CSE 2020 Lab 07 Queues

Lab Exercise

Write C++ programs to implement Queue ADT using doubly linked structure, and then test it.

- Define the `Queue` class template in the file `DoublyLinkedQueue.cpp`.

```
// DoublyLinkedQueue.cpp
#ifndef QUEUE_H
#define QUEUE_H
#include <iostream>
using namespace std;

template <typename T>
class Queue
{
public:
  // Default constructor
  Queue()
  {
          front = new NodeType;
          back = new NodeType;
          front->next = back;
          back->prev = front;
  }

  //Desctructor
  ~Queue()
  {
          clear();
          delete front;
          delete back;
  }

  // Makes the queue to the empty state.
  void clear()
  {

  }

  // Checks if the queue is empty.
  bool empty() const
  {

  }

  // Inserts item at the end of the queue.
  void enqueue(const T& item)
  {

  }

  // Removes the element at the start of the queue.
  void dequeue()
  {

  }

  // returns the front element
  const T& front_element() const
  {

  }

  // Prints the elements of the queue.
  void print() const
  {
          NodeType* ptr = front->next;
          while (ptr != back)
          {
```

```
                cout << ptr->data << ", ";
                ptr = ptr->next;
        }
        cout << endl;
  }

private:
  struct NodeType
  {
        T data;
        NodeType* next;
        NodeType* prev;

        NodeType(const T & d = T()): data(d), prev(nullptr), next(nullptr)
        {}
  };

  NodeType* front; // points to the header node
  NodeType* back;  // points to the tail node
};

#endif
```

- Please read the comments carefully and implement the `Queue` template class.
- The main function is contained in the file `lab07.cpp`.

```
// lab07.cpp
#include "Employee.cpp"
#include "DoublyLinkedQueue.cpp"

int main()
{
....
}
```

The main function,

1. Declares a queue which stores int values.
2. Prompts users to enter integers, enqueues the entered values, stops entering the integers when users enter 0.
3. Prints the elements of queue.
4. Prompts the user to enter a number `k`, and dequeues `k` values from the front of the queue.
5. Prints the elements of queue.

6. Declares a queue which stores strings.
7. Prompts users to enter strings, enqueue the entered strings, stops entering the strings when the user enter "exit".
8. Prints the elements of queue.
9. Prompts users to enter a number `k`, and dequeues `k` strings from the front of the queue.
10. Prints the elements of queue.

11. Declares a queue which stores Employee objects.
12. Prompts users to enter employee's id, name, and department name, create an employee object, enqueue the created employee boject, stops entering the employees when the user enter id as 0.
13. Prints the elements of queue.
14. Prompts users to enter a number `k`, and dequeues `k` employees from the front of the queue.
15. Prints the elements of queue.

The expected result:

```
Enqueue positive numbers (enter -1 to stop): 1 2 3 4 5 6 7 8 9 0
print queue: 1, 2, 3, 4, 5, 6, 7, 8, 9,

How many numbers to be removed from queue: 5
```

```
print queue: 6, 7, 8, 9,

Enqueue string (enter exit to stop): aaa bbb ccc ddd eee fff exit

print queue: aaa, bbb, ccc, ddd, eee, fff,

How many strings to be removed from queue: 3

print queue: ddd, eee, fff,


Enqueue employee's id, name, dept (enter id 0 to stop): 1 bob CSE
Enqueue employee's id, name, dept: 2 mary Math
Enqueue employee's id, name, dept: 3 joe Art
Enqueue employee's id, name, dept: 4 Ellen CSE
Enqueue employee's id, name, dept: 0 no no

print queue: 1 bob CSE, 2 mary Math, 3 joe Art, 4 Ellen CSE,

How many strings to be removed from queue: 2

print queue: 3 joe Art, 4 Ellen CSE,
```

Compilation

This lab exercise should be put under `cse2020/lab07` subdirectory.

```
$g++ -c DoublyLinkedQueue.cpp
$g++ lab07.cpp -o lab07
$./lab07
...
$script lab07log.txt
...
```

Hand In

- DoublyLinkedQueue.cpp: the class file.
- lab07.cpp: the test file containing `main()` funcion.
- lab07log.txt: the script file which captures the result.

Yan Zhang
