

实验 2 线性表的设计与实现

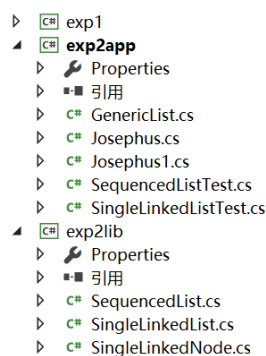
实验目的：

理解线性表的基本概念及其基本操作，熟练运用于编程任务中；熟练运用 C# 自引用类的方式实现线性链表的基本操作；熟练掌握 Visual Studio 进行项目管理。

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 exp2app），用于实现线性表数据结构的测试、演示和应用。2) 一个“类库”型项目（如称作 exp2lib），用于实现自定义的线性表数据结构，这些类都声明在命名空间 DSAGL 中。项目 exp2app 需要引用 exp2lib 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。

本次实验完成后，项目及类源代码文件结构如下所示：



```
exp1
├── exp2app
│   ├── Properties
│   ├── 引用
│   ├── GenericList.cs
│   ├── Josephus.cs
│   ├── Josephus1.cs
│   ├── SequencedListTest.cs
│   └── SingleLinkedListTest.cs
└── exp2lib
    ├── Properties
    ├── 引用
    ├── SequencedList.cs
    ├── SingleLinkedList.cs
    └── SingleLinkedListNode.cs
```

2. 在 exp2app 项目中定义一个含 Main 方法的类 (GenericList.cs)，在其中利用 List<T> 类定义和初始化一个 int 类型的线性表，在表中添加 (Add) 和插入 (Insert) 新的元素；定义一个自定义 Student 类，定义和初始化一个 Student 类型的线性表，在表中添加 (Add) 和插入 (Insert) 新的元素。说明线性表与数组主要特性上的异同。认识和使用 List<T>、Array 等类型，认识和使用泛型类。

3. 在 exp2lib 项目中编程实现一个包含起标志作用的头结点的单向链表类。它的头结点不包含数据元素，仅起标志作用，头结点的链指向链表的第一个数据结点（这即是讲义上的实现方案）。设计泛型链表类和结点类 (SingleLinkedList 和 SingleLinkedListNode)。如果由于教学进度安排原因，该项实验任务可替换为编程实现顺序表结构 (SequencedList)。

4. 将任务 2 中所有使用 List<T> 类的地方替换为自己设计的单向链表类 SingleLinkedList (或顺序表 SequencedList)，检查自己的设计与 List<T> 类在基本性质上是否相同。

5. 选做实验：设计和实现一种不含头结点的单向链表类。与不含头结点的设计相比较，具有头结点的单向链表类的设计，在一些操作的实现上带来了方便，例如无需判断链表是否是单数据结点的情况。

6. 选做实验：选择一种合适的线性表类求解约瑟夫 (Josephus) 环问题。