编译原理 PA1

20238132022 物联网 2 班朱彦渊

2025 年 9 月 30 日

所选数据结构：队列

所有代码均保存在 queue.cl 中。由于在 linux console 下使用中文输入可能会导致乱码（本来写了，但是写完注释后面居然自动乱了，直接又跑不了了 TAT，后面改了好久才回来），所以不在代码中写注释，对部分代码的解释转到文档中进行。代码分为三个部分，分别是 Node 类 Queue 类和 Main 类。其中 Node 作为 Queue 的节点，有 Data 和 Next 两个变量。Queue 为队列本身，有 head，tail，size 三个变量，以及课件要求的功能函数. cool 语言的类中的变量应该默认为 private 的，所以在定义 Node 类的时候我使用了 setNext(), getData(), getNext() 三个函数进行辅助。由于偷懒，队列的元素默认为 Int 型，如果使用 Object 型的话，应该要在输出的时候分类调用不同的 IO 方法。Main 函数用于测试队列，该程序的运行结果如下图所示，符合课件要求的最小功能集.

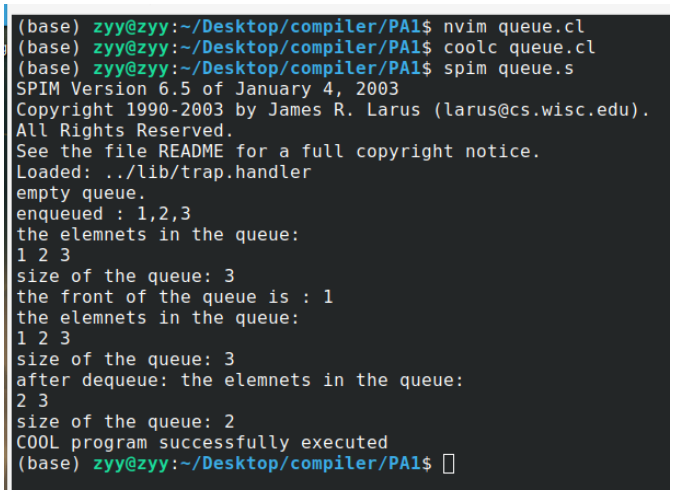

Fig. 1. 程序运行结果图

完整代码如下:

```
1     -- 队 列 节 点
2  class Node inherits Object{
3    data : Int;
4    next : Node;
5
6    init(d : Int, n : Node) : Node {
7      {
8        data <- d;
9        next <- n;
10       self;
11     }
12   };
```

```
13
14    setNext( n : Node) : Object{
15      {
16        next <- n;
17        self;
18      }
19    };
20
21    getData() : Int{
22      data
23    };
24
25    getNext() : Node{
26      next
27    };
28
29  };
30
31  --队列
32  class Queue inherits IO{
33    head : Node;
34    tail : Node;
35    size : Int;
36    init() :  Queue{
37      {
38        size <- 0;
39        self;
40      }
41    };
42
43    isEmpty() : Bool {
44      isvoid head
45    };
46
47    enqueue (data : Int) : Object {
48      {
49      let n : Node in
50      let newNode : Node <- (new Node).init(data, n) in
51      {
52        if isEmpty() then{
53          head <- newNode;
```

```
54          tail <- newNode;
55        }
56        else{
57          tail.setNext(newNode);
58          tail <- newNode;
59        }fi;
60
61        size <- size + 1;
62      };
63      }
64    };
65
66    dequeue () : Int{
67      if isEmpty() then
68        {
69          out_string("Error! Can't dequeue an empty queue!\n");
70          0;
71        }
72      else{
73        let data2rm : Int <- head.getData() in {
74          head <- head.getNext();
75          size <- size - 1;
76
77          data2rm;
78        };
79      }
80      fi
81    };
82
83
84    front () : Int{
85      if isEmpty() then
86        {
87          out_string("Error! Empty queue!\n");
88          0;
89        }
90      else{
91        head.getData();
92      }
93      fi
94    };
```

```
95
96
97    print () : Object {
98      if isEmpty() then{
99        out_string("empty queue!\n");
100     }
101     else{
102       let curr : Node <- head in{
103         out_string("the elemnets in the queue:\n");
104         while not (isvoid curr) loop{
105           out_int(curr.getData());
106           out_string(" ");
107           curr <- curr.getNext();
108         }pool;
109         out_string("\n");
110         out_string("size of the queue: ");
111         out_int(size);
112         out_string("\n");
113       };
114     } fi
115
116   };
117 };
118
119 -- Main函数,用于测试.
120 class Main inherits IO {
121   main() : Object {
122     let q : Queue <- new Queue in {
123
124       if q.isEmpty() then
125         out_string("empty queue.\n")
126       else
127         out_string("not empty.\n")
128       fi;
129
130       out_string("enqueued : 1,2,3\n");
131       q.enqueue(1);
132       q.enqueue(2);
133       q.enqueue(3);
134       q.print();
135
```

```
136         out_string("the front of the queue is : ");
137         out_int(q.front());
138         out_string("\n");
139         q.print();
140 
141         q.dequeue();
142         out_string("after dequeue: ");
143         q.print();
144 
145 
146       }
147     };
148 };
```