8.2.4 Solutions

Solution

1. **In `PALineVehiclesValidator.gs`, add a function to check for at most two vehicles and add the method to the function `doValidate()`.**

```
9  class PALineVehiclesValidator extends PolicyLineValidation<entity.PersonalAutoLine> {

31   override function doValidate() {
32    atLeastOneVehicle()
33    atMostNumVehicles(2)
34    allGaragesInSameState()
35    vinIsUniqueInPeriod()
36    validateEachVehicle()
37   }

116  public function atMostNumVehicles(num: int) {
117   Context.addToVisited(this, "atMostTwoVehicles()")
118   if (paLine.Vehicles.Count > num and Context.isAtLeast(ValidationLevel.TC_DEFAULT)){
119    var msg = DisplayKey.get("Ext.Validator.AtMostTwoVehicles", num)
120    if ( Context.isAtLeast(ValidationLevel.TC_QUICKQUOTABLE))
121     Result.addError(paLine, ValidationLevel.TC_QUICKQUOTABLE, msg, VEHICLES_WIZARD_STEP)
122    else
123     Result.addWarning(paLine, ValidationLevel.TC_DEFAULT, msg, VEHICLES_WIZARD_STEP)
124   }
125  }
126
127 }
```

2. **Create the display key.**

   Press Alt-Enter on the display key used in the method above and create the new display key.

   ```
   Ext.Validator.AtMostTwoVehicles = A Personal Auto policy can have at most {0}
   vehicles.
   ```

3. **Stop and restart server using Run -> Debug Server or simply Reload Changed Classes in Studio.**

8.3 References

Review

Creating a new validation class:

New class should extend `PCValidationBase` or `PCValidation`. Use an existing validation class as a reference.

For new entities:

- Create new validation class for entity you want to validate or
- Create new validation class when you create a new LOB

For existing entities:

- If a validation class for the entity exists add validation check methods to that class
- If the validation class does not exist, then create new validation class

Review

Steps to add a validation check:

1. **Edit/add appropriate validation class.**

   First you will have to decide whether this validation can be added to an existing class or a new validation class needs to be created. If, on the other hand, you needed to add a validation check for a new entity that didn't already have any validation checks and no validation checks exist for it, you would need to create a new validation class to validate that specific entity.

2. **Add the validation check method.**

   You will need to restart the server because of display key addition.

3. **Call method from `validationImpl()`.**

   The `validateImpl` method of any class calls all the methods in a logical sequence.

   Note: In case where you had to create a new validation class then you will also have to create a `validateImpl()` method and then add the method call statement to the method. For the example, we are creating a validation check for a personal vehicle and adding the method call statement to the `validateImpl()` method of the `PersonalVehicleValidation` class.

   If you are extending `PolicyLineValidation` class, you will have to create a `doValidate()` method and call the methods that validate a single issue in a logical order.

4. **Invoke the method.**

   The validation class methods must be invoked explicitly and are not called automatically. If it is already in the validation chaining path, you do not have to invoke it.

5. **Verify the result in the UI.**

Review

Invoking class-based validation:

Class-based validation must be explicitly invoked through code, job processes, wizard steps, workflow steps, integration plug-ins, or from any Gosu code.

To invoke validation from a particular PolicyCenter location:

- Wizard steps - use beforeSave attribute for that step.
- Pop-ups - use beforeCommit attribute for that pop-up.

Review

Validation Chaining is the process of one validation class calling another validation class to perform additional validation checks.

- Call `validate()` method which in turn calls other methods.
- Invoke validate method on another validation class (may have to loop through a set of objects).
- Classes chain to validations of entities they hold.
- Refer to `PolicyPeriodValidation` class for examples of validation chaining. It calls other validation classes, such as:

  `PolicyContactRoleValidation`,

  `PolicyContactRoleForSameContactValidation`,

  `PolicyLocationValidation`, `AnswerValidation` and

  `PolicyLineValidation`.

Stop