package edu.javacourse.queue;

public class QueueTest

{

    public static void main(String[] arg) {

        ObjectQueue<String> queue = new ObjectQueue<>();

        for(int i=0; i<10; i++) {

            queue.push("Строка:" + i\*100);

        }

        for(int i=0; i<queue.size(); i++) {

            String s = queue.get(i);

            System.out.println(s);

        }

        System.out.println("===============");

        while(queue.size() > 0) {

            String s = queue.pull();

            System.out.println(s + " Размер:" + queue.size());

        }

    }

}

package edu.javacourse.queue;

public class ObjectQueue<T>

{

    // Указатель на первый элемент

    private ObjectBox<T> head = null;

    // Указатель на последний элемент

    private ObjectBox<T> tail = null;

    // Поле для хранения размера очереди

    private int size = 0;

    public void push(T obj) {

        // Сразу создаем вспомогательный объект и помещаем новый элемент в него

        ObjectBox<T> ob = new ObjectBox<>();

        ob.setObject(obj);

        // Если очередь пустая - в ней еще нет элементов

        if (head == null) {

            // Теперь наша голова указывает на наш первый элемент

            head = ob;

        } else {

            // Если это не первый элемент, то надо, чтобы последний элемент в очереди

            // указывал на вновь прибывший элемент

            tail.setNext(ob);

        }

        // И в любом случае нам надо наш "хвост" переместить на новый элемент

        // Если это первый элемент, то "голова" и "хвост" будут указывать на один и тот же элемент

        tail = ob;

        // Увеличиваем размер нашей очереди

        size++;

    }

    public T pull() {

        // Если у нас нет элементов, то возвращаем null

        if (size == 0) {

            return null;

        }

        // Получаем наш объект из вспомогательного класса из "головы"

        T obj = head.getObject();

        // Перемещаем "голову" на следующий элемент

        head = head.getNext();

        // Если это был единственный элемент, то head станет равен null

        // и тогда tail (хвост) тоже дожен указать на null.

        if (head == null) {

            tail = null;

        }

        // Уменьшаем размер очереди

        size--;

        // Возвращаем значение

        return obj;

    }

    public T get(int index) {

        // Если нет элементов или индекс больше размера или индекс меньше 0

        if(size == 0 || index >= size || index < 0) {

            return null;

        }

        // Устанавлваем указатель, который будем перемещать на "голову"

        ObjectBox<T> current = head;

        // В этом случае позиция равну 0

        int pos = 0;

        // Пока позиция не достигла нужного индекса

        while(pos < index) {

            // Перемещаемся на следующий элемент

            current = current.getNext();

            // И увеличиваем позицию

            pos++;

        }

        // Мы дошли до нужной позиции и теперь можем вернуть элемент

        T obj = current.getObject();

        return obj;

    }

    public int size() {

        return size;

    }

    // Наш вспомогательный класс будет закрыт от посторонних глаз

    private class ObjectBox<T>

    {

        // Поле для хранения объекта

        private T object;

        // Поле для указания на следующий элемент в цепочке.

        // Если оно равно NULL - значит это последний элемент

        private ObjectBox<T> next;

        public T getObject() {

            return object;

        }

        public void setObject(T object) {

            this.object = object;

        }

        public ObjectBox getNext() {

            return next;

        }

        public void setNext(ObjectBox<T> next) {

            this.next = next;

        }

    }

}

run:

Строка:0

Строка:100

Строка:200

Строка:300

Строка:400

Строка:500

Строка:600

Строка:700

Строка:800

Строка:900

===============

Строка:0 Размер:9

Строка:100 Размер:8

Строка:200 Размер:7

Строка:300 Размер:6

Строка:400 Размер:5

Строка:500 Размер:4

Строка:600 Размер:3

Строка:700 Размер:2

Строка:800 Размер:1

Строка:900 Размер:0

СБОРКА УСПЕШНО ЗАВЕРШЕНА (общее время: 0 секунд)