

Examination #1

Name: _____ **Instructor's Solution** _____ SUID: _____

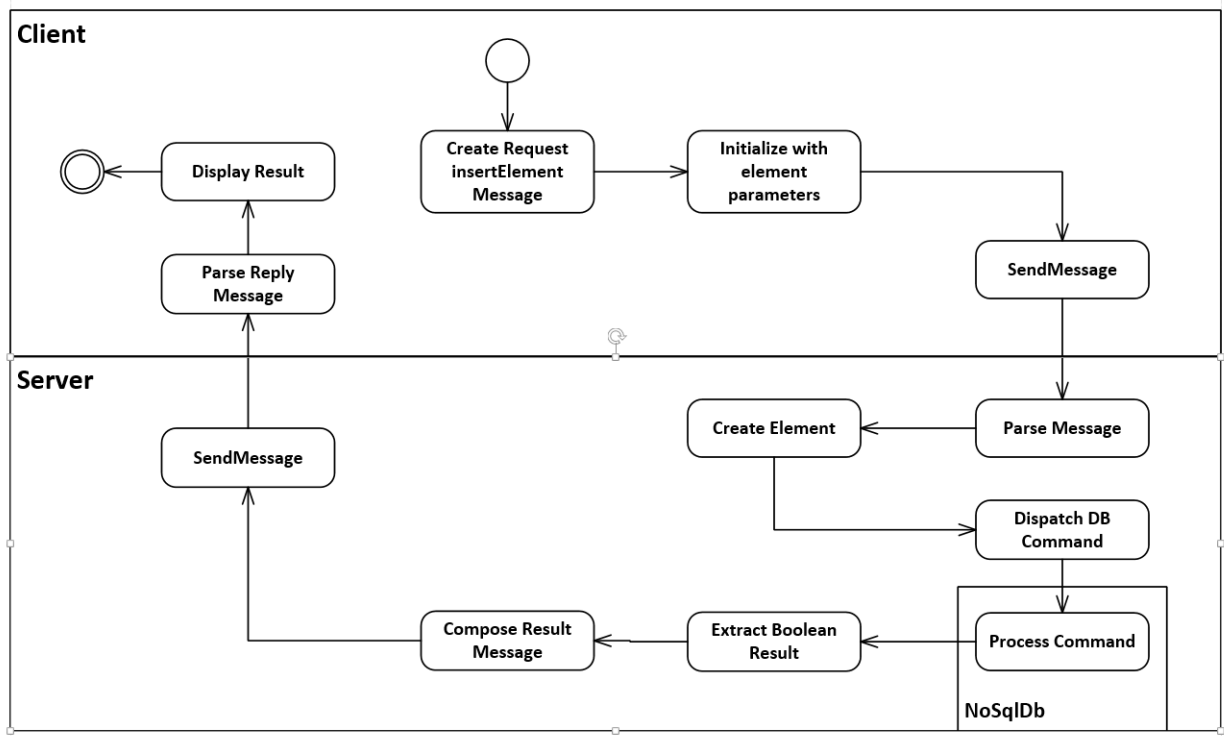
This is a closed book examination. Please place all your books on the floor beside you. You may keep one page of notes on your desktop in addition to this exam package. All examinations will be collected promptly at the end of the class period. Please be prepared to quickly hand in your examination at that time.

If you have any questions, please do not leave your seat. Raise your hand and I will come to your desk to discuss your question. I will answer all questions about the meaning of the wording of any question. I may choose not to answer other questions.

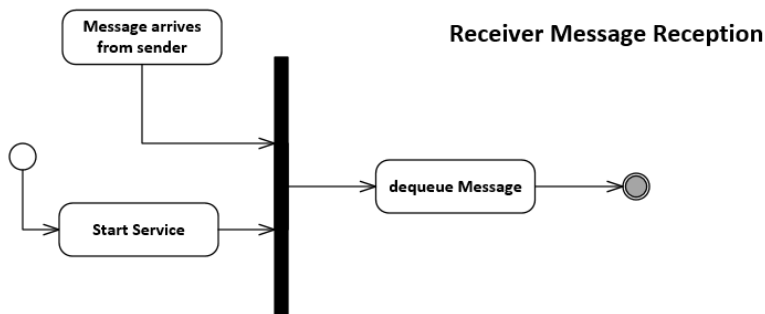
You will find it helpful to review all questions before beginning. All questions are given equal weight for grading, but not all questions have the same difficulty. Therefore, it is very much to your advantage to answer first those questions you believe to be easiest.

- For Project #4 you are required to test a NoSqlDb database using a message-passing communication system. Draw an activity diagram that describes how a client requests the NoSqlDb to add a new item and the resulting server processing. You need to show all the activities involved in that operation. You are not expected to show the communication system activities required to transmit the message other than a single activity: send message.

Answer:



The cmd shown above is an insertElement request, and the parameters are all of the element's data persisted to an XML string¹, using the same techniques used by the NoSqlDb. The result Query, for an insert command will just indicate whether the insertion succeeded or not (it will fail if the supplied key is already used in the server's data store).



¹ See answer to MT1Q4.

2. Write all the code to define and use a C# thread that reads a queue to get jobs to execute. By job we mean processing that carries out some specified activity. You can make up some simple processing as part of this answer.

Answer:

```
// The idea:
// - To pass processing from parent to child thread we need a delegate
//   bound to a lambda that defines the processing job.
// - We need a blocking queue parameterized on the delegate
// - The thread extracts jobs from the queue in a loop
//   and invokes each one.
//
class MT1Q2
{
    public SWTools.BlockingQueue<Action> bQ { get; set; }
        = new SWTools.BlockingQueue<Action>();

    void ThreadProc()
    {
        while(true)
        {
            Action act = bQ.deQ();
            if (act == null)
                break;
            act.Invoke();
        }
    }
    static void Main(string[] args)
    {
        "MT1Q2 - thread that runs queued work items".title('=');
        MT1Q2 mt1q2 = new MT1Q2();
        ThreadStart ts = mt1q2.ThreadProc;
        Thread t = new Thread(ts);
        t.Start();

        Action act1 = () => {
            for(int i=0; i<3; ++i)
            {
                Console.WriteLine("\n doing first task");
            }
            Console.WriteLine("\n finished first task");
        };
        mt1q2.bQ.enQ(act1);
        Action act2 = () => { Console.WriteLine("\n finished second task"); };
        act2 += () => { Console.WriteLine("\n now I'm done"); };
        mt1q2.bQ.enQ(act2);
        mt1q2.bQ.enQ(null); // no more tasks
        t.Join();
        Console.WriteLine("\n\n");
    }
}
```

3. What is System.Type and how is it used. Write a fragment of code to illustrate your answer.

Answer:

System.Type is a container for reflection results. You can use myObject.GetType() to fill it with reflection information from myObject. Here's an example:

```
class MT1Q3
{
    static void Main(string[] args)
    {
        MT1Q3 mt1q3 = new MT1Q3();
        System.Type t = mt1q3.GetType();
        Console.WriteLine("\n t.Name: {0}", t.Name);
        Console.WriteLine("\n t.Namespace: {0}", t.Namespace);
        System.Reflection.MethodInfo[] mi = t.GetMethods(
            BindingFlags.NonPublic | BindingFlags.Public |
            BindingFlags.Static | BindingFlags.Instance
        );
        foreach (var minfo in mi)
            Console.WriteLine("\n {0}", minfo.Name);
        Console.WriteLine("\n\n");
    }
}
```

4. Assuming you use the Data Contract defined in the Instructor's CommPrototype, write XML markup for a message body that provides all the information needed for the remote NoSqlDb database to insert an element. Show how to use System.Xml.Linq.XDocument to construct the body. Please use a key of type int and a payload of type PL_String for the Element to be added.

Answer:

This answer is for the slightly more complicated case of DBElement<string, PL_ListOfStrings>. The only differences between this and a complete answer to the question asked are:

- <key>key1</key> → <key>3</key>
- child string keys → child int keys
- <payload> ... </payload> as below → <payload>a string value</payload>

```

Insert element message body
-----
<insertElement>
  <description>insert element into database</description>
  <key>key1</key>
  <name>testElement</name>
  <descr>test element</descr>
  <timestamp>11/7/2015 7:35:39 PM</timestamp>
  <children>
    <key>key2</key>
    <key>key3</key>
  </children>
  <payload>
    <item>item1</item>
    <item>item2</item>
  </payload>
</insertElement>

public class DBElementProxy // could have used a DBElement instead of proxy
{
  public string name { get; set; }
  public string descr { get; set; }
  public string timeStamp { get; set; }
  public List<string> children { get; set; } = new List<string>();
  public List<string> payload { get; set; } = new List<string>();
}

public static string buildInsertMessageBody(XDocument doc, string key, DBElementProxy proxy)
{
  // doc contains templates for messages - we're using the insertElement template

  XElement insert = doc.Element("messages").Element("insertElement"); // get template
  insert.Element("key").Value = key;
  insert.Element("name").Value = proxy.name;
  insert.Element("descr").Value = proxy.descr;
  insert.Element("timeStamp").Value = proxy.timeStamp;

  foreach(string k in proxy.children)
  {
    XElement item = new XElement("key");
    item.Value = k;
    insert.Element("children").Add(item);
  }

  foreach(string item in proxy.payload)
  {
    XElement itm = new XElement("item");
    itm.Value = item;
    insert.Element("payload").Add(itm);
  }
  return insert.ToString();
}

```

5. What is a generic type? What was the purpose of defining generic types in Project #2? What code do you have to provide to enable the compiler to accept a call to a specified method that you expect the generic type to support?

Answer:

A generic type is a class with an unspecified or partially specified type parameter. The parameter may be used to declare fields, properties, or method signatures.

Generic types allow us to write code once, that will be instantiated by application code with two or more types, instead of writing one non-generic class for each application type. They also help us build flexible implementations that may not need to be changed to accommodate new types.

The compiler will not accept calls on instances of the generic parameter unless it knows that the generic type provides that method. The only direct way to do that is to have the generic parameter implement an interface, `ImyInterface`, with a declaration for the method you want to call, and have the generic class use a constraint of the form: where `T : ImyInterface`. This partial specification promises the compiler that `T` has the called method.

6. Write all the code for a C# function that returns nothing and takes a single argument and modifies its state or value in the caller's scope. You may use specific types and invent any simple modifications you like. Consider all cases.

Answer:

case 1 – reference type, case 2 – value type

```
class MT1Q6
{
    //---- answer starts here ----

    static void changeStateOfArgumentInCallersScope(StringBuilder arg)
    {
        arg.Append(" modified");
    }
    static void changeStateOfArgumentInCallersScope(ref double arg)
    {
        arg += 2.5;
    }
    //---- answer ends here ----

    static void Main(string[] args)
    {
        "MT1Q6 - Function with side-effects".title('=');

        // if argument is reference type, just pass it by value
        StringBuilder sb = new StringBuilder("original state");
        Console.Write("\n Original state of StringBuilder is \"{0}\"", sb.ToString());

        MT1Q6.changeStateOfArgumentInCallersScope(sb);
        Console.Write("\n State of StringBuilder after call is \"{0}\"", sb.ToString());
        Console.WriteLine();

        // if argument is value type you must pass it by reference

        double d = 3.1415927;
        Console.Write("\n Original state of double is \"{0}\"", d);

        MT1Q6.changeStateOfArgumentInCallersScope(ref d);
        Console.Write("\n State of double after call is \"{0}\"", d);

        Console.Write("\n\n");
    }
}
```

7. What does the following code do:

```
Dictionary<int, string> dict = new Dictionary<int, string>().  
  
// some code here that may have added key/value pairs to dict.  
  
dict[3] = "three is enough";
```

Answer:

If the dictionary does not contain the key 3, then the pair { 3, "three is enough"} is inserted into the dictionary dict using a hash function to compute a table address.

If the dictionary does contain the key 3, then it's previous string value is replaced by the new "three is enough" value.