PassTest

Bessere Qualität , bessere Dienstleistungen!

Q&A

Exam : **70-504(C#)**

Title : TS: Microsoft .NET Framework 3.5 -C Windows Workflow Foundation

Version : Demo

1. You are writing a sequential console workflow that consists of a delay activity and a code activity, as shown in the exhibit. (Click the Exhibit button for the sequential console workflow image)

In the execution code of the second activity, you try to modify the workflow as follows:

```
private void codeActivity_ExecuteCode(object sender, EventArgs e)

{

CodeActivity delay = sender as CodeActivity;

Console.WriteLine(delay.Name);

WorkflowChanges workflowChanges = new WorkflowChanges(this);

CodeActivity codeActivity = new CodeActivity();

codeActivity.Name = "codeActivity2";

codeActivity.ExecuteCode += new EventHandler(codeActivity2_ExecuteCode);

workflowChanges.TransientWorkflow.Activities.Add(codeActivity);

this.ApplyWorkflowChanges(workflowChanges);

}

private void codeActivity2_ExecuteCode(object sender, EventArgs e)

{

CodeActivity codeActivity = sender as CodeActivity;

Console.WriteLine(codeActivity.Name);

Console.ReadLine();

}
```

You also have set the modifiability of the workflow to a code condition that is set to the following function:

```
private void UpdateCondition(object sender, ConditionalEventArgs e)

{

if (TimeSpan.Compare(this.delayActivity.TimeoutDuration, new TimeSpan(0, 0, 5)) > 0) {

e.Result = false;

}

else {

e.Result = true;

}

}
```

Which code segment should you use to handle the exception?

A. workflowChanges.TransientWorkflow.Activities.Add(codeActivity);

try {

this.ApplyWorkflowChanges(workflowChanges);

}

catch (ArgumentOutOfRangeException ex) {

Console.WriteLine(ex.GetType().ToString());

Console.ReadLine();

}

B. workflowChanges.TransientWorkflow.Activities.Add(codeActivity);

try {

this.ApplyWorkflowChanges(workflowChanges);

}

catch (InvalidProgramException ex) {

Console.WriteLine(ex.GetType().ToString());

Console.ReadLine();

}

C. workflowChanges.TransientWorkflow.Activities.Add(codeActivity);

try {

this.ApplyWorkflowChanges(workflowChanges);

}

catch (InvalidOperationException ex) {

Console.WriteLine(ex.GetType().ToString());

Console.ReadLine();

}

D. workflowChanges.TransientWorkflow.Activities.Add(codeActivity);

try {

this.ApplyWorkflowChanges(workflowChanges);

}

catch (OverflowException ex) {

Console.WriteLine(ex.GetType().ToString());

Console.ReadLine();

}

Answer: C

2. A windows application invokes a workflow that performs complex calculations. When the calculation is done, the workflow returns the calculated value to the windows application, which must display the calculated value in a textbox control, as follows:

public void ShowData(string value)

{

mytextBox.Text = value

}

You need to ensure the workflow updates the value in the textbox control.

What should you do?

A. Have the workflow directly call the ShowData() method.

B. Have the workflow call a method in an interface decorated with an

[ExternalDataExchange()] attribute that invokes the ShowData() method directly

C. Set the workflow to call a method in an interface decorated with an [ExternalDataEventArg()] attribute that invokes the ShowData() method directly

D. Set the workflow to call a method in a interface decorated with an [ExternalDataExchange()] attribute that invokes the ShowData() method indirectly through a delegate if the this.InvokeRequired condition is true.

Answer: D

3. You create an application in which users design simple sequential workflows. The designs are stored as XOML in a SQL database. You need to start one of these sequential workflows from within your own workflow.

What should you do?

A. Include a custom activity with a code segment that includes an invocation to the CreateWorkflow method of the WorkflowRuntime class and then starts the workflow. The signature of the CreateWorkflow

method invoked uses only the workflowType parameter.

B. Include a custom activity with a code segment that includes an invocation to the CreateWorkflow method of WorkflowRuntime class and then starts the workflow. The signature of the CreateWorkflow method invoked uses only the XmlReader and workflowDefinitionReader parameters.

C. Include a custom activity with a code segment that includes an invocation to the CreateWorkflow method of the WorkflowRuntime class and then starts the workflow. The signature of the CreateWorkflow method invoked uses only the workflowType, Dictionary<string,Object> namedArgumentValues, and Guid instanceId parameters.

D. Include and configure an InvokeWorkflow activity

Answer: B

4. You use a built-in tracking service to track specific workflow parameters.

You need to check whether the workflow parameters have been stored in the tracking database.

What should you do? (Each correct answer presents part of a solution. Choose two.)

A. Display the contents of the WorkflowInstance table of the tracking database.

B. Include the SqlTrackingQuery class in a code segment to retrieve tracked workflows and SqlTrackingWorkflowInstance class to inspect them.

C. Use the ActivityTrackingLocation class to determine if the value has been set to a database.

D. Display the contents of the TrackingDataItem table of the tracking database.

Answer: BD

5. You create a Windows Workflow Foundation application by using Microsoft .NET Framework 3.5. You use the state machine workflow in the application.

You plan to implement a mechanism that allows a host application to query a state machine workflow instance that is currently executing.

You write the following code segment. (Line numbers are included for reference only.)

01 WorkflowRuntime runtime = new WorkflowRuntime();

02 WorkflowInstance instance =

03   runtime.CreateWorkflow(typeof(Workflow1));

04 instance.Start();

05

You need to identify the current state of the workflow.

Which code segment should you insert at line 05?

A. string currentstate = instance.GetWorkflowDefinition().ToString();

B. string currentstate = instance.GetWorkflowDefinition().ExecutionStatus.ToString();

C.　　StateMachineWorkflowInstance　　smwi　　=　　new　　StateMachineWorkflowInstance(runtime, instance.InstanceId);

string currentstate = smwi.StateHistory[0];

D.　　StateMachineWorkflowInstance　　smwi　　=　　new　　StateMachineWorkflowInstance(runtime, instance.InstanceId);

string currentstate = smwi.CurrentStateName;

Answer: D

6. You create a Windows Workflow Foundation application by using Microsoft .NET Framework 3.5. The application contains a state workflow.

You write the following code segment.

WorkflowRuntime runtime = new WorkflowRuntime();

WorkflowInstance instance = runtime.CreateWorkflow(

  typeof(DynamicUpdateWorkflow));

instance.Start();

StateMachineWorkflowInstance smwi =

  new StateMachineWorkflowInstance(runtime,

  instance.InstanceId);

A dependency property named Status is defined in this workflow.

The value of a variable named amount is used to set the state of the workflow.

You need to ensure that the host application changes the state of the workflow on the basis of the value of the amount variable.

What are the two possible code segments that you can use to achieve this goal? (Each correct answer presents a complete solution. Choose two.)

A. if (amount >= 1000)

smwi.SetState("HighValueState");

else

 smwi.SetState("LowValueState");

B. if (amount >= 1000)

 smwi.StateMachineWorkflow.SetValue

 (DynamicUpdateWorkflow.StatusProperty,"HighValueState")

else

 smwi.StateMachineWorkflow.SetValue

 (DynamicUpdateWorkflow.StatusProperty, "LowValueState");

C. if (amount >= 1000)

 instance.GetWorkflowDefinition().SetValue

 (DynamicUpdateWorkflow.StatusProperty,"HighValueState");

else

 instance.GetWorkflowDefinition().SetValue

 (DynamicUpdateWorkflow.StatusProperty,"LowValueState");

D. if (amount >= 1000)

{

 StateActivity high = (StateActivity)

 smwi.StateMachineWorkflow.Activities["HighValueState"];

 smwi.SetState(high);

}

else

{

 StateActivity low = (StateActivity)

 smwi.StateMachineWorkflow.Activities["LowValueState"];

 smwi.SetState(low);

}

Answer: AD

7. You create a Windows Workflow Foundation application by using Microsoft .NET Framework 3.5.

The application uses a code-only workflow. You define the workflow logic by programmatically adding activities.

You write the following code segment.

```
public sealed partial class Workflow1
  : SequentialWorkflowActivity {
  ...
  }
```

You also write the following code segment to add a DelayActivity activity to a workflow.

```
this.CanModifyActivities = true;
DelayActivity delay = new DelayActivity();
delay.TimeoutDuration = new TimeSpan(0, 0, 5);
this.Activities.Add(delay);
this.CanModifyActivities = false;
```

You need to insert the DelayActivity activity code segment at the correct location in the workflow.

What should you do?

A. Place the following code segment in the class constructor.

```
public Workflow1() {
   ...
}
```

B. Use the following code segment to override the InitializeProperties method.

```
protected override void InitializeProperties() {
   ...
}
```

C. Use the following code segment to override the Execute method.

```
protected override ActivityExecutionStatus Execute(
   ActivityExecutionContext executionContext) {
   ...
}
```

D. Override the Initialize method in the following manner.

```
protected override void Initialize(
```

?IServiceProvider provider) {

   base.Initialize(activity);

   ...

}

Answer: A

8. You create a workflow host application by using Microsoft .NET Framework 3.5. You use Windows

Workflow Foundation to create the application.

You need to configure the workflow runtime to ensure that all the workflow instances run asynchronously.

Which code segment should you use?

A. WorkflowRuntime runtime = new WorkflowRuntime();

runtime.StartRuntime();

WorkflowInstance instance =

  runtime.CreateWorkflow(typeof(CustomerWorkflow));

instance.Start();

B. WorkflowRuntime runtime = new WorkflowRuntime();

runtime.StartRuntime();

ManualWorkflowSchedulerService scheduler =

  runtime.GetService<ManualWorkflowSchedulerService>();

WorkflowInstance instance =

  runtime.CreateWorkflow(typeof(CustomerWorkflow));

scheduler.RunWorkflow(instance.InstanceId);

C. WorkflowRuntime runtime = new WorkflowRuntime();

ManualWorkflowSchedulerService scheduler =

  new ManualWorkflowSchedulerService();

runtime.AddService(scheduler);

runtime.StartRuntime();

WorkflowInstance instance =

  runtime.CreateWorkflow(typeof(CustomerWorkflow));

instance.Start();

D. WorkflowRuntime runtime = new WorkflowRuntime();

runtime.StartRuntime();

DefaultWorkflowSchedulerService scheduler =

  new DefaultWorkflowSchedulerService();

runtime.AddService(scheduler);

WorkflowInstance instance =

  runtime.CreateWorkflow(typeof(CustomerWorkflow));

instance.Start();

Answer: A

9. You create a Windows Workflow Foundation application by using Microsoft .NET Framework 3.5.

The application uses a workflow host that runs in a console application. The workflow runtime uses the

default threading behavior.

You plan to attach the workflow runtime to two events. You write the following code segment to handle the

events.

static void runtime_WorkflowCompleted(

  object sender, WorkflowCompletedEventArgs e) {

    waitHandle.Set();

}

You also define the class fields in the following manner.

private static AutoResetEvent waitHandle;

private static WorkflowRuntime runtime =

  new WorkflowRuntime();

You write the following code segment in the Main method of the console application.

01 runtime.WorkflowCompleted +=

02   new EventHandler<WorkflowCompletedEventArgs>(

03 runtime_WorkflowCompleted);

04 WorkflowInstance wi = runtime.CreateWorkflow(typeof(WF));

05

You need to ensure that the console application does not exit before the workflow instance completes

execution.

Which code segment should you insert at line 05?

A. waitHandle = new AutoResetEvent(false);

waitHandle.WaitOne();

wi.Start();

B. waitHandle = new AutoResetEvent(true);

wi.Start();

waitHandle.WaitOne();

C. waitHandle = new AutoResetEvent(false);

wi.Start();

waitHandle.WaitOne();

D. waitHandle = new AutoResetEvent(true);

waitHandle.WaitOne();

wi.Start();

Answer: C

10. You create a Windows Workflow Foundation application by using Microsoft .NET Framework 3.5. The application uses a sequential workflow.

The workflow is implemented in a class named ProcessOrders. The workflow contains a dependency property named EmployeeID.

You need to ensure that the EmployeeID property is assigned a value when the host application tries to create a new workflow instance.

Which code segment should you use?

A. WorkflowRuntime runtime = new WorkflowRuntime();

ProcessOrders processOrders = new ProcessOrders();

processOrders.EmployeeID = "NBK";

WorkflowInstance instance =

  runtime.CreateWorkflow(typeof(ProcessOrders));

B. WorkflowRuntime runtime = new WorkflowRuntime();

ProcessOrders processOrders = new ProcessOrders ();

processOrders.SetValue(ProcessOrders.EmployeeIDProperty ,

? "NBK");

WorkflowInstance instance =

  runtime.CreateWorkflow(typeof(ProcessOrders));

C. WorkflowRuntime runtime = new WorkflowRuntime();

Dictionary<string, object> dict =

  new Dictionary<string,object>();

dict.Add("EmployeeID", "NBK");

WorkflowInstance instance =

  runtime.CreateWorkflow(typeof(ProcessOrders), dict);

D. WorkflowRuntime runtime = new WorkflowRuntime();

Dictionary<string, object> dict =

  new Dictionary<string,object>();

dict.Add("EmployeeIDProperty", "NBK");

WorkflowInstance instance =

  runtime.CreateWorkflow(typeof(ProcessOrders), dict);

Answer: C

11. You create a Windows Workflow Foundation application by using Microsoft .NET Framework 3.5. The

application contains a sequential workflow.

A new business policy requires the workflow to execute all the activities except the POCreated activity.

You write the following code segment in the host application. (Line numbers are included for reference

only.)

01 WorkflowRuntime runtime = new WorkflowRuntime();

02 WorkflowInstance workflowInstance =

03 runtime.GetWorkflow(instanceId);

04 Activity wRoot = workflowInstance.GetWorkflowDefinition();

05

The variable instanceId contains the ID of the workflow.

You need to ensure that the new business policy is applied.

Which code segment should you insert at line 05?

A. WorkflowChanges changes = new WorkflowChanges(wRoot);

Activity POCreated =

?workflowInstance.GetWorkflowDefinition();

changes.TransientWorkflow.Activities.Remove(POCreated);

workflowInstance.Load();

B. WorkflowChanges changes = new WorkflowChanges(wRoot);

Activity POCreated =

 changes.TransientWorkflow.Activities["POCreated"];

changes.TransientWorkflow.Activities.Remove(POCreated);

workflowInstance.ApplyWorkflowChanges(changes);

C. workflowInstance.Unload();

WorkflowChanges changes = new WorkflowChanges(wRoot);

Activity POCreated =

 changes.TransientWorkflow.Activities["POCreated"];

changes.TransientWorkflow.Activities.Remove(POCreated);

workflowInstance.Load();

D. workflowInstance.Suspend("removing activity");

WorkflowChanges changes = new WorkflowChanges(wRoot);

Activity POCreated =

 changes.TransientWorkflow.Activities["POCreated"];

changes.TransientWorkflow.Activities.Remove(POCreated);

workflowInstance.Load();

Answer: B

12. You create a Windows Workflow Foundation application by using Microsoft .NET Framework 3.5. The

application uses a sequential workflow as shown in the following exhibit. (Click the Exhibit button for the

sequential workflow image.)
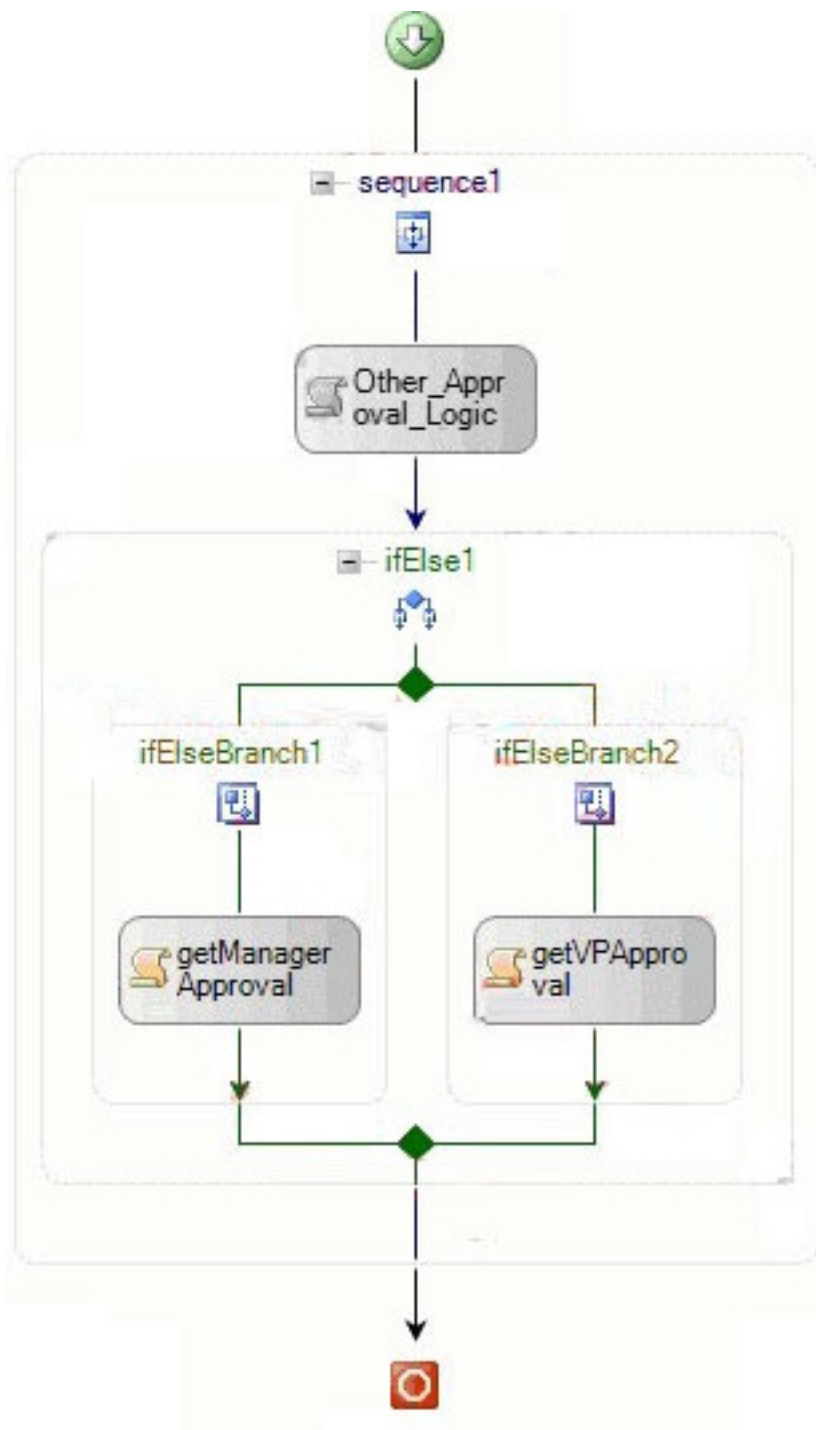
The workflow implements an if condition as shown in the following exhibit. (Click the Exhibit button for the

if condition image.)
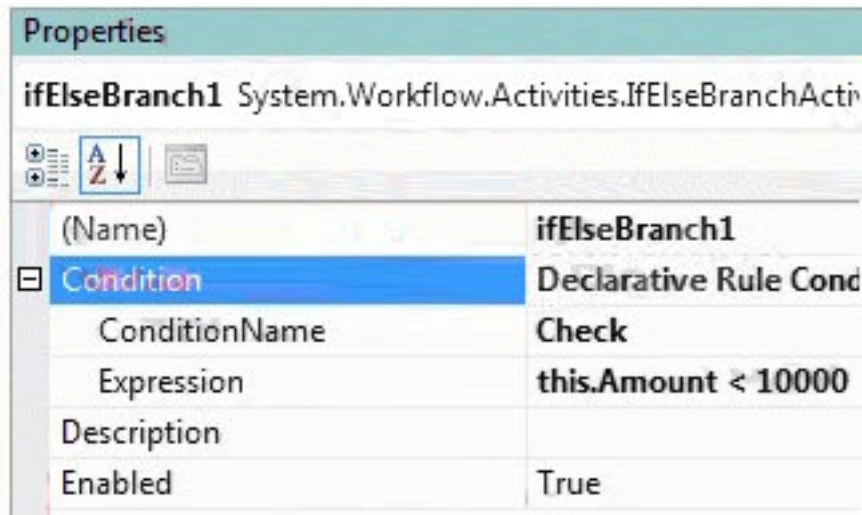
A new business policy requires the application to ascertain whether the amount is less than 15,000 instead of the current default.

You write the following code segment in the host application. (Line numbers are included for reference only.)

01 Int32 newAmount = 15000;

02 WorkflowChanges workflowchanges =

03   new WorkflowChanges(instance.GetWorkflowDefinition());

04 CompositeActivity transient =

05 workflowchanges.TransientWorkflow;

06 RuleDefinitions ruleDefinitions = (RuleDefinitions)

07   transient.GetValue(RuleDefinitions.RuleDefinitionsProperty);

08 RuleConditionCollection conditions =

09   ruleDefinitions.Conditions;

10

11 (condition1.Expression as CodeBinaryOperatorExpression).Right

12 ?= new CodePrimitiveExpression(newAmount);

    instance.ApplyWorkflowChanges(workflowchanges);

You need to build a host application that modifies the condition according to the business requirement in workflow instances that are currently executing.

Which code segment should you insert at line 10?

A. RuleExpressionCondition condition1 =

(RuleExpressionCondition)conditions["Check"];

B. RuleExpressionCondition condition1 =

 (RuleExpressionCondition)conditions["ifElseBranch1"];

C. RuleExpressionCondition condition1 =

 (RuleExpressionCondition)conditions["ifElseBranch2"];

D. RuleExpressionCondition condition1 =

 (RuleExpressionCondition)conditions["Declarative Rule Condition"];

Answer: A

13. You create a Windows Workflow Foundation application by using Microsoft .NET Framework 3.5. The

application uses a sequential workflow that meets the following requirements:

h The workflow application calls an external method to simultaneously notify two users about a task.

h The host application raises the TaskCompleted event after each user completes a task.

The workflow has two HandleExternalEvent activities that handle the TaskCompleted event in parallel for

two users.

You need to ensure that each TaskCompleted event is handled by the HandleExternalEvent activity

mapped to the respective user.

Which code segment should you use?

A. [ExternalDataExchange]

public interface ITaskService

{

   void CreateTask(string taskId, string assignee, string text);

   event EventHandler<TaskEventArgs> TaskCompleted;

}

B. [ExternalDataExchange]

public interface ITaskService

{

   void CreateTask(string taskId, string assignee, string text);

   [CorrelationAlias("taskId", "e.Id")]

   [CorrelationInitializer()]

    event EventHandler<TaskEventArgs> TaskCompleted;

}

C. [ExternalDataExchange]

public interface ITaskService

{

    [CorrelationInitializer]

    void CreateTask(string taskId, string assignee, string text);

    [CorrelationAlias("taskId", "e.Id")]

    event EventHandler<TaskEventArgs> TaskCompleted;

}

D. [ExternalDataExchange]

[CorrelationParameter("taskId")]

public interface ITaskService

{

    [CorrelationInitializer]

    void CreateTask(string taskId, string assignee, string text);

    [CorrelationAlias("taskId", "e.Id")]

    event EventHandler<TaskEventArgs> TaskCompleted;

}

Answer: D

14. You are creating a Windows Workflow Foundation workflow by using Microsoft .NET Framework 3.5.

The workflow host must receive data from workflow instances by using a communication service named

CustomerDataExchange.

You need to configure the workflow runtime services to enable communication between the host and the

workflow instances.

Which code segment should you use?

A. WorkflowRuntime runtime = new WorkflowRuntime();

CustomerDataExchange cde = new CustomerDataExchange();

runtime.AddService(cde);

B. WorkflowRuntime runtime = new WorkflowRuntime();

CustomerDataExchange cde = new CustomerDataExchange();

ExternalDataExchangeService dataService =

  new ExternalDataExchangeService();

dataService.AddService(cde);

C. WorkflowRuntime runtime = new WorkflowRuntime();

ExternalDataExchangeService dataService =

  new ExternalDataExchangeService();

CustomerDataExchange cde = new CustomerDataExchange();

dataService.AddService(cde);

runtime.AddService(dataService);

D. WorkflowRuntime runtime = new WorkflowRuntime();

ExternalDataExchangeService dataService =

  new ExternalDataExchangeService();

runtime.AddService(dataService);

CustomerDataExchange cde = new CustomerDataExchange();

dataService.AddService(cde);

Answer: D

15. You are creating a workflow application by using Microsoft .NET Framework 3.5. The application uses

local communication.

The workflow host must receive data from an existing workflow instance.

You need to define the communication between the workflow host and the workflow instance.

Which code segment should you use?

A. [ExternalDataExchange]

public interface ICommunicationService {

  void CallHost(string someData);

}

B. [ExternalDataExchange]

public interface ICommunicationService {

event EventHandler<ExternalDataEventArgs> CallHost;

}

C. public class CommunicationService

? : ExternalDataExchangeService {

public event EventHandler<ExternalDataEventArgs> CallHost;

}

D. public class CommunicationService

? : ExternalDataExchangeService {

public void CallHost(string someData) {

base.AddService(this);

}

}

Answer: A