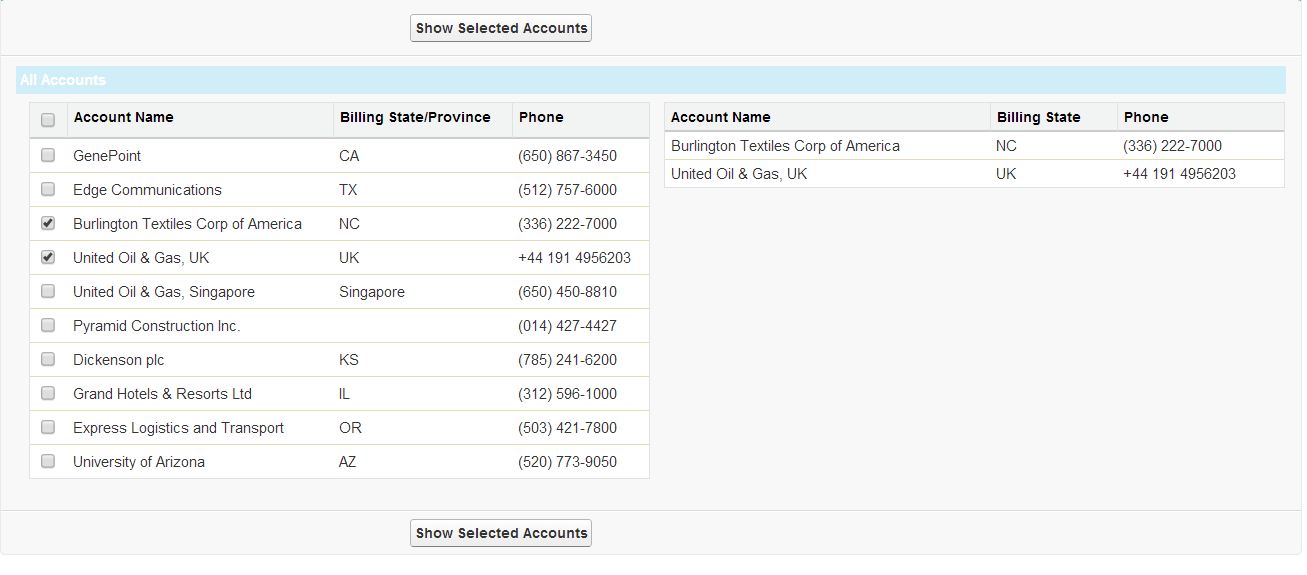
**Wrapper class in Apex Salesforce**

A wrapper or container class is a class, a data structure, or an abstract data type which contains different objects or collection of objects as its members.  
A wrapper class is a custom object defined by programmer wherein he defines the wrapper class properties. Consider a custom object in salesforce, what do you have in it? fields right? different fields of different data types. Similarly wrapper class is a custom class which has different data types or properties as per requirement. We can wrap different objects types or any other types in a wrapper class.

In the Visualforce most important use case is to display a table of records with a check box and then process only the records that are selected.

In the example below, we are displaying list of accounts with checkbox. End user can select account and then click on Show Selected accounts button. Then selected account will be displayed in table.

[Click for Demo](http://demopoint-developer-edition.ap1.force.com/apex/wrapperAccountSelect)

[](http://www.sfdcpoint.com/wp-content/uploads/2014/02/Wrapper-Class-example.jpg)

**Visualforce Code:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42 | <apex:page controller="AccountSelectClassController" sidebar="false">      <script type="text/javascript">          function selectAllCheckboxes(obj,receivedInputID){              var inputCheckBox = document.getElementsByTagName("input");              for(var i=0; i<inputCheckBox.length; i++){                  if(inputCheckBox[i].id.indexOf(receivedInputID)!=-1){                      inputCheckBox[i].checked = obj.checked;                  }              }          }      </script>      <apex:form >          <apex:pageBlock >              <apex:pageBlockButtons >                  <apex:commandButton value="Show Selected Accounts" action="{!processSelected}" rerender="table2"/>              </apex:pageBlockButtons>                <apex:pageblockSection title="All Accounts" collapsible="false" columns="2">                    <apex:pageBlockTable value="{!wrapAccountList}" var="accWrap" id="table" title="All Accounts">                      <apex:column >                          <apex:facet name="header">                              <apex:inputCheckbox onclick="selectAllCheckboxes(this,'inputId')"/>                          </apex:facet>                          <apex:inputCheckbox value="{!accWrap.selected}" id="inputId"/>                      </apex:column>                      <apex:column value="{!accWrap.acc.Name}" />                      <apex:column value="{!accWrap.acc.BillingState}" />                      <apex:column value="{!accWrap.acc.Phone}" />                  </apex:pageBlockTable>                    <apex:pageBlockTable value="{!selectedAccounts}" var="c" id="table2" title="Selected Accounts">                      <apex:column value="{!c.Name}" headerValue="Account Name"/>                      <apex:column value="{!c.BillingState}" headerValue="Billing State"/>                      <apex:column value="{!c.Phone}" headerValue="Phone"/>                  </apex:pageBlockTable>                </apex:pageblockSection>          </apex:pageBlock>      </apex:form>    </apex:page> |

**Apex Class Controller:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37 | public class AccountSelectClassController{        //Our collection of the class/wrapper objects wrapAccount      public List<wrapAccount> wrapAccountList {get; set;}      public List<Account> selectedAccounts{get;set;}        public AccountSelectClassController(){          if(wrapAccountList == null) {              wrapAccountList = new List<wrapAccount>();              for(Account a: [select Id, Name,BillingState, Website, Phone from Account limit 10]) {                  // As each Account is processed we create a new wrapAccount object and add it to the wrapAccountList                  wrapAccountList.add(new wrapAccount(a));              }          }      }        public void processSelected() {      selectedAccounts = new List<Account>();            for(wrapAccount wrapAccountObj : wrapAccountList) {              if(wrapAccountObj.selected == true) {                  selectedAccounts.add(wrapAccountObj.acc);              }          }      }        // This is our wrapper/container class. In this example a wrapper class contains both the standard salesforce object Account and a Boolean value      public class wrapAccount {          public Account acc {get; set;}          public Boolean selected {get; set;}            public wrapAccount(Account a) {              acc = a;              selected = false;          }      }  } |