

**Lab report**

|  |  |
| --- | --- |
| **Course**: | Class Libraries and Data Structures |
| **Semester**: | 1st semester of the academic year **2019-2020** |
| **Major**: | Software Engineering |
| **Class**: | 2018 |
| **Student Name**: | SONG,Xingjian（宋行健） |
| **Student ID:** | 222019321062006 |
| **Teacher:** | ZHAO, Hengjun (赵恒军) |

**School of Computer and Information Science**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | | Iterator | | | |
| Date | | Oct 31，2019 | Type | | √ Confirmatory  √ Design  □Comprehensive |
| 1. **Objective & Requirements**    1. Know the use of iterator and understand its implementation details    2. Can use iterator to traverse a list to finish a certain task | | | | | |
| 1. **Experimental environment (**platform and software**)**   Windows 7 (or higher versions) + Visual Studio 2010 (or higher versions) | | | | | |
| 1. **Experimental content and design** (Main Content, Procedure, Codes and Results)   Task 1  Using the source code sent to you about the linked list template with iterator, implement a the method in the company class   * + - findBestPaid()   and test your implementation in the main function.  Task 2   * + - 1. Based on the source code of the container template using array storage, implement an iterator inner class for it       2. Based on the implemented iterator, implement the findBestPaid() method for the Company class, just as what you do in Task 1   Task1:  首先创建一个名为curr的临时变量，存放当前迭代器所遍历的节点，并初始化到链表头位置。在循环遍历的时候，计数变量自增的同时，curr同时自增，这主要是运算符++重载的功劳。在遍历的同时进行判断，如果当前节点的工资更大，则通过\*将此节点所储存的变量赋值给bestPaid。    Task2:  无参的构造函数，直接将构造器指针curr\_ptr指向null，至于有参数的构造函数则将指针curr\_ptr指向参数指针。重载自增函数++直接对指针自增即可，析取\*和比较==同理。    Begin函数以当前数组的下标为0的位置创建一个迭代器并返回，在这里elemArray是数组起始位置的指针，End函数以空值创建一个迭代器并返回，null即为数组最后的末尾位置的下一位。    findBestPaid函数与实验一类似。 | | | | | |
| 1. **Result analysis and discussion**（Analysis of experimental results and summing up the harvest and the existing problems）   Task1:  在创建一个名为curr的临时变量时，因为迭代器Iterator为ListTemp的内嵌类，所以在调用时要注意格式ListTemp<Employee>::Iterator curr = container.Begin()；需要先写外层类域。  另一个需要注意的地方是当前节点数据与bestPaid比较时要进行析取操作，因为它是一个指针。  运行结果如下：    Task2:  数组迭代器和链表迭代器大体上类似，不同之处在于链表的访问需要进行next指针的传递，而数组可以指直接进行自增、自减的运算。  另外，在写begin函数的时候，我对如何表示数组的起始位置产生了疑惑。在请教了老师后，注意到elemArray这个指针在数组定义之初即指向了数组的起始位置。  运行结果如下： | | | | | |
| Comments & Evaluation | Content & Design (A-E) | | |  | |
| Procedure & Codes (A-E) | | |  | |
| Results (A-E) | | |  | |
| Analysis & Discussion (A-E) | | |  | |
| Score (A-E):  Feedback comments: | | | | |