

## ***Examination #3***

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.

1. A matrix is a two dimensional structure containing values of a specified type in rows and columns. Each of the values is accessed by two indices, say i and j. Write all the C# code for a Matrix data structure.

Answer:

```
namespace MT3Q1
{
    public class Matrix<T>
    {
        T[,] matRep;
        int rows_;
        int cols_;

        public Matrix(int i, int j)
        {
            rows_ = i; cols_ = j;
            matRep = new T[rows_,cols_];
        }
        public T this[int i, int j]
        {
            get { return matRep[i,j]; }
            set { matRep[i,j] = value; }
        }
        public void show()
        {
            for (int i = 0; i < rows_; ++i)
            {
                Console.WriteLine("\n ");
                for (int j = 0; j < cols_; ++j)
                    Console.WriteLine(" {0}", this[i, j]);
            }
            Console.WriteLine("\n");
        }
    }
}
class Program
{
    static void Main(string[] args)
    {
        Matrix<double> mat = new Matrix<double>(3,2);
        for (int i = 0; i < 3; ++i)
            for (int j = 0; j < 2; ++j)
            {
                mat[i,j] = i*j - 3;
            }
        mat.show();
    }
}
```

2. Describe the syntax and semantics of passing a C# reference type by reference.

Answer:

To pass a reference type R by reference to a function fun you declare the function like this:

```
returnType fun(ref R) { ... }
```

and you invoke it like this:

```
R myR = new R();  
returnType ret = fun(ref myR);
```

When fun is invoked a new reference is created on the stackframe of fun that refers to the caller's reference. So the function accesses the object on the managed heap through the caller's reference. The result is that any change of object state made in the function is seen by the caller. Also, if the function allocates a new object bound to its reference, the caller also sees the new object. Note that this is not the case for passing a reference type by value.

3. Describe the structure of a Windows Communication Foundation channel. You might consider doing this by describing each of its classes visible to the application.

Answer:

A WCF Service channel requires a service on one end and client on the other. The visible parts of the service are:

**Service contract:** an interface that describes the ServiceOperations, which may depend on a DataContract and/or MessageContract.

**Service implementation** implements the service contract with ServiceBehaviors.

**ServiceHost** is hosted by a windows program, windows service, or may be hosted in IIS. It defines the protocol for communication and endpoint specifics, e.g., address, binding, and contract.

**Service client** uses ChannelFactory or ClientBase to create a proxy which exposes, in the client's address space, the ServiceOperations.

4. Write an XML representation for data input to the distance metric computation for your Project #2. Write LINQ queries to access that information from an XML file.

Answer:

```
<?xml version="1.0" encoding="utf-8" ?>
<DistInput>
  <file>
    <filename>file1</filename>
    <function>
      <name>fun1</name>
      <cc>3</cc>
    </function>
    <function>
      <name>fun2</name>
      <cc>8</cc>
    </function>
  </file>
  <file>
    <filename>file2</filename>
    <function>
      <name>fun1</name>
      <cc>4</cc>
    </function>
    <function>
      <name>fun2</name>
      <cc>8</cc>
    </function>
  </file>
</DistInput>

class Program
{
    static void Main(string[] args)
    {
        XDocument doc = XDocument.Load(@"..\..\Dist.xml");
        Console.Write(doc.ToString());
        Console.WriteLine();
        var files = from e in
                      doc.Elements("DistInput")
                      .Elements("file")
                      select e;

        foreach (var file in files)
        {
            var q2 = from e in file.Elements("filename") select e;
            Console.WriteLine("\n {0}", q2.ElementAt(0).Value);
            var q3 = from e in file.Elements("function").Elements() select e;
            int numFuncs = q3.Count() / 2;
            for (int i = 0; i < numFuncs; ++i)
            {
                Console.WriteLine("\n {0}", q3.ElementAt(2 * i).Value);
                Console.WriteLine(" {0}", q3.ElementAt(2 * i + 1).Value);
            }
        }
        Console.WriteLine();
    }
}
```

5. You've been asked to describe the software architecture of your Project #2. What will you discuss?

Answer:

You need to discuss all the expected parts of an OCD for Project #2, e.g., uses and user groups, project package structure, responsibilities and interactions of each of the packages, and any critical issues faced by the development team.

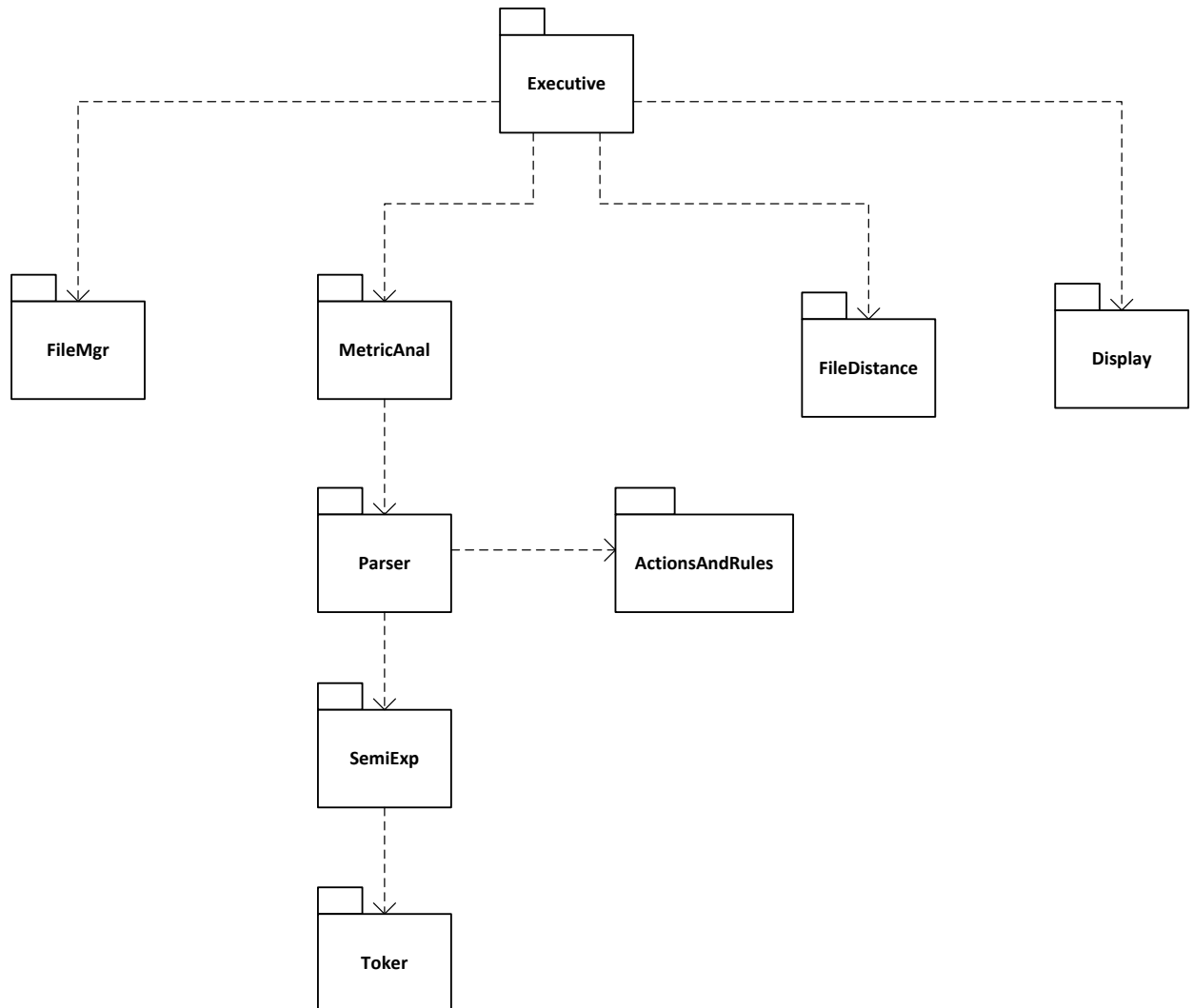
Often you will sketch user interfaces, discuss any interesting data structures and the sharing of data across the application, and document any algorithms or unique design ideas that will make the developed code flexible and robust. The Parser is a good example.

Specific to project #2 you should discuss the adequacy of the distance metric (see my solution for MT1Q7), and the need to gracefully handle small projects for individual developers and very large sets of files that make up a project baseline for Quality Assurance analysis.

6. Draw a package diagram representing a design you may use for your Project #2 implementation.

Answer:

The executive package coordinates the flow of program activities, based on user inputs. FileMgr collects file references for subsequent analysis, Parser and SemiExp and Toker and ActionsAndRules support syntactic analysis of files to evaluate function sizes and cyclomatic complexities. Finally, the Display package presents the analysis results to the user.



7. Why is Windows Presentation Foundation (WPF) a good choice for implementing the user interface for Project #4?

Answer:

WPF provides a very effective drawing library, much easier to use than previous frameworks like WinForms and Microsoft Foundation Classes.

It supports a pervasive notification model. So when a graphical shape is dragged to a new location it is easy to set up notifications to make any connector ends move to maintain connectivity.

WPF provides several panel types with different behaviors. So, for example, the Canvas panel supports absolute positioning in both the coordinates of the window, but also along the normal to the window. So it is easy to implement pop-up and sliding child elements for tool box and property selection.