

Thực hành ngăn xếp và hàng đợi

Nguyễn Mạnh Hiền

hiennm@tlu.edu.vn

Bài thực hành 1

- (a) Cài đặt cấu trúc dữ liệu ngăn xếp (dùng mảng) theo khung chương trình cho bên dưới.
- (b) Bổ sung hàm in tất cả các phần tử trong ngăn xếp mà không phải xóa các phần tử.
- (c) Bổ sung hàm kiểm tra xem một giá trị x có ở trong ngăn xếp hay không.

```
#include <iostream>
using namespace std;

template <typename T>
class Stack
{
public:
    // Dung lượng ngăn xếp (capacity) có giá trị ngầm định là 100.
    Stack(int capacity = 100) { ... }

    ~Stack() { ... }

    // Kiểm tra ngăn xếp có rỗng hay không.
    bool empty() { ... }

    // Lấy kích thước (số phần tử hiện có) của ngăn xếp.
    int getSize() { ... }

    // Thêm phần tử e vào ngăn xếp.
    void push(T e) { ... }

    // Xóa phần tử khỏi ngăn xếp.
    void pop() { ... }

    // Lấy phần tử nằm ở đỉnh ngăn xếp (nhưng không xóa).
    T top() { ... }

private:
    T * theArray; // Con trỏ tới mảng chứa các phần tử.
    int topOfStack; // Chưa vị trí của phần tử nằm ở đỉnh ngăn xếp.
};

int main()
{
    Stack<char> s;

    if (s.empty())
        cout << "Ngăn xếp đang rỗng" << endl;

    s.push('B');
    s.push('E');
    s.push('K');
```

```

s.push('A');
s.push('C');

cout << "Kich thuoc ngan xep sau khi chen: " << s.getSize()
<< endl; // Se in ra 5

cout << "Cac phan tu trong ngan xep: ";
while (!s.empty()) // Se in ra: C A K E B
{
    cout << s.top() << " ";
    s.pop();
}
cout << endl;

cout << "Kich thuoc ngan xep sau khi xoa rong: " << s.getSize()
<< endl; // Se in ra 0

return 0;
}

```

Bài thực hành 2

- (a) Cài đặt cấu trúc dữ liệu hàng đợi (dùng danh sách liên kết đơn) theo khung chương trình cho bên dưới.
- (b) Bổ sung hàm in tất cả các phần tử trong hàng đợi mà không phải xóa các phần tử.
- (c) Bổ sung hàm kiểm tra xem một giá trị x có ở trong hàng đợi hay không.

```

#include <iostream>
using namespace std;

template <typename T>
class Queue
{
public:
    Queue() { ... }

    ~Queue() { ... }

    // Kiểm tra hàng đợi có đang rỗng hay không.
    bool empty() { ... }

    // Lấy kích thước (số phần tử hiện có) của hàng đợi.
    int getSize() { ... }

    // Thêm phần tử e vào cuối hàng đợi.
    void enqueue(T e) { ... }

    // Xóa phần tử ở đầu hàng đợi.
    T dequeue() { ... }

private:
    struct Node
    {

```

```

        T elem;
        Node * next;
    };

    Node * front; // Con tro toi nut dau danh sach
    Node * back;  // Con tro toi nut cuoi danh sach
    int size;     // Kich thuoc cua hang doi
};

int main()
{
    Queue<int> q;
    q.enqueue(3);
    q.enqueue(6);
    q.enqueue(1);
    q.enqueue(2);

    cout << "Kich thuoc hang doi sau khi chen: " << q.getSize()
    << endl; // Se in ra 4

    cout << "Cac phan tu: ";
    while (!q.empty()) // Se in ra 3 6 1 2
        cout << q.dequeue() << " ";
    cout << endl;

    cout << "Kich thuoc hang doi sau khi xoa: " << q.getSize()
    << endl; // Se in ra 0

    return 0;
}

```