# KillTest

Higher Quality, Better Service!

# Q&A

http://www.killtest.com

We offer free update service for one year.

**Exam** : **70-548(VB)**

**Title** : PRO:Design & Develop Wdws-Based Appl by Using MS.NET Frmwk

**Version** : DEMO

1. You create Microsoft Windows-based applications. You create a banking application that will be used by the account managers of the bank.

You identify a method to simulate the deposit functionality of a savings account. The method will calculate the final balance when monthly deposit, number of months, and quarterly rate are given. The application requirements state that the following criteria must be used to calculate the balance amount:

·Apply the quarterly interest rate to the balance amount of the account every three months.

·Apply the quarterly interest rate before the monthly deposit is calculated for the third month.

You translate the outlined specification into pseudo code. You write the following lines of code. (Line numbers are included for reference only.)

```
Method
Public Shared Function SimulateSavings() As Decimal
Input parameters
Dim months As Integer
Dim monthlyPayment As Decimal
Dim quarterlyRate As Decimal
Pseudo code
01 Declare balance variable, initialize it to zero
02
03 Return balance
```

You need to insert the appropriate code in line 02.

Which code segment should you insert?

A. 01 Declare integer variable, x
   02 For x=1 to months/3
   2.1 balance = balance + 3 * monthlyPayment
   2.2 balance = (1 + quarterlyRate) * balance

B. 01 Declare integer variable, x
   02 For x=1 to months/3
   2.1 balance = balance + 2 * monthlyPayment

2.2 balance = (1 + quarterlyRate) * balance

2.3 balance = balance + monthlyPayment

C. 01 Declare integer variable, x

02 For x=1 to months

2.1 balance = balance + monthlyPayment

2.2 if x Mod 3 is 0 then balance = (1 + quarterlyRate) * balance

D. 01 Declare integer variable, x

02 For x=1 to months

2.1 if x Mod 3 is 0 then balance = (1 + quarterlyRate) * balance

2.2 balance = balance + monthlyPayment

**Answer:** D

2. You create Microsoft Windows-based applications. You are creating a component that will be used by several client applications.

The component contains the following code segment. (Line numbers are included for reference only.)

```
01  Namespace MyComponent
02      Public Class Account
03          Private mAccountNo As String
04          Private mBalance As Decimal
05          Public Sub New(ByVal AcctNo As String)
06              ...
07          End Sub
08          Public Sub Withdraw(ByVal Amount As Decimal)
09              ...
10          End Sub
11          Public Sub Deposit(ByVal Amount As Decimal)
12              ...
13          End Sub
14      End Class
15      Public Class SavingsAccount
16          Inherits Account
17          Public Sub New(ByVal AcctNo As String)
18              MyBase.New(AcctNo)
19              ...
```

```
20          End Sub
21          Public Sub ApplyInterestRate(ByVal Amount As Decimal)
22              ...
23          End Sub
24      End Class
25  End Namespace
```

You need to redesign the Account class and the SavingsAccount class to meet the following requirements:

·Developers must not be able to instantiate the Account class from client applications.

·Developers must not be able to extend the functionality of the SavingsAccount class.

·Developers must be able to instantiate the SavingsAccount class from client applications.

Which two actions should you perform? (Each correct answer presents part of the solution. Choose two).

A. Implement only Private constructors for the Account class.

B. Implement only Private constructors for the SavingsAccount class.

C. Implement only Friend constructors for the Account class.

D. Implement the SavingsAccount class as a MustInherit public class.

E. Implement the SavingsAccount class as a concrete non-inheritable class.

**Answer:** C AND E

3. You create Microsoft Windows-based applications. You are designing an inventory management solution for a warehouse. The solution must address the following requirements:

·Access inventory data in a Microsoft SQL Server 2005 database.

·Generate XML documents representing purchase orders based on an XML schema provided by a trading partner.

·Use the minimum amount of code possible.

·Use the minimum amount of I/O operations possible.

You need to develop the data handling capabilities of the solution to meet the requirements.

Which three data handling mechanisms should you select? (Each correct answer presents part of the solution. Choose three.)

A. Use an XmlReader object to retrieve inventory data from the database and populate a DataSet object.

B. Use a DataAdapter object to retrieve inventory data from the database and populate a DataSet object.

C. Use methods from the DataSet class to generate a new XML file that contains data to be used to generate a purchase order.

D. Use methods from the DataSet class to generate a new XmlDataDocument object that contains data to

be used to generate a purchase order.

E. Use an XslCompiledTransform object to generate the purchase order XML file.

F. Use an XmlWriter object to generate the purchase order XML file.

**Answer:** B AND D AND E

4. You create Microsoft Windows-based applications. You create an application that accesses data on a Microsoft SQL Server 2005 database. You write the following code segment. (Line numbers are included for reference only.)

```
01  Private Sub LoadData()
02
03      cn.Open()
04      daProducts.Fill(ds)
05      daCategories.Fill(ds)
06      cn.Close()
07
08  End Sub
```

The cn variable points to a SqlConnection object. The SqlConnection object will be opened almost every time this code segment executes.

You need to complete this code segment to ensure that the application continues to run even if the SqlConnection object is open. You also need to ensure that the performance remains unaffected.

What should you do?

A. Add a Try block on line 02 along with a matching Catch block beginning on line 07 to handle the possible exception.

B. Add a Try block on line 02 along with a matching Finally block beginning on line 07 to handle the possible exception.

C. Replace line 03 with the following code.

```
If Not (cn.State = ConnectionState.Open) Then
    cn.Open()
End If
```

D. Replace line 03 with the following code.

```
If Not (cn.State = ConnectionState.Closed) Then
```

```
        cn.Open()
    End If
```

**Answer:** C

5. You create Microsoft Windows-based applications. You are creating a method. Your applications will call the method multiple times. You write the following lines of code for the method.

```
01  Public Function BuildSQL(ByVal strFields As String, ByVal strTable As String, ByVal strFilterId As String)
As String
02      Dim sqlInstruction As String = "SELECT "
03      sqlInstruction += strFields
04      sqlInstruction += " FROM "
05      sqlInstruction += strTable
06      sqlInstruction += " WHERE id ="
07      sqlInstruction += strFilterid
08      Return sqlInstruction
09  End Function
```

The method generates performance issues.

You need to minimize the performance issues that the multiple string concatenations generate.

What should you do?

A. Use a single complex string concatenation.

B. Use an array of strings.

C. Use an ArrayList object.

D. Use a StringBuilder object.

**Answer:** D

6. You create Microsoft Windows-based applications. You are creating an application that will connect to a Microsoft SQL Server 2005 database. You write the following code segment for a method contained in the application. (Line numbers are included for reference only.)

```
01  Private cn As SqlConnection
02  Private Function getConn() As Object
03      cn = New SqlConnection("data source = localhost;initial Catalog = Accounting;integrated security = true")
04      Return cn
05  End Function
```

In the production environment, the database will be stored by a server on the network.

You need to eliminate the requirement to recompile the application when you deploy it to the production environment. You want to achieve this by using minimum amount of programming effort.

What should you do?

A. Create an application configuration file to store the connection string. Change the code to read the connection string from the configuration file.

B. Create an XML file in the application folder to store the connection string. Change the code to use an XMLReader object to connect to a file stream and read the connection string.

C. Create a component that returns the connection string. Change the code to use the component to get the connection string.

D. Create a text file to store the connection string. Change the code to use a TextReader object to connect to a file stream and read the connection string.

**Answer:** A

7. You create Microsoft Windows-based applications. You are reviewing code for an application that is created for a bank.You find that a Microsoft Windows Form includes the following code segment.

```
Partial Public Class ATMDeposit
    Inherits Form
    Private account As BankAccount
    Public Sub New()
        InitializeComponent()
    End Sub
    Private Sub New_Load(ByVal sender As Object, ByVal e As EventArgs)
        account = New BankAccount()
    End Sub
    Private Sub cmdDeposit_Click(ByVal sender As Object, ByVal e As EventArgs)
    account.Deposit(Decimal.Parse(txtAmount.Text))
    End Sub
End Class
```

You analyze the code segment and find that the form handles no other events.

You need to suggest changes to improve reliability.

Which two actions should you perform? (Each correct answer presents part of the solution. Choose two.)

A. Add an event handler for the TextChanged event for the txtAmount textbox to validate the data typed by the user.

B. Add an event handler for the Validating event for the txtAmount textbox to validate the data typed by the user.

C. Add a Try...Catch block to the cmdDeposit_Click method.

D. Add a Try...Catch block to the ATMDeposit_Load method.

E. Add a Try...Catch block to the ATMDeposit constructor.

**Answer:** B AND C

8. You create Microsoft Windows-based applications.

You receive the following code segment to review. (Line numbers are included for reference only.)

```
01  Partial Public Class frmReceivables
02      Inherits Form
03      Private ds As DataSet
04      Public Sub New()
05          InitializeComponent()
06      End Sub
07      Private Sub New(ByVal sender As Object, ByVal e As EventArgs)
08          Dim cn As SqlConnection = New SqlConnection(strConnectionString)
09          Dim daInvoices As SqlDataAdapter = New SqlDataAdapter("SELECT * FROM Invoices", cn)
10          Dim daCustomers As SqlDataAdapter = New SqlDataAdapter("SELECT * FROM Customers", cn)
11          ds = New DataSet("Receivables")
12          daInvoices.Fill(ds)
13          daCustomers.Fill(ds)
14      End Sub
15  End Class
```

The strConnectionString variable is pre-populated from the application configuration file. Query statements will remain unchanged throughout the life cycle of the application. Connection pooling is not being used.

This code segment accesses a Microsoft SQL Server 2000 database. The ds dataset is bound to a data grid view so that users can view and update data in the database. The code currently compiles correctly

and works as intended.

You need to enhance performance and reliability for this code.

Which two actions should you recommend? (Each correct answer presents part of the solution. Choose two.)

A. Use an ODBC DSN instead of a connection string.

B. Use OleDbDataAdapter objects instead of SqlDataAdapter objects to populate the dataset.

C. Add a line of code before line 12 to open the database connection.

D. Add a Try...Catch block and close the connection in the catch block.

E. Add a Try...Catch...Finally block and close the connection in the Finally block.

**Answer:** C AND E

9. You create Microsoft Windows-based applications. You review code for an application that is developed for a bank. You need to test a method named Deposit in one of the application components. The following code segment represents the Deposit method. (Line numbers are included for reference only.)

```
01  Public Sub Deposit(ByVal amount As Decimal)
03      If Not (amount > 0) Then
04          Throw New Exception("Invalid deposit amountNot ")
05      Else
06          Me.balance += amount
07      End If
08  End Sub
```

You use the Microsoft Visual Studio 2005 test feature to automatically generate the following unit test. (Line numbers are included for reference only.)

```
01  <TestMethod()> _
02  Public Sub DepositTest()
03      Dim target As BankAccount = New BankAccount() 'balance will be ZERO
04      Dim amount As Decimal = 100
05      target.Deposit(amount)
06      Assert.Inconclusive("A method that does not return a value cannot be verified.")
07  End Sub
```

You need to change the test method to return a conclusive result.

Which line of code should you use to replace the code on line 06?

A. Assert.AreEqual(100,target.Balance)

B. Assert.IsTrue(target.Balance <> 100)

C. Debug.Assert(target.Balance = 100,passed)

D. Debug.Assert(target.Balance = 100,failed)

**Answer:** A

10. You create Microsoft Windows-based applications. You are testing a component named BankAccount. You write the following code segment for the BankAccount component. (Line numbers are included for reference only.)

```
01  Public Class BankAccount
02      Private _balance As Decimal
03      Public Property Balance() As Decimal
04          Get
05              Return Me._balance
06          End Get
07          Set(ByVal value As Decimal)
08              Me._balance = value
09          End Set
10      End Property
11      Public Sub Withdraw(ByVal amount As Decimal)
12          If Not (amount > 0) Then
13              Throw New Exception("Invalid withdraw amount!")
14          ElseIf (amount > Me.balance) Then
15              Throw New Exception("Insufficient balance")
16          Else
17              Me.balance = (Me.balance - amount)
18          End If
19      End Sub
20      Public Sub Deposit(ByVal amount As Decimal)
21          If Not (amount > 0) Then
22              Throw New Exception("Invalid deposit amount!")
```

```
23          Else
24              Me.balance = (Me.balance + amount)
25          End If
26      End Sub
27  End Class
```

The test project executes a valid withdraw operation and a valid deposit operation. It also verifies the account balance.

Your companys check-in policy requires that the primary code path is tested before check in. Full testing will be completed later.

You need to establish the lowest acceptable code coverage metric.

What should you test?

A. Test all methods.

B. Do not test exceptions. Test all other methods.

C. Test all code, including variable declarations.

D. Do not test accessor methods. Test all other methods and exceptions.

**Answer:** B