第一次平时作业

姚文顶

2025 年 9 月 29 日

目录

1 第一次作业:cool语言写队列

github链接：github.com

对于队列而言，主要就是三件事：入队出队和查队。以及开始和结束：创队和出队

先设置队列元素——即数据本身和下一个结点位置，用于链式存储。

用头尾指针指向第一个和最后一个结点位置即可完成队列（一直用下一结点位置查到头尾指针相遇），入队判断是否是第一，第一头尾指针都要设置位置；否则就设置尾指针位置。

查队，创建一个新指针，在头指针位置一直查到尾即可，也不需要出队操作。然后每次查一个+1即可完成计数操作

出队，就.....头指针指向元素数据转存，然后头指针指向下一个数据。如果是最后一个元素则不动。

（其实可以在入队的时候就计数，头指针指向队首用于计数，然后下一个元素才是队列首元素，然后出队的时候出第二位即可（即front指向元素的next进行修改））

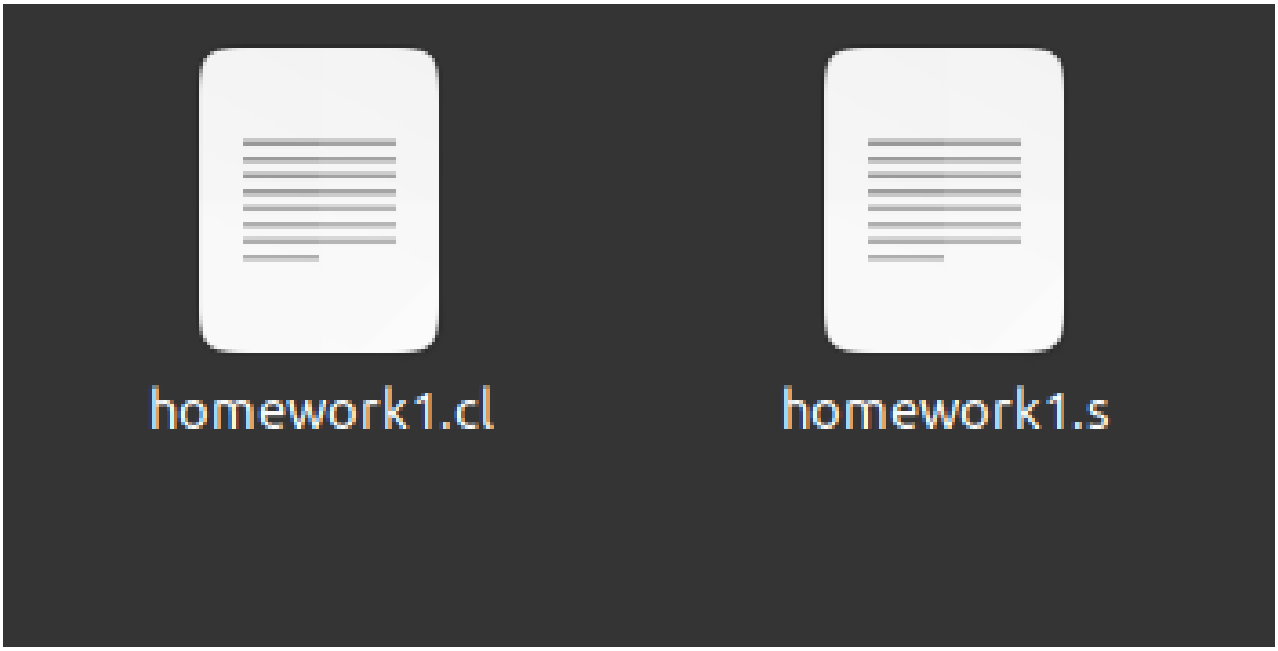
于是便有如下代码：

```
--定义结点类型
class NewNode inherits Object{
    data : Object;
    next : NewNode;

--设置结点初始化
    init( i : Object, n : NewNode) : NewNode {
        {
            data <- i;
            next <- n;
            self;
        }
    };

--便利代码阅读的函数
    gData() : Object {
        data
    };
    gNext() : NewNode{
        next
    };
    setNext(n : NewNode) : SELF_TYPE{
        {
            next <- n;
            self;
        }
    };
};

--设置队列
class Queue inherits Object{
    front : NewNode;
    rear : NewNode;
    --初始化
    initQueue() : SELF_TYPE{
        {
            let v1 : NewNode in front <- v1;
            let v2 : NewNode in rear <- v2;
            self;
        }
    };

    --判断队列是否为空
    isEmpty() : Bool{
        isvoid front
    };

    --enqueue(data)  入队操作
    enqueue(data : Object) : SELF_TYPE{
        {
            let v : NewNode in
            let nNode : NewNode <- (new NewNode).init(data, v) in {
                if isEmpty() then{
                    front <- nNode;
                    rear <- nNode;
                }
                else{
                    rear.setNext(nNode);
                    rear <- nNode;
                } fi;
            };
            self;
        }
    };
```

```
    --dequeue()  出队操作
    dequeue() : Object {
        if   isEmpty() then{
            (new IO).out_string("Nothing in Queue\n");
        }
        else {
            let d : Object <- front.gData() in{
                front <- front.gNext();
                if isEmpty() then{
                    let v : NewNode in rear <- v;
                }else{
                    new Object;
                }fi;
                d;
            };
        }fi
    };

    --frontData()  查队首元素
    frontData() : Object{
        if isEmpty() then{
            new Object;
        }
        else {
            front.gData();
        }fi
    };

    --print()  输出所有元素
    print() : Object{
        if isEmpty() then{
            (new IO).out_string("Nothing in Queue\n");
        }
        else{
            let tPoint : NewNode <- front in
            while not isvoid tPoint loop{
                case tPoint.gData() of      --个同元素个同输出
                    s : String => (new IO).out_string(s);
                    i : Int => (new IO).out_int(i);
                    o : Object => (new IO).out_string(o.type_name());
                esac;
                (new IO).out_string(" ");
                tPoint <- tPoint.gNext();
            }pool;
        }fi
    };

    --获取队长
    size() : Int {
        let count : Int <- 0, tPoint : NewNode<- front in{
        while not isvoid tPoint loop{
            count <- count + 1;
            tPoint <- tPoint.gNext();
        }pool;
        count;
        }
    };
};

class Main inherits IO {
    main() : Object {
        let q : Queue <- (new Queue).initQueue() in {
            q.print();
```

附注：其实队列本身不难。主要还是熟悉cool语言的格式，比如¡-，isvoid,:,let in之类的使用方法。总体而言，感觉还行。

```
                q.enqueue("firstone");--测试字符串和数字的输出
                q.enqueue(2);
                q.enqueue("3rd");
                q.print();


            if q.isEmpty() then {
              out_string("Queue is empty, no front data\n");
            } else {
                out_string("Front: ");
                case q.frontData() of
                    s : String => out_string(s);
                    i : Int    => out_int(i);
                        o : Object => out_string(o.type_name());
                esac;
          out_string("\n");
        } fi;
                q.dequeue();
                q.print();
                --看看出队一个的输出
                q.dequeue();
                q.dequeue();
                q.print();
                --全出队输出
            }
    };
};
```




```
yaokc@ubuntu:~$ spim homework1.s
SPIM Version 6.5 of January 4, 2003
Copyright 1990-2003 by James R. Larus (larus@cs.wisc.edu).
All Rights Reserved.
See the file README for a full copyright notice.
Loaded: ../lib/trap.handler
Nothing in Queue
firstone 2 3rd Front: firstone
2 3rd Nothing in Queue
COOL program successfully executed
yaokc@ubuntu:~$ S
```