

```

// -----General Methods-----
private function validator_breed($value)
{
$breed_file = simplexml:load_file($this->breedxml);
$xmlText = $breed_file->asXML();
if(stristr($xmlText, $value) === FALSE)
{
return FALSE;
}
else
{
return TRUE;
}
}
private function change_dog_data($type)
{
if ( file_exists("e8dog_container.php") ) {
require_once("e8dog_container.php"); // use chapter 5 container w exception
handling
} else {
throw new Exception("Dog container file missing or corrupt");
}
$container = new dog_container("dogdata"); // sets the tag name to look for in XML file
$properties_array = array("dogdata"); // not used but must be passed into create_object
$dog_data = $container->create_object($properties_array); // creates dog_data object
$method_array = get_class_methods($dog_data);
$last_position = count($method_array) - 1;
$method_name = $method_array[$last_position];
if (($this->index > -1) && ($type == "Delete"))
{
$record_Array = $this->index;
$dog_data->$method_name("Delete",$record_Array);
}
else if (($this->index == -1) && ($type == "Insert/Update"))
{
$record_Array = array(array('dog_name'=>"$this->dog_name", 'dog_weight'=>"$this->
dog_weight", 'dog_color'=>"$this->dog_color", 'dog_breed'=>"$this->dog_breed"));
$dog_data->$method_name("Insert",$record_Array);
}
else if ($type == "Insert/Update")
{
$record_Array = array($this->index => array('dog_name'=>"$this->dog_name",
'dog_weight'=>"$this->dog_weight", 'dog_color'=>"$this->dog_color",
'dog_breed'=>"$this->dog_breed"));
$dog_data->$method_name("Update",$record_Array);
}
$dog_data = NULL;
}
}

```