getElementById('outputAddElemToListField').innerHTML = '<p><strong>Елемент додано до списку покупок!</strong></p>';  
   } *else*{  
   document.getElementById('outputAddElemToListField').innerHTML = '<p><strong>Значення введено не правильно!</strong></p>';  
   }  
   }  
   }  
  });  
    
  *let* buttonMarkProductInList = document.querySelector('#buttonMarkProductInList');  
  buttonMarkProductInList.addEventListener('click', () => {  
   *if*(shoppingList === *null*) {  
   document.getElementById('outputMarkProductInListField').innerHTML = '<p><strong>Список покупок пустий!</strong></p>';  
   } *else*{  
   *let* productName = document.getElementById('markProductInList').value;  
   *if*(productName === *null*){  
   document.getElementById('outputMarkProductInListField').innerHTML = '<p><strong>Поле для введення пусті!</strong></p>';  
   } *else*{  
   *if*(*checkExistProductName*(shoppingList, productName)){  
   shoppingList = *markProductStatus*(shoppingList, productName);  
   document.getElementById('outputMarkProductInListField').innerHTML = '<p><strong>Продукт було помічено як куплений!</strong></p>';  
   } *else*{  
   document.getElementById('outputMarkProductInListField').innerHTML = '<p><strong>Продукт НЕ було помічено як куплений!</strong></p>';  
   }  
   }  
   }  
  });



***Виконання:***

* HTML:
* <!doctype *html*>  
  <html *lang*="en">  
  <head>  
   <meta *charset*="UTF-8">  
   <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0, minimum-scale=1.0">  
   <meta *http-equiv*="X-UA-Compatible" *content*="ie=edge">  
   <title>Task 4.2</title>  
   <link *rel*="stylesheet" *href*="../styles/css/task4\_2.css">  
  </head>  
  <body>  
  <div *class*="wrapper">  
   <div *class*="container">  
   <section *class*="button\_\_section">  
   <label *for*="buttonCreateReceipt" *class*="createList">Створити чек</label>  
   <input *type*="button" *value*="Створити чек" *id*="buttonCreateReceipt">  
   <div *id*=outputCreateReceiptField"></div>  
   </section>  
   <section *class*="button\_\_section">  
   <label *for*="buttonOutputReceipt">Вивести чек</label>  
   <input *type*="button" *value*="Вивести чек" *id*="buttonOutputReceipt">  
   <div *id*="outputReceiptField"></div>  
   </section>  
   <section *class*="button\_\_section">  
   <label *for*="buttonCountTotalReceiptSum">Підрахувати загальну суму покупки</label>  
   <input *type*="button" *value*="Підрахувати загальну суму" *id*="buttonCountTotalReceiptSum">  
   <div *id*="outputCountTotalReceiptSumField"></div>  
   </section>  
   <section *class*="button\_\_section">  
   <label *for*="buttonGetMostExpensivePurchase">Отримати найдорожчу покупку чеку</label>  
   <input *type*="button" *value*="Отримати" *id*="buttonGetMostExpensivePurchase">  
   <div *id*="outputGetMostExpensivePurchaseField"></div>  
   </section>  
   <section *class*="button\_\_section">  
   <label *for*="buttonCountAverageCostPurchase">Підрахувати середню вартість одного товару</label>  
   <input *type*="button" *value*="Підрахувати середню вартість" *id*="buttonCountAverageCostPurchase">  
   <div *id*="outputCountAverageCostPurchaseField"></div>  
   </section>  
   </div>  
  </div>  
    
  <script *src*="../js/task4\_2.js"></script>  
  </body>  
  </html>
* SASS:
* \*  
   margin: 0  
   padding: 0  
   box-sizing: border-box  
    
  .*wrapper* width: 100%  
   height: 100vh  
   display: flex  
   align-items: center  
   flex-direction: column  
    
  .*button\_\_section* border: 2px solid black  
   margin: 10px 0  
   padding: 10px  
   max-width: 35vw  
    
   input[*type*="button"]  
   border: 1px solid black  
   background-color: lightgrey  
   padding: 5px 10px  
   margin: 5px 0  
   font-weight: bold  
   cursor: pointer  
    
  .*outputSection* border: 1px solid black  
   margin: 10px 0  
    
   p  
   text-align: center  
   padding-bottom: 5px
* JS:
* 'use strict';  
    
  *function getRandomFloat*(*fromNumber*, *toNumber*) {  
   *return* Math.round((Math.random() \* (*toNumber* - *fromNumber*) + *fromNumber*) \* 100) / 100;  
  }  
    
  *function getRandomInt*(*fromNumber*, *toNumber*) {  
   *return* Math.round(Math.random() \* (*toNumber* - *fromNumber*) + *fromNumber*);  
  }  
    
  *function Receipt*(*productName*, *productCount*, *productPrice*){  
   *this*.productName = *productName*;  
   *this*.productCount = *productCount*;  
   *this*.productPrice = *productPrice*;  
  }  
    
  *function createReceipt*() {  
   *let* newReceipt = [];  
   *let* someProductNames = ['Помідори', 'Кетчуп', 'Гречка', 'Гарбуз', 'Печиво',  
   'Вода', 'Вино', 'Виноград', 'Манго', 'Соняшникова олія'];  
    
   *for* (*let* i = 0; i < 10; i++) {  
   newReceipt.push(*new Receipt*(someProductNames[i], *getRandomInt*(0, 10), *getRandomFloat*(1, 250)));  
   }  
    
   *return* newReceipt;  
  }  
    
  *function getReceiptOutput*(*receipt*){  
   *let* outputString = '';  
    
   *for* (*let* i = 0; i < *receipt*.length; i++) {  
   outputString += `<section class="outputSection"><p class="id">№${i + 1}</p>`;  
   outputString += `<p>Назва продукту: <strong>${*receipt*[i].productName}</strong></p>`;  
   outputString += `<p>Кількість для покупки: <strong>${*receipt*[i].productCount}</strong></p>`;  
   outputString += `<p>Ціна покупки за 1шт./кг.: <strong>${*receipt*[i].productPrice}</strong></p></section>`;  
   }  
    
   *return* outputString;  
  }  
    
  *function getTotalReceiptSum*(*receipt*){  
   *let* receiptSum = 0;  
    
   *for* (*let* i = 0; i < *receipt*.length; i++) {  
   receiptSum += *receipt*[i].productPrice;  
   }  
    
   *return* receiptSum;  
  }  
    
  *function getMostExpensivePurchase*(*receipt*){  
   *let* receiptPrices = [];  
    
   *for* (*let* i = 0; i < *receipt*.length; i++) {  
   receiptPrices.push(*receipt*[i].productPrice)  
   }  
    
   *return* Math.max(...receiptPrices);  
  }  
    
  *function countAverageCostPurchase*(*receipt*) {  
   *return* Math.round((*getTotalReceiptSum*(*receipt*) / *receipt*.length) \* 100) / 100;  
  }  
    
  *let* receipt = *null*;  
  *let* buttonCreateReceipt = document.querySelector('#buttonCreateReceipt');  
  buttonCreateReceipt.addEventListener('click', () => {  
   receipt = *createReceipt*();  
   document.getElementById('outputCreateReceiptField"').innerHTML = '<p><strong>Чек створено!</strong></p>';  
  });  
    
  *let* buttonOutputReceipt = document.querySelector('#buttonOutputReceipt');  
  buttonOutputReceipt.addEventListener('click', () => {  
   *if*(receipt === *null*) {  
   document.getElementById('outputReceiptField').innerHTML = '<p><strong>Чек не створений!</strong></p>';  
   } *else* {  
   document.getElementById('outputReceiptField').innerHTML = *getReceiptOutput*(receipt);  
   }  
  });  
    
  *let* buttonCountTotalReceiptSum = document.querySelector('#buttonCountTotalReceiptSum');  
  buttonCountTotalReceiptSum.addEventListener('click', () => {  
   *if*(receipt === *null*) {  
   document.getElementById('outputCountTotalReceiptSumField').innerHTML = '<p><strong>Чек не створений!</strong></p>';  
   } *else* {  
   document.getElementById('outputCountTotalReceiptSumField').innerHTML = `Загальна сума покупки: ${*getTotalReceiptSum*(receipt)}`  
   }  
  });  
    
  *let* buttonGetMostExpensivePurchase = document.querySelector('#buttonGetMostExpensivePurchase');  
  buttonGetMostExpensivePurchase.addEventListener('click', () => {  
   *if*(receipt === *null*) {  
   document.getElementById('outputGetMostExpensivePurchaseField').innerHTML = '<p><strong>Чек не створений!</strong></p>';  
   } *else* {  
   document.getElementById('outputGetMostExpensivePurchaseField').innerHTML = `Найдорожчий товар в чеку: ${*getMostExpensivePurchase*(receipt)}`  
   }  
  });  
    
  *let* buttonCountAverageCostPurchase = document.querySelector('#buttonCountAverageCostPurchase');  
  buttonCountAverageCostPurchase.addEventListener('click', () => {  
   *if*(receipt === *null*) {  
   document.getElementById('outputCountAverageCostPurchaseField').innerHTML = '<p><strong>Чек не створений!</strong></p>';  
   } *else* {  
   document.getElementById('outputCountAverageCostPurchaseField').innerHTML = `Середня вартість 1 товару в чеку: ${*countAverageCostPurchase*(receipt)}`  
   }  
  });

***Демонастрація виконання:*** під час лабораторної роботи.

***Репозиторій лабораторної роботи:***

* https://github.com/AndriiBabushko/Frontend

***Висновок:*** під час виконання лабораторної роботи було отримано навички роботи з рядками, об'єктами та масивами на мові JavaScript, використовуючи методи даних типів даних.