# Department of Computing

**SE-312: Software Construction**

**Class: BESE 9AB**

# Lab 04: The Module Pattern

**Date: 15 March 2021**

**Hassaan Ali Mehmood**

**BESE 9A**

**264947**

**Time: 09:00-12:00pm & 02:00-05:00pm**

# Instructor: Dr. Seema Jehan

**Lab Engineer: Mr. Aftab Farooq**

# 

# Lab 04: The Module Pattern

**Objectives**

The objective of this lab is helping students to familiarize themselves with basic concepts of the Module Pattern. They will practice the concept of module pattern by implementing three modules (Budget Controller, UI Controller, and AppController) for the provided Budget app.

**Tools/Software Requirement**

Notepad, browser

**Helping Material:**

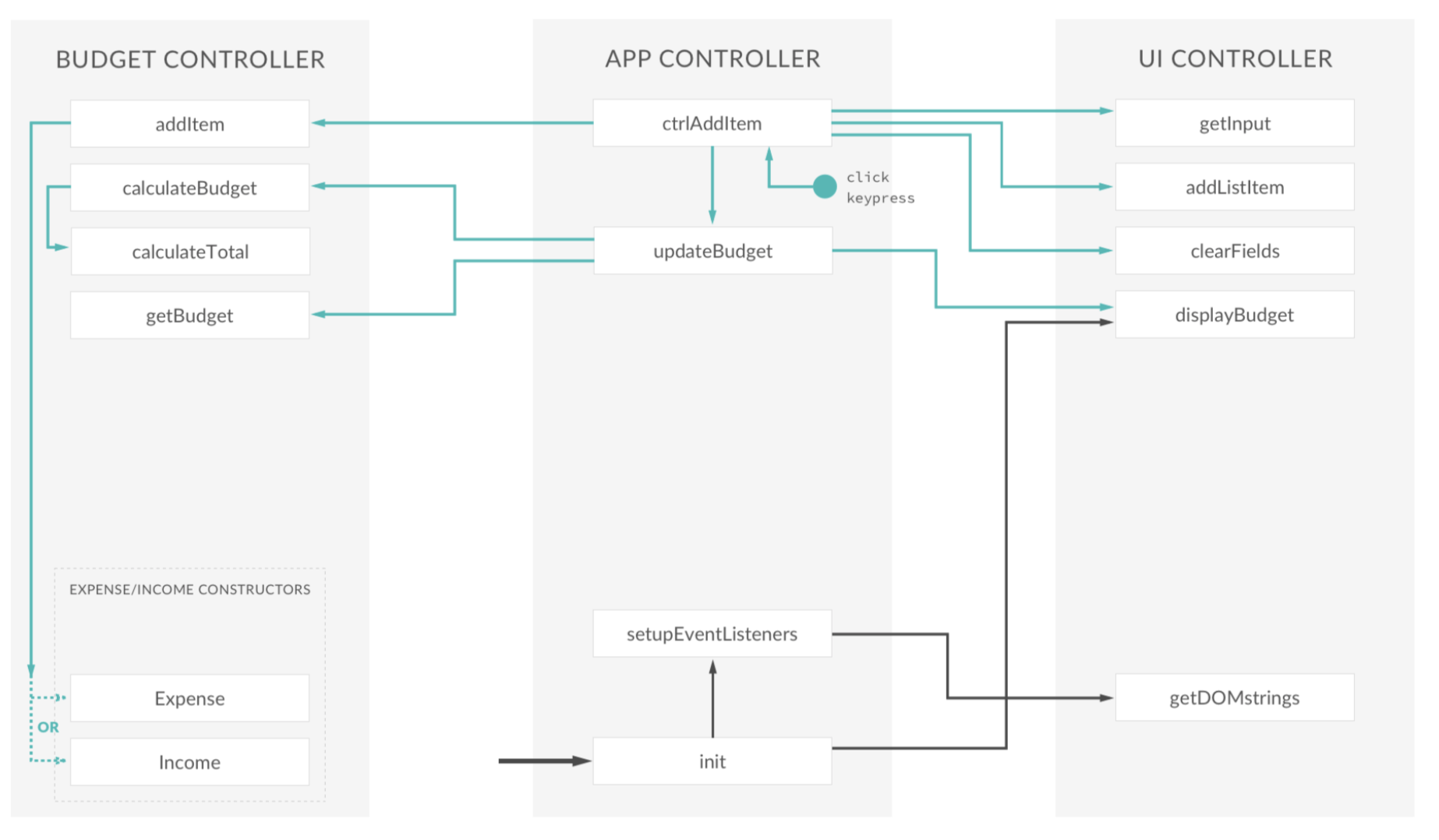
File Uploaded for Assignment 1 on LMS

**Lab Tasks:**

**TASK1:**

Implement the Budget App that allows:

1. To add incomes and expenses for a certain month
   1. Add event handler to get input values
   2. Add the new item to the data structure
   3. Add the new item to the UI
2. Calculates the savings and updates the Budget
   1. Calculate the Budget
   2. Update the UI



|  |
| --- |
| **Solution** |
| **Task 1 Code:**  // BUDGET CONTROLLERvar budgetController = (**function**() {    var Expense = **function**(id, description, value) {  **this**.id = id;  **this**.description = description;  **this**.value = value;  **this**.percentage = -1;  };      Expense.prototype.calcPercentage = **function**(totalIncome) {  **if** (totalIncome > 0) {  **this**.percentage = Math.round((**this**.value / totalIncome) \* 100);  } **else** {  **this**.percentage = -1;  }  };      Expense.prototype.getPercentage = **function**() {  **return** **this**.percentage;  };      var Income = **function**(id, description, value) {  **this**.id = id;  **this**.description = description;  **this**.value = value;  };      var calculateTotal = **function**(type) {  var sum = 0;  data.allItems[type].forEach(**function**(cur) {  sum += cur.value;  });  data.totals[type] = sum;  };      var data = {  **allItems**: {  **exp**: [],  **inc**: []  },  **totals**: {  **exp**: 0,  **inc**: 0  },  **budget**: 0,  **percentage**: -1  };      **return** {  **addItem**: **function**(type, des, val) {  var newItem, ID;      // Create new ID  **if** (data.allItems[type].length > 0) {  ID = data.allItems[type][data.allItems[type].length - 1].id + 1;  } **else** {  ID = 0;  }    // Create new item based on 'inc' or 'exp' type  **if** (type === 'exp') {  newItem = **new** Expense(ID, des, val);  } **else** **if** (type === 'inc') {  newItem = **new** Income(ID, des, val);  }    // Push it into our data structure  data.allItems[type].push(newItem);    // Return the new element  **return** newItem;  },      **deleteItem**: **function**(type, id) {  var ids, index;      ids = data.allItems[type].map(**function**(current) {  **return** current.id;  });  index = ids.indexOf(id);  **if** (index !== -1) {  data.allItems[type].splice(index, 1);  }    },      **calculateBudget**: **function**() {    // calculate total income and expenses  calculateTotal('exp');  calculateTotal('inc');    // Calculate the budget: income - expenses  data.budget = data.totals.inc - data.totals.exp;    // calculate the percentage of income that we spent  **if** (data.totals.inc > 0) {  data.percentage = Math.round((data.totals.exp / data.totals.inc) \* 100);  } **else** {  data.percentage = -1;  }    // Expense = 100 and income 300, spent 33.333% = 100/300 = 0.3333 \* 100  },    **calculatePercentages**: **function**() {    data.allItems.exp.forEach(**function**(cur) {  cur.calcPercentage(data.totals.inc);  });  },      **getPercentages**: **function**() {  var allPerc = data.allItems.exp.map(**function**(cur) {  **return** cur.getPercentage();  });  **return** allPerc;  },      **getBudget**: **function**() {  **return** {  **budget**: data.budget,  **totalInc**: data.totals.inc,  **totalExp**: data.totals.exp,  **percentage**: data.percentage  };  },    **testing**: **function**() {  console.log(data);  }  };    })();  // UI CONTROLLERvar UIController = (**function**() {    var DOMstrings = {  **inputType**: '.add\_\_type',  **inputDescription**: '.add\_\_description',  **inputValue**: '.add\_\_value',  **inputBtn**: '.add\_\_btn',  **incomeContainer**: '.income\_\_list',  **expensesContainer**: '.expenses\_\_list',  **budgetLabel**: '.budget\_\_value',  **incomeLabel**: '.budget\_\_income--value',  **expensesLabel**: '.budget\_\_expenses--value',  **percentageLabel**: '.budget\_\_expenses--percentage',  **container**: '.container',  **expensesPercLabel**: '.item\_\_percentage',  **dateLabel**: '.budget\_\_title--month'  };      var formatNumber = **function**(num, type) {  var numSplit, int, dec, type;    num = Math.abs(num);  num = num.toFixed(2);  numSplit = num.split('.');  int = numSplit[0];  **if** (int.length > 3) {  int = int.substr(0, int.length - 3) + ',' + int.substr(int.length - 3, 3); //input 23510, output 23,510  }  dec = numSplit[1];  **return** (type === 'exp' ? '-' : '+') + ' ' + int + '.' + dec;  };      var nodeListForEach = **function**(list, callback) {  **for** (var i = 0; i < list.length; i++) {  callback(list[i], i);  }  };      **return** {  **getInput**: **function**() {  **return** {  **type**: document.querySelector(DOMstrings.inputType).value, // Will be either inc or exp  **description**: document.querySelector(DOMstrings.inputDescription).value,  **value**: parseFloat(document.querySelector(DOMstrings.inputValue).value)  };  },      **addListItem**: **function**(obj, type) {  var html, newHtml, element;  // Create HTML string with placeholder text    **if** (type === 'inc') {  element = DOMstrings.incomeContainer;    html = '<div class="item clearfix" id="inc-%id%"> <div class="item\_\_description">%description%</div><div class="right clearfix"><div class="item\_\_value">%value%</div><div class="item\_\_delete"><button class="item\_\_delete--btn"><i class="ion-ios-close-outline"></i></button></div></div></div>';  } **else** **if** (type === 'exp') {  element = DOMstrings.expensesContainer;    html = '<div class="item clearfix" id="exp-%id%"><div class="item\_\_description">%description%</div><div class="right clearfix"><div class="item\_\_value">%value%</div><div class="item\_\_percentage">21%</div><div class="item\_\_delete"><button class="item\_\_delete--btn"><i class="ion-ios-close-outline"></i></button></div></div></div>';  }    // Replace the placeholder text with some actual data  newHtml = html.replace('%id%', obj.id);  newHtml = newHtml.replace('%description%', obj.description);  newHtml = newHtml.replace('%value%', formatNumber(obj.value, type));    // Insert the HTML into the DOM  document.querySelector(element).insertAdjacentHTML('beforeend', newHtml);  },      **deleteListItem**: **function**(selectorID) {    var el = document.getElementById(selectorID);  el.parentNode.removeChild(el);    },      **clearFields**: **function**() {  var fields, fieldsArr;    fields = document.querySelectorAll(DOMstrings.inputDescription + ', ' + DOMstrings.inputValue);    fieldsArr = Array.prototype.slice.call(fields);    fieldsArr.forEach(**function**(current, index, array) {  current.value = "";  });    fieldsArr[0].focus();  },      **displayBudget**: **function**(obj) {  var type;  obj.budget > 0 ? type = 'inc' : type = 'exp';    document.querySelector(DOMstrings.budgetLabel).textContent = formatNumber(obj.budget, type);  document.querySelector(DOMstrings.incomeLabel).textContent = formatNumber(obj.totalInc, 'inc');  document.querySelector(DOMstrings.expensesLabel).textContent = formatNumber(obj.totalExp, 'exp');    **if** (obj.percentage > 0) {  document.querySelector(DOMstrings.percentageLabel).textContent = obj.percentage + '%';  } **else** {  document.querySelector(DOMstrings.percentageLabel).textContent = '---';  }    },      **displayPercentages**: **function**(percentages) {    var fields = document.querySelectorAll(DOMstrings.expensesPercLabel);    nodeListForEach(fields, **function**(current, index) {    **if** (percentages[index] > 0) {  current.textContent = percentages[index] + '%';  } **else** {  current.textContent = '---';  }  });    },      **displayMonth**: **function**() {  var now, months, month, year;    now = **new** Date();  //var christmas = new Date(2016, 11, 25);    months = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December'];  month = now.getMonth();    year = now.getFullYear();  document.querySelector(DOMstrings.dateLabel).textContent = months[month] + ' ' + year;  },      **changedType**: **function**() {    var fields = document.querySelectorAll(  DOMstrings.inputType + ',' +  DOMstrings.inputDescription + ',' +  DOMstrings.inputValue);    nodeListForEach(fields, **function**(cur) {  cur.classList.toggle('red-focus');  });    document.querySelector(DOMstrings.inputBtn).classList.toggle('red');    },      **getDOMstrings**: **function**() {  **return** DOMstrings;  }  };    })();  // GLOBAL APP CONTROLLERvar controller = (**function**(budgetCtrl, UICtrl) {    var setupEventListeners = **function**() {  var DOM = UICtrl.getDOMstrings();    document.querySelector(DOM.inputBtn).addEventListener('click', ctrlAddItem);  document.addEventListener('keypress', **function**(event) {  **if** (event.keyCode === 13 || event.which === 13) {  ctrlAddItem();  }  });    document.querySelector(DOM.container).addEventListener('click', ctrlDeleteItem);    document.querySelector(DOM.inputType).addEventListener('change', UICtrl.changedType);  };      var updateBudget = **function**() {    // 1. Calculate the budget  budgetCtrl.calculateBudget();    // 2. Return the budget  var budget = budgetCtrl.getBudget();    // 3. Display the budget on the UI  UICtrl.displayBudget(budget);  };      var updatePercentages = **function**() {    // 1. Calculate percentages  budgetCtrl.calculatePercentages();    // 2. Read percentages from the budget controller  var percentages = budgetCtrl.getPercentages();    // 3. Update the UI with the new percentages  UICtrl.displayPercentages(percentages);  };      var ctrlAddItem = **function**() {  var input, newItem;    // 1. Get the field input data  input = UICtrl.getInput();    **if** (input.description !== "" && !isNaN(input.value) && input.value > 0) {  // 2. Add the item to the budget controller  newItem = budgetCtrl.addItem(input.type, input.description, input.value);  // 3. Add the item to the UI  UICtrl.addListItem(newItem, input.type);  // 4. Clear the fields  UICtrl.clearFields();  // 5. Calculate and update budget  updateBudget();    // 6. Calculate and update percentages  updatePercentages();  }  };      var ctrlDeleteItem = **function**(event) {  var itemID, splitID, type, ID;    itemID = event.target.parentNode.parentNode.parentNode.parentNode.id;    **if** (itemID) {    //inc-1  splitID = itemID.split('-');  type = splitID[0];  ID = parseInt(splitID[1]);    // 1. delete the item from the data structure  budgetCtrl.deleteItem(type, ID);    // 2. Delete the item from the UI  UICtrl.deleteListItem(itemID);    // 3. Update and show the new budget  updateBudget();    // 4. Calculate and update percentages  updatePercentages();  }  };      **return** {  **init**: **function**() {  console.log('Application has started.');  UICtrl.displayMonth();  UICtrl.displayBudget({  **budget**: 0,  **totalInc**: 0,  **totalExp**: 0,  **percentage**: -1  });  setupEventListeners();  }  };    })(budgetController, UIController);  controller.init();  **Task 1 Output Screenshot:**  **screencapture-file-D-university-6th-semester-Software-construction-Labs-Lab4-supporting-material-index-html-2021-03-21-15_21_50** |

### Deliverables

Compile a single word document by filling in the solution part and submit this Word file on LMS. This lab grading policy is as follows: The lab is graded between 0 to 10 marks. The submitted solution can get a maximum of 5 marks. At the end of each lab or in the next lab, there will be a viva related to the tasks. The viva has a weightage of 5 marks. Insert the solution/answer in this document. You must show the implementation of the tasks in the designing tool, along with your complete Word document to get your work graded. You must also submit this Word document on the LMS. In case of any problems with submissions on LMS, submit your Lab assignments by emailing it to Mr. Aftab Farooq: [aftab.farooq@seecs.edu.pk](mailto:aftab.farooq@seecs.edu.pk).