



# **Δομές Δεδομένων & Ανάλυση Αλγορίθμων (Εργ.)**

**Δημοσθένης Σταμάτης**

**Ενδεικτικές Απαντήσεις σε θέματα Εξέτασης  
της 1<sup>ης</sup> Προόδου του εργαστηριακού μαθήματος**

### ΘΕΜΑ 1. (5 μονάδες)

Θεωρήστε ότι δίνεται η δομή δεδομένων **στοίβα (stack)**, η οποία είναι υλοποιημένη με τη βοήθεια πίνακα (κλάση `ArrayStack` της Εργ. Ασκήσης 2). Θεωρήστε επίσης ότι έχει δημιουργηθεί μία στοίβα στην οποία έχει εισαχθεί ένας αριθμός από δεδομένα τύπου φοιτητή. Να γράψετε μία μέθοδο **count8s** η οποία να μετρά πόσοι από τους φοιτητές που βρίσκονται στη στοίβα έχουν βαθμολογηθεί με 8. Δώστε ένα πλήρες πρόγραμμα για τον έλεγχο της ορθής λειτουργίας της μεθόδου σας.

**Περιορισμοί:** (α) Η στοίβα μετά την ολοκλήρωση της καταμέτρησης θα πρέπει να έχει ακριβώς την ίδια μορφή με την αρχική. (β) Δεν επιτρέπεται να τροποποιήσετε την κλάση υλοποίησης της στοίβας `ArrayStack`.

```
public static int count8s(ArrayStack S) {
    int count=0;
    int size=S.size();
    ArrayStack SB= new ArrayStack(size);
    for(int i=0;i<size;i++) {
        Object obj = S.top();
        if (((Student)obj).getVathmos()>=8) count++;
        SB.push(S.pop());
    }
    for(int i=0;i<size;i++) {
        S.push(SB.pop());
    }
    return count;
}

public static void main(String[] args) {
    Student S1 = new Student(111,"onoma1",1,9);
    Student S2 = new Student(222,"onoma2",2,6);
    Student S3 = new Student(333,"onoma3",0,8);
    Student S4 = new Student(444,"onoma4",3,4);
    ArrayStack ST=new ArrayStack();
    ST.push(S1);
    ST.push(S2);
    ST.push(S3);
    ST.push(S4);
    System.out.println(count8s(ST));
}
```

## **ΘΕΜΑ 2. (5 μονάδες)**

Στην κλάση της συνδεδεμένης λίστας (της Εργ. Άσκηση 4), προσθέστε μία μέθοδο η οποία να διαγράφει το δεύτερο κόμβο, εφόσον αυτός υπάρχει, και να επιστρέφει το περιεχόμενό του. Αν ο κόμβος δεν υπάρχει θα πρέπει η μέθοδος να ρίχνει σχετική εξαίρεση. Δώστε ένα πλήρες πρόγραμμα για τον έλεγχο της ορθής λειτουργίας της μεθόδου σας.

```
public Object removeSecond() throws NoSuchElementException {  
    if (size() < 2)  
        throw new NoSuchElementException("List with less than 2 elements");  
    Object obj1 = removeFirst();  
    Object obj2 = removeFirst();  
    insertFirst(obj1);  
    return obj2;  
}
```

```
public static void main(String[] args) {  
    Student S1 = new Student(111, "onoma1", 1, 9);  
    Student S2 = new Student(222, "onoma2", 2, 6);  
    Student S3 = new Student(333, "onoma3", 0, 8);  
    Student S4 = new Student(444, "onoma4", 3, 4);  
  
    LinkedList L = new LinkedList();  
    L.insertLast(S1);  
    L.insertLast(S2);  
    L.insertLast(S3);  
    L.insertLast(S4);  
  
    Student R = (Student) L.removeSecond();  
    System.out.println(R.toString());  
}
```

### ΘΕΜΑ 1C. (5 μονάδες)

Θεωρήστε ότι δίνεται η δομή δεδομένων **συνδεδεμένη λίστα (linked list)** (όπως είναι υλοποιημένη στην Εργ. Άσκηση 4). Θεωρήστε επίσης ότι έχει δημιουργηθεί μία συνδεδεμένη λίστα στην οποία έχει εισαχθεί ένας αριθμός από δεδομένα τύπου φοιτητή. Να γράψετε μία μέθοδο **succeeded** η οποία να μετρά πόσοι από τους φοιτητές που βρίσκονται στη ~~σείρα~~ **συνδεδεμένη λίστα** έχουν περάσει το μάθημα (βαθμός  $\geq 5$ ). Δώστε ένα πλήρες πρόγραμμα για τον έλεγχο της ορθής λειτουργίας της μεθόδου σας.

**Περιορισμοί:** (α) Η λίστα μετά την ολοκλήρωση της καταμέτρησης θα πρέπει να έχει ακριβώς την ίδια μορφή με την αρχική. (β) Δεν επιτρέπεται να τροποποιήσετε την κλάση υλοποίησης της συνδεδεμένης λίστας.

```
public static int succeeded(LinkedList L){
    SLListNode current = L.getFirst();
    int count = 0;
    while(current != null){
        if(((Student)current.getNodeData()).getVathmos() >= 5)
            count++;
        current = current.getNextNode();
    }
    return count;
}

public static void main(String[] args) {
    Student S1 = new Student(111, "onoma1", 1, 3);
    Student S2 = new Student(222, "onoma2", 2, 6);
    Student S3 = new Student(333, "onoma3", 0, 8);
    Student S4 = new Student(444, "onoma4", 3, 8);

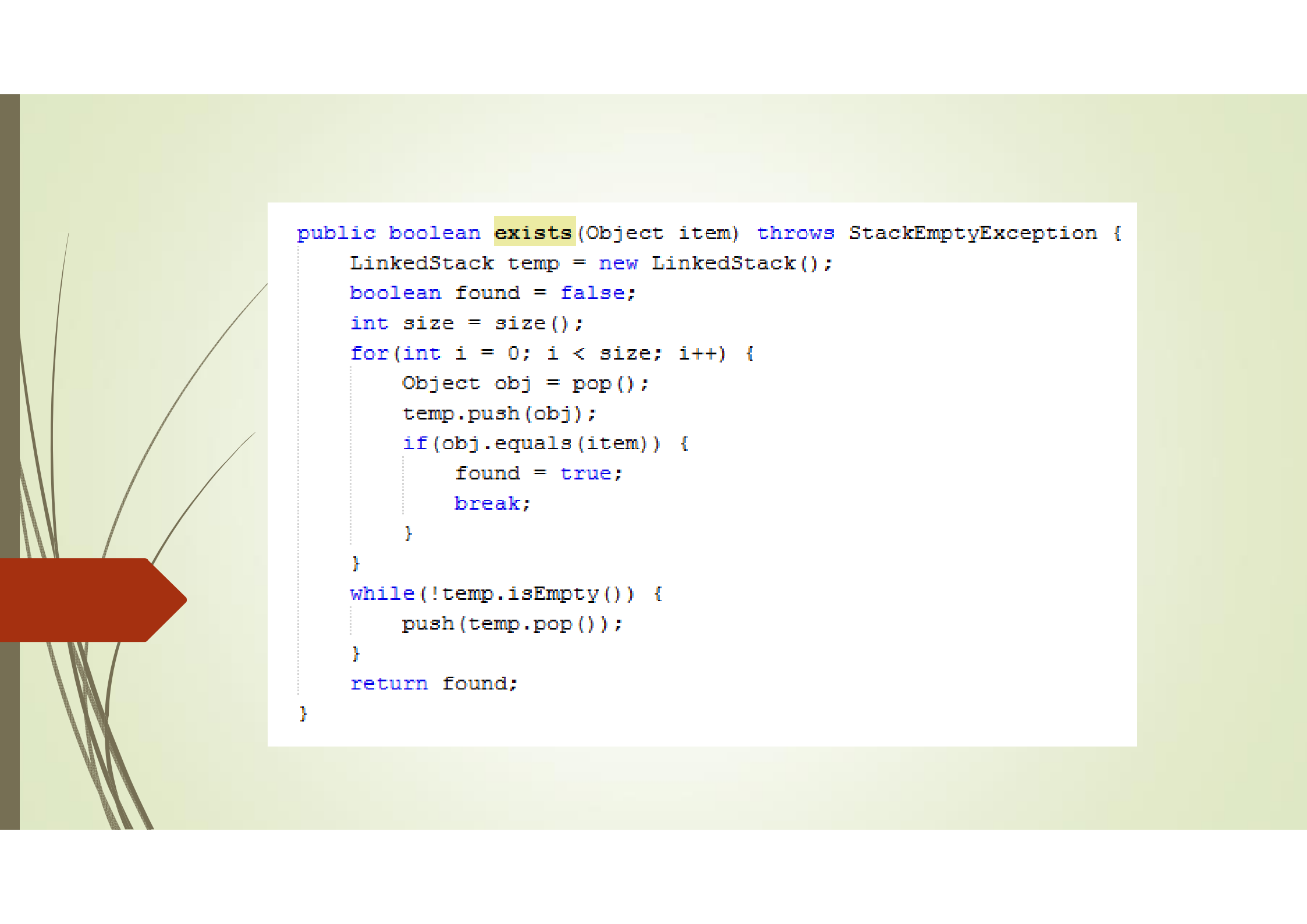
    LinkedList L = new LinkedList();
    L.insertLast(S1);
    L.insertLast(S2);
    L.insertLast(S3);
    L.insertLast(S4);

    System.out.println(succeeded(L));
}
```

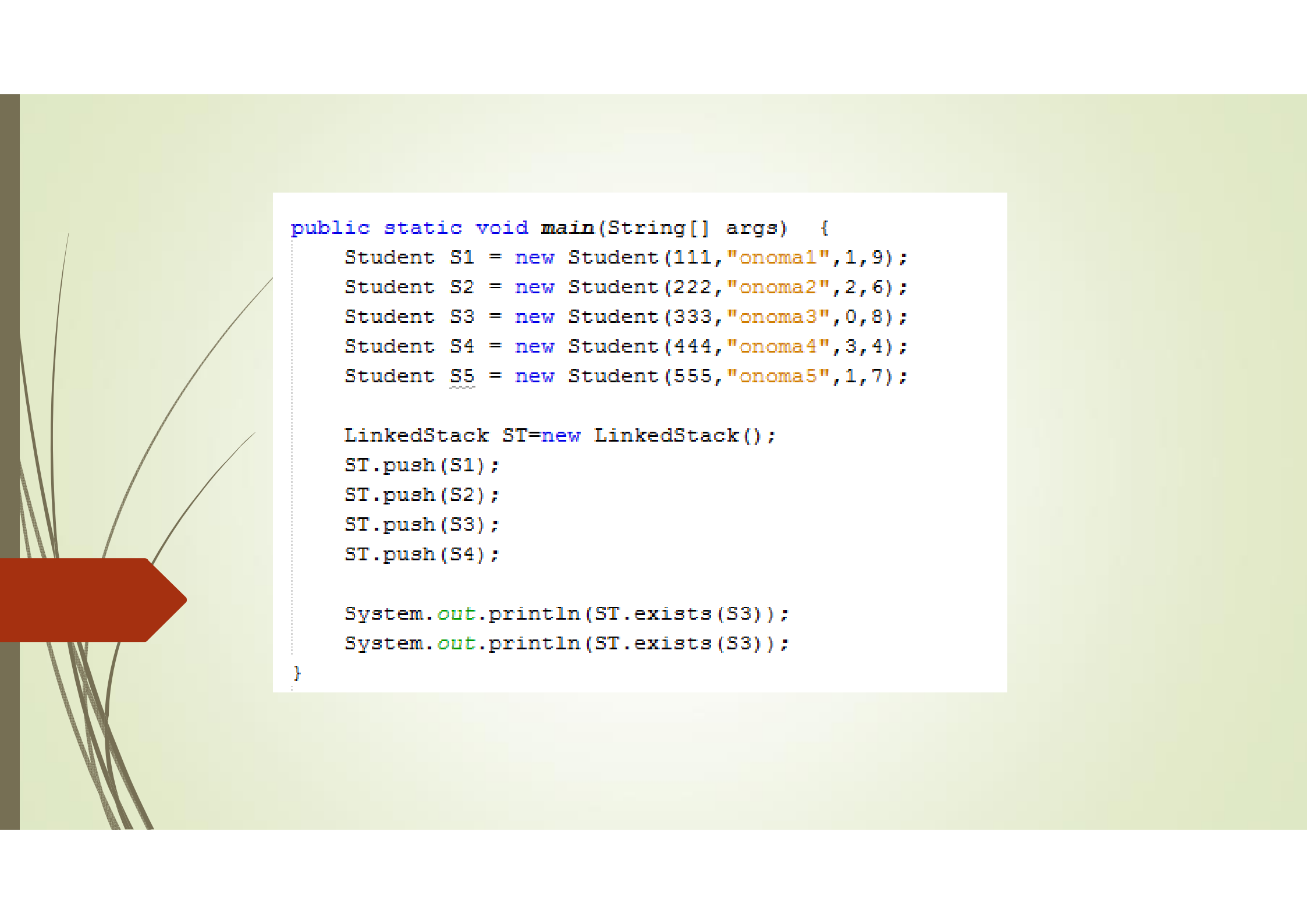
### ΘΕΜΑ 2C. (5 μονάδες)

Θεωρήστε ότι δίνεται η δομή δεδομένων **στοίβα (stack)**, η οποία είναι υλοποιημένη με τη βοήθεια συνδεδεμένης λίστας (σχετική κλάση της Εργ. Ασκήσης 5). Στην κλάση υλοποίησης προσθέστε μία **boolean μέθοδο** η οποία δοθέντος ενός αντικειμένου να επιστρέφει true εάν αυτό υπάρχει μέσα στη στοίβα και false στην αντίθετη περίπτωση. Δώστε ένα πλήρες πρόγραμμα για τον έλεγχο της ορθής λειτουργίας της μεθόδου σας.





```
public boolean exists(Object item) throws StackEmptyException {  
    LinkedStack temp = new LinkedStack();  
    boolean found = false;  
    int size = size();  
    for(int i = 0; i < size; i++) {  
        Object obj = pop();  
        temp.push(obj);  
        if(obj.equals(item)) {  
            found = true;  
            break;  
        }  
    }  
    while(!temp.isEmpty()) {  
        push(temp.pop());  
    }  
    return found;  
}
```



```
public static void main(String[] args)  {
    Student S1 = new Student(111,"onoma1",1,9);
    Student S2 = new Student(222,"onoma2",2,6);
    Student S3 = new Student(333,"onoma3",0,8);
    Student S4 = new Student(444,"onoma4",3,4);
    Student S5 = new Student(555,"onoma5",1,7);

    LinkedStack ST=new LinkedStack();
    ST.push(S1);
    ST.push(S2);
    ST.push(S3);
    ST.push(S4);

    System.out.println(ST.exists(S3));
    System.out.println(ST.exists(S3));
}
```