**实验7 查找表结构和查找算法的设计、实现与应用**

实验目的：

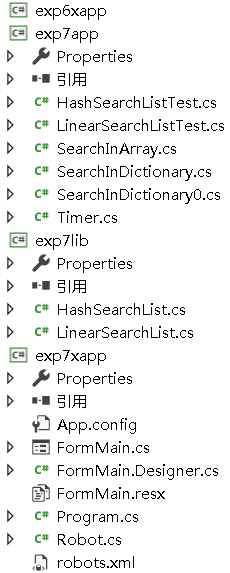
熟练掌握顺序表和有序表的数据结构设计和查找算法实现及其平均查找长度的计算方法；深刻理解哈希表的构造方法及其与其它查找表结构的实质性差别，熟练运用于编程任务中。熟练掌握Visual Studio进行调试和多项目管理。

任务要求：

1. 熟练掌握Visual Studio进行调试与多项目和类的创建和管理。完成本次全部实验，需在实验的总解决方案中新建三个项目：1）一个“类库（.Net Framework）”型项目（如称作exp7lib），用于实现自定义的查找表结构和查找算法，这些类都声明在命名空间DSA中。2）一个“控制台应用（.Net Framework）”类型的项目（如称作exp7app）, 用于查找表实例的创建和查找算法的测试、演示和应用。项目exp7app需要引用项目exp7lib中的类库模块，并在测试和应用代码中加入“using DSA”指令，以方便源代码的编辑。3）一个“Windows窗体应用”类型的项目（如称作exp7xapp）, 用于最后一个实验的设计（选做）。

部分复杂的算法，可以借助Visual Studio的调试功能，如单步调试、设置断点等，发现逻辑错误，或更好地理解算法的动态特性。

本次实验完成后，项目及类源代码文件结构如下所示：



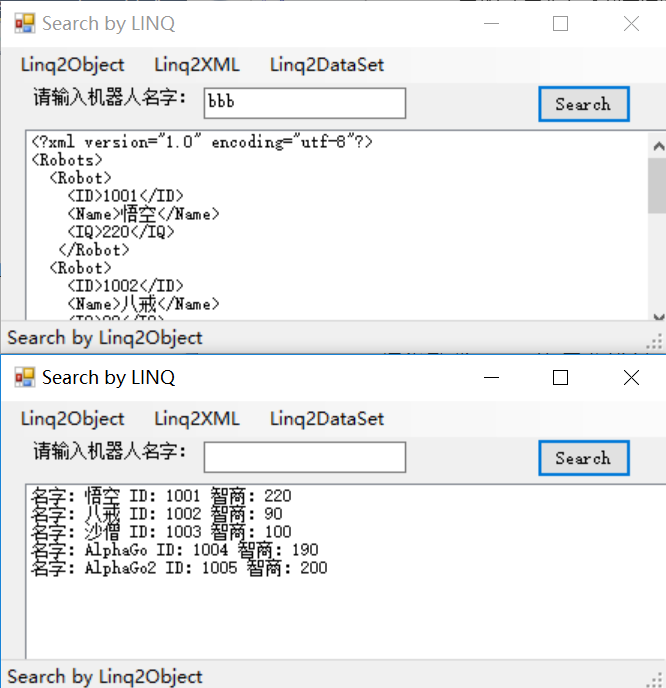
2. 在exp7app项目中定义一个含Main方法的类（SearchInArray.cs），在其中定义和随机初始化一个（具有10000个元素/值域在0到9999的）整数数组，在数组中顺序查找值为5, 15, 25, 35, 45, 55, 65, 75, 85, 95, 105, 115, 125, 135, 145, 155, 165, 175, 185, 195的20个数据；对数组中的数据进行排序，然后在数组中二分查找这20个数据。试着比较两种查找算法所花费的不同时间。熟练掌握顺序查找算法、二分查找算法、类库中Array类型。

3. 在exp7app项目中定义一个含Main方法的类（SearchInDictionary.cs），在其中定义和初始化一个有序string数组（具有20000个元素，元素值分别为“1”到“20000”）和Dictionary<string,int>实例（具有20000个形如(“1”,1)和(“2”,2)的“键-值对”，键分别为“1”到“20000”，分别在数组和Dictionary实例中查找值为“ 500”, “ 1500”, “ 2500”,…, “10500”, “11500”,…, “19500”等20个数据。试着比较在不同的数据结构和算法（有序数组可以用顺序查找也可用二分查找）所花费的不同时间。熟悉类库中Dictionary类型。

4. 在exp7lib项目中编程实现自定义的顺序查找表类LinearSearchList，实现其中的造表（Add，Insert）、顺序查找(IndexOf)、二分查找 (BinarySearch)等算法。在exp7app项目中定义一个含Main方法的测试类（LinearSearchListTest.cs），在其中定义和随机初始化一个查找表实例，测试顺序查找和二分查找算法。

5. 在exp7lib项目中设计并编程实现自定义的哈希链表类HashSearchList，实现其中的造表与查找（Hash，Add, Search）等算法。在exp7app项目中定义一个含Main方法的测试类（HashSearchListTest.cs），在其中定义和随机初始化一个Hash查找表实例，测试Hash查找算法。

6. 选做实验：在exp7xapp项目中编写一份XML文件robots.xml，包含一组数据，具有ID、Name、IQ等字段。编写一个Windows窗体应用程序，具有如下图所示的界面，可以让用户通过姓名查找相应的数据。初步熟悉LINQ语句并通过LINQ在数据库中查询（Linq to Object，Linq to XML, Linq to DataSet）。



XML文件robots.xml的内容如下：

<?xml version="1.0" encoding="utf-8"?>

<Robots>

<Robot>

<ID>1001</ID>

<Name>悟空</Name>

<IQ>220</IQ>

</Robot>

<Robot> <ID>1002</ID> <Name>八戒</Name> <IQ>90</IQ> </Robot>

<Robot> <ID>1003</ID> <Name>沙僧</Name> <IQ>100</IQ> </Robot>

<Robot> <ID>1004</ID> <Name>AlphaGo</Name> <IQ>190</IQ> </Robot>

<Robot>

<ID>1005</ID> <Name>AlphaGo2</Name> <IQ>200</IQ>

</Robot>

</Robots>