

Sprawozdanie nr 2

Temat: Programowanie obiektowe, aplikacja TODO

Przebieg zadania:

Czym jest programowanie obiektowe

Jest to paradygmat programowania, w którym programy definiuje się za pomocą obiektów – elementów łączących stan i zachowanie. Obiektowy program komputerowy wyrażony jest jako zbiór takich obiektów, komunikujących się pomiędzy sobą w celu wykonywania zadań.

Podejście to różni się od tradycyjnego programowania proceduralnego, gdzie dane i procedury nie są ze sobą bezpośrednio związane. Programowanie obiektowe ma ułatwić pisanie, konserwację i wielokrotne użycie programów lub ich fragmentów.

Html

```
<!DOCTYPE html>

<html lang="pl" >

<head>

  <meta charset="UTF-8">

  <title>Lista zadań</title>

  <link rel="stylesheet" href="style.css">

</head>

<body>

  <div class="main">

    <h2>Lista zadań</h2>

    <h3>Dodaj zadanie</h3>

    <p>

      <input id="new-task" type="text"><button>Dodaj</button>

    </p>

    <h3>Do zrobienia</h3>

    <ul id="incomplete-tasks">

    </ul>

    <h3>Wykonane</h3>

    <ul id="completed-tasks">
```

```
        </ul>

    </div>

    <script src="script.js"></script>

</body>

</html>
```

CSS

```
body {

    background: #fff;

    color: #333;

    font-family: Lato, sans-serif;

    background-color:#e4e4e4;

}


.main {

    display: block;

    width: 400px;

    margin: 10px auto 100px;

    background-color:#fff;

    padding:0px 10px 10px 10px;

    border-radius:10px

}


h2 {

    text-align:center;

    padding-top:10px;

    margin-bottom:0px;

}


ul {

    margin: 0;

    padding: 0;

}


li * {

    float: left;

}
```

li,

h3 {

clear: both;

list-style: none;

}

input,

button {

outline: none;

}

button {

background: none;

border: 0px;

color: #888;

font-size: 15px;

width: 60px;

margin: 10px 0 0;

font-family: Lato, sans-serif;

cursor: pointer;

}

button:hover {

color: #333;

}

h3,

label[for='new-task'] {

color: #333;

font-weight: 700;

font-size: 15px;

border-bottom: 2px solid #333;

padding: 20px 0 10px;

margin: 0;

text-transform: uppercase;

}

input[type="text"] {

```
    margin: 0;

    font-size: 18px;

    line-height: 18px;

    height: 18px;

    padding: 10px;

    border: 1px solid #ddd;

    background: #fff;

    border-radius: 6px;

    font-family: Lato, sans-serif;

    color: #888;
}
```

```
input[type="text"]:focus {

    color: #333;

}
```

```
label[for='new-task'] {

    display: block;

    margin: 0 0 20px;

}
```

```
input#new-task {

    float: left;

    width: 318px;

}
```

```
p > button:hover {

    color: #0FC57C;

}
```

```
li {

    overflow: hidden;

    padding: 20px 0;

    border-bottom: 1px solid #eee;

}
```

```
li > input[type="checkbox"] {

    margin: 0 10px;

}
```

```

        position: relative;

        top: 15px;
    }

    li > label {

        font-size: 18px;

        line-height: 40px;

        width: 237px;

        padding: 0 0 0 11px;
    }

    li > input[type="text"] {

        width: 226px;
    }

    li > .delete:hover {

        color: #CF2323;
    }

    #completed-tasks label {

        text-decoration: line-through;

        color: #888;
    }

    ul li input[type=text] {

        display: none;
    }

    ul li.editMode input[type=text] {

        display: block;
    }

    ul li.editMode label {

        display: none;
    }

```

JS

```

var taskInput = document.getElementById("new-task");

var addButton = document.getElementsByTagName("button")[0];

```

```
var incompleteTasksHolder = document.getElementById("incomplete-tasks");
var completedTasksHolder = document.getElementById("completed-tasks");
var createNewTaskElement = function(taskString) {
```

```
    var listItem = document.createElement("li");
```

```
    var checkBox = document.createElement("input");
```

```
    var label = document.createElement("label");
```

```
    var editInput = document.createElement("input");
```

```
    var editButton = document.createElement("button");
```

```
    var deleteButton = document.createElement("button");
```

```
    checkBox.type = "checkbox";
```

```
    editInput.type = "text";
```

```
    editButton.innerText = "Edytuj";
```

```
    editButton.className = "edit";
```

```
    deleteButton.innerText = "Usuń";
```

```
    deleteButton.className = "delete";
```

```
    label.innerText = taskString;
```

```
    listItem.appendChild(checkBox);
```

```
    listItem.appendChild(label);
```

```
    listItem.appendChild(editInput);
```

```
    listItem.appendChild(editButton);
```

```
    listItem.appendChild(deleteButton);
```

```
    return listItem;
```

```
}
```

```
var addTask = function() {
```

```
    var listItem = createNewTaskElement(taskInput.value);
```

```
    incompleteTasksHolder.appendChild(listItem);
```

```
    bindTaskEvents(listItem, taskCompleted);
```

```
    taskInput.value = "";
```

```
}
```

```
var editTask = function() {  
    var listItem = this.parentNode;  
  
    var editInput = listItem.querySelector("input[type=text]");  
    var label = listItem.querySelector("label");  
  
    var containsClass = listItem.classList.contains("editMode");  
  
    if (containsClass) {  
        label.innerText = editInput.value;  
    } else {  
        editInput.value = label.innerText;  
    }  
  
    listItem.classList.toggle("editMode");  
  
}
```

```
var deleteTask = function() {  
    var listItem = this.parentNode;  
    var ul = listItem.parentNode;  
  
    ul.removeChild(listItem);  
  
}
```

```
var taskCompleted = function() {  
    var listItem = this.parentNode;  
    completedTasksHolder.appendChild(listItem);  
    bindTaskEvents(listItem, taskIncomplete);  
  
}
```

```
var taskIncomplete = function() {  
    var listItem = this.parentNode;  
    incompleteTasksHolder.appendChild(listItem);  
    bindTaskEvents(listItem, taskCompleted);  
  
}
```

```
var bindTaskEvents = function(taskListItem, checkBoxEventHandler) {
```

```
var checkBox = taskListItem.querySelector("input[type=checkbox]");
var editButton = taskListItem.querySelector("button.edit");
var deleteButton = taskListItem.querySelector("button.delete");

editButton.onclick = editTask;

deleteButton.onclick = deleteTask;

checkBox.onchange = checkBoxEventHandler;
}

addButton.addEventListener("click", addTask);

for (var i = 0; i < incompleteTasksHolder.children.length; i++) {
    bindTaskEvents(incompleteTasksHolder.children[i], taskCompleted);
}

for (var i = 0; i < completedTasksHolder.children.length; i++) {
    bindTaskEvents(completedTasksHolder.children[i], taskIncomplete);
}
```