

实验 3 栈与队列的设计与实现

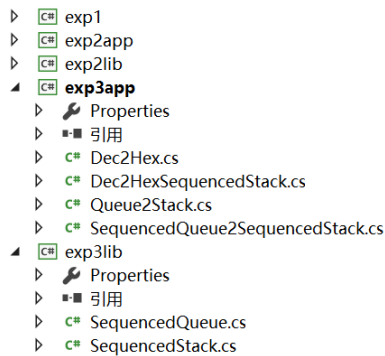
实验目的：

理解栈与队列的基本概念及其基本操作，熟练运用于编程任务中；熟练掌握 Visual Studio 进行多项目管理。

题意要求：

1. 熟练掌握 Visual Studio 进行项目和类的建立和管理。完成本次全部实验，需在实验的总解决方案中新建两个项目：1) 一个控制台应用程序类型的项目（如称作 exp3app），用于实现栈与队列数据结构的测试、演示和应用。2) 一个“类库”型项目（如称作 exp3lib），用于实现自定义的栈与队列数据结构，这些类都声明在命名空间 DSAGL 中。项目 exp3app 需要引用 exp3lib 类库模块，并在测试和应用代码中加入“using DSAGL”指令，以方便源代码的编辑。exp3app 项目和 exp3lib 类库项目可能都需要引用 exp2lib 类库模块（前一章实验的成果）。

本次实验完成后，项目及类源代码文件结构如下所示：



```
exp1
exp2app
exp2lib
exp3app
├── Properties
├── 引用
├── Dec2Hex.cs
├── Dec2HexSequencedStack.cs
├── Queue2Stack.cs
├── SequencedQueue2SequencedStack.cs
exp3lib
├── Properties
├── 引用
├── SequencedQueue.cs
├── SequencedStack.cs
```

2. 利用栈进行数制转换。在 exp3app 项目中定义一个含 Main 方法的类（Dec2Hex.cs），在其中利用 Stack<T>类定义和初始化一个 int 类型的栈对象，将 10 进制整数转换为 16 进制整数，正确输出结果。说明栈在该过程所起的作用。认识和使用 Stack<T>泛型类。

3. 利用队列和栈将一个序列反序。在 exp3app 项目中定义一个含 Main 方法的类（Queue2Stack.cs），在其中利用 Queue<T>类定义和初始化一个 int 类型的队列对象，借用一个栈对象将队列中的原序列反序，输出反序后的结果。认识和使用 Queue<T>泛型类。

4. 在 exp3lib 项目中编程实现自定义栈类与队列类。可以只考虑顺序存储的实现方案，设计为泛型类（SequencedStack<T>和 SequencedQueue<T>）。

5. 在 exp3app 项目中定义含 Main 方法的类（Dec2HexSequencedStack.cs, SequencedQueue2SequencedStack.cs），将任务 2 和 3 中所有的 Stack 和 Queue 类替换为自己设计的栈类 SequencedStack 与队列类 SequencedQueue，检查自己的设计与 Stack 和 Queue 类在基本性质上是否相同。

6. 选做实验：表达式求值，将中缀表达式转换为后缀表达式，再求后缀表达式的值。