

Πανεπιστήμιο Κρήτης
Τμήμα Επιστήμης Υπολογιστών

HY-252 – Αντικειμενοστρεφής Προγραμματισμός
Βασίλης Χριστοφίδης

Επαναληπτική Εξέταση (3 ώρες)
Ημερομηνία: 2 Σεπτεμβρίου 2005

Όνοματεπώνυμο:
Αριθμός Μητρώου:

Άσκηση 1 (10 μονάδες)

Σας δίνονται οι παρακάτω δηλώσεις δύο διεπαφών (IK και IM) και τριών κλάσεων (A, B, και C) Java, οι οποίες υλοποιούν κάποιες από τις προηγούμενες διεπαφές:

```
public interface IK {  
    abstract public void k();  
}  
public interface IM {  
    abstract public void m();  
}  
public class A implements IK, IM {  
    public void k() { System.out.println("A: k"); }  
    public void m() { System.out.println("A: m"); }  
}  
public class B implements IM {  
    public void k() { System.out.println("B: k"); }  
    public void m() { System.out.println("B: m"); }  
}  
public class C extends B implements IK {  
    public void k() { System.out.println("C: k"); }  
}
```

1.1 (5 μονάδες) Ποιες από τις παρακάτω εντολές, που εμπλέκουν αναφορές σε διεπαφές (interface references), θεωρούνται σαν λάθη από τον μεταφραστή της Java (compile-time errors); Υπογραμμίστε τις λάθος εντολές και δικαιολογήστε σύντομα την απάντησή σας.

```
IK t; // interface reference  
t = new A();  
t = new B();  
t = new C();  
t.k();  
t.m();
```

Λύση:

To make use of interfaces, Java introduces *interface references*, which are like object references, except that an interface reference of type *I* is allowed to point to objects of *any* type *C* such that the class *C* implements the interface *I*. For example, the interface reference *t* of type *IK* (introduced above) is

allowed to point to an object of type A or C but not of type B, even though B happens to contain a method k():

```
IK t;
t = new A(); // LEGAL
t = new B(); // ILLEGAL
t = new C(); // LEGAL
t.k(); // LEGAL (regardless of dynamic type of t)
t.m(); // ILLEGAL (regardless of dynamic type of t)
```

1.2 (5 μονάδες) Τι θα τυπωθεί μετά την εκτέλεση του παρακάτω κώδικα, που εμπλέκει κλήσεις σε μεθόδους διεπαφών (interface method call), όταν η μεταβλητή x==1; Δικαιολογήστε σύντομα την απάντησή σας.

```
IK ik; // interface reference
B b; // object reference
if (x==1) {
    b = new B();
    ik = new A();
} else {
    b = new C();
    ik = new C();
}
b.k(); // virtual method call
ik.k(); // interface method call
```

Λύση:

Compiling Java interfaces boils down to supporting a new kind of method call, named *interface method call*. This call is like the (non-static) *virtual method call*, except that the object on which you invoke a method comes from an interface reference, as opposed to an object reference. When x==1, the program outputs

```
B: k
A: k
```

Άσκηση 2 (20 μονάδες)

Η παρακάτω κλάση Pen φιλοδοξεί να ζωγραφίζει σε μια συσκευή (device) γραφικής παράστασης, χρησιμοποιώντας αυθαίρετο πλάτος και ύψος εικονοκυττάρων (pixel). Κάθε στιγμιότυπο της κλάσης χρησιμοποιεί ένα σημείο (point) για να γνωρίζει την ακριβή της θέση στη συσκευή γραφικής παράστασης. Η κλάση PenDriver προορίζεται να χρησιμοποιεί την Pen, αλλά όταν εκτελέσουμε την κύρια μέθοδό της εκτυπώνεται μόνο "pen[030] @ (10, 30)". Υπάρχουν τουλάχιστον τέσσερα λογικά λάθη στον ακόλουθο κώδικα. Βάλτε σε έναν κύκλο τις λάθος εντολές και προτείνετε κατάλληλες διορθώσεις τους.

```
/** Represents a point in the graphical plane. */
public class Point {
    int x;
    int y;
    /** Default constructor. */
    Point(int x, int y) {
        this.x = x;
        this.y = y; }
}
```

```

    public String toString () {
        return "(" + x + ", " + y + ")"; }
}

/**Represents a pen that has width&height in pixels. */
public class Pen extends Point {
    static int width; /** Each pen knows its width. */
    static int height; /** Each pen knows its height. */
    Point loc; /** Each pen knows its location. */
    /** Constructor is given initial (w,h) values. */
    public Pen(int w, int h) {
        super (w,h); }
    /** Return string for object, like "pen[3x7] @ point
        (10,30)" */
    public String toString() {
        return "pen[" + width + x + height + "] @ " + loc; }
}

public class PenDriver {
    public static void main(String[] args) {
        /** Construct Pen to be 3x7 @ point (10,30) */
        Pen p = new Pen (3,7);
        p.loc = new Point (10,30);
        System.out.println (p); }
}

```

Λύση:

1. [8 points] In Pen class, the use of 'static' with width and height is inappropriate. This would enforce all Pen objects to have the same width and height, and the intent of the programmer is clearly to have this information associated with each Pen object.
2. [4 points] If Pen extends Point, then it shouldn't have a loc field, since this information would be inherited as (x, y) from Point
3. [4 points] If Pen extends Point, then the constructor taking (w, h) should not simply call super; rather, it should set width and height as appropriate.
4. [4 points] In the Pen class, the toString() implementation has variable x referenced when it should be the string "x" or the character 'x'. The only reason this code works is because (miraculously) the Pen class inherits instance variable 'x' from Point.

Άσκηση 3 (10 μονάδες)

Η Java ορίζει την διεπαφή Comparable ως ακολούθως:

```

interface Comparable {
    int compareTo(Object);
}

```

όπου η κλήση `x.compareTo(y)` πρέπει να επιστρέφει `-1` εάν το `x` είναι μικρότερο από το `y`, `0` εάν είναι ίσα και `1` στην αντίθετη περίπτωση. Ορίστε την κλάση `MyArray`, που υλοποιεί την διεπαφή `Comparable`, της οποίας τα αντικείμενα συμπεριφέρονται σαν πίνακες ακεραίων. Τα στιγμιότυπα της `MyArray` θα πρέπει να συγκρίνονται χρησιμοποιώντας το άθροισμα των στοιχείων τους. Για παράδειγμα:

```
//create an array and initialize the elements
int[] a = new int[] {1, 2, 3, 4};
int[] b = new int[] {-1, 2, -3, 4, -5};
//the elements of m1 are those of a
MyArray m1 = new MyArray(a);
//the elements of m2 are those of b
MyArray m2 = new MyArray(b);
//prints 1, since 1+2+3+4 > -1+2-3+4-5
System.out.println(m1.compareTo(m2));
```

Υλοποιήστε την μέθοδο `compareTo()` της κλάσης `MyArray`, έτσι ώστε ο παραπάνω κώδικας να δουλεύει σωστά.

Λύση:

```
class MyArray implements Comparable {
    private int[] a;
    MyArray(int[] b) { a = b; }
    public int compareTo(Object o) {
        MyArray other = (MyArray) o;
        int mysum = 0;
        int othersum = 0;
        for(int i=0; i<a.length; i++)
            mysum = mysum + a[i];
        for(int i=0; i<other.a.length; i++)
            othersum = othersum + other.a[i];
        if (mysum < othersum) return -1;
        else if (mysum == othersum) return 0;
        else return 1;
    }
}
```

Άσκηση 4 (25 μονάδες)

Μια ταξινομημένη ακολουθία αγγλικών λέξεων πρόκειται να υποβληθεί σε επεξεργασία από ένα πρόγραμμα Java. Το πρόγραμμα θα πρέπει να υποστηρίζει διάφορες λειτουργίες, όπως η ανάκτηση μιας λέξης από μια συγκεκριμένη θέση της ακολουθίας, η αφαίρεση μια λέξης και προσθήκη μιας νέας λέξης. Σε κάθε λειτουργία η ακολουθία θα πρέπει να διατηρείται πάντα ταξινομημένη.

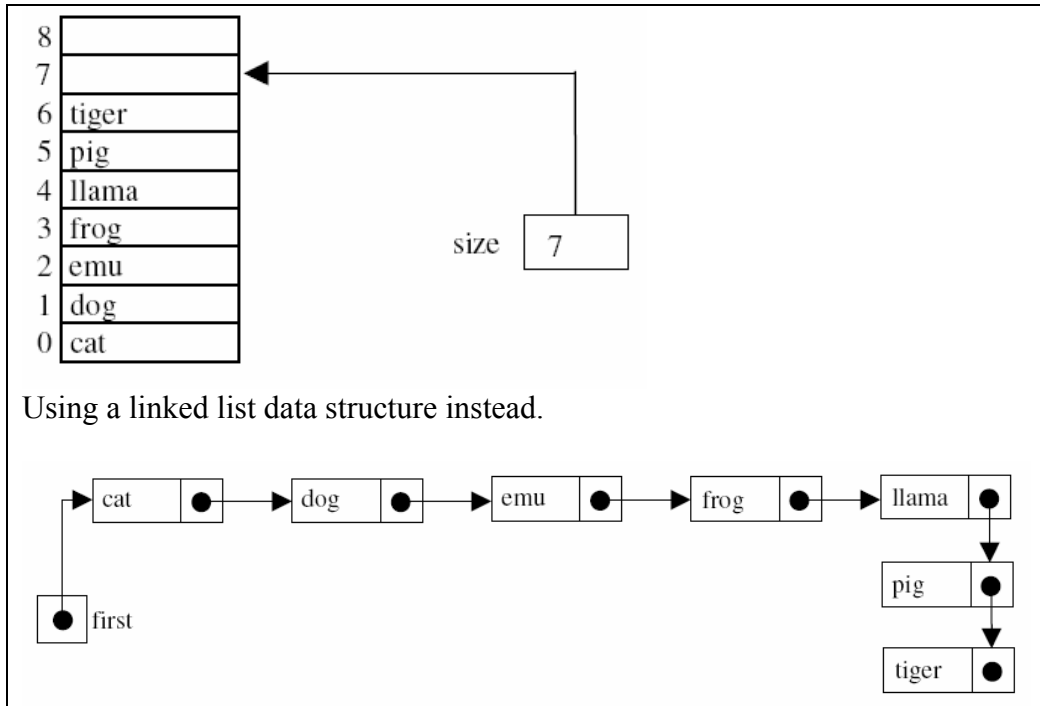
4.1 (8 μονάδες) Χρησιμοποιώντας κατάλληλα γραφήματα, εξηγήστε πώς

- ένας πίνακας
- μια συνδεδεμένη λίστα

θα μπορούσαν να χρησιμοποιηθούν ως δομή δεδομένων για να αποθηκεύσουν μια ταξινομημένη ακολουθία λέξεων.

Λύση:

The following illustration shows how the sequence of words (cat, dog, emu, frog, llama, pig, tiger) would be stored using an array data structure.



4.2 (9 μονάδες) Συγκρίνετε την αποδοτικότητα των δύο προηγούμενων υλοποιήσεων μιας ταξινομημένης ακολουθίας αγγλικών λέξεων, σχετικά με τις λειτουργίες που ανακτούν, που αφαιρούν και που προσθέτουν μια λέξη.

Λύση:		
Operation	Array Structure	Linked List Structure
Retrieve word at given position	Most efficient $O(1)$ operation, since direct access structure	Least efficient $O(n)$ operation, since sequential access structure
Remove a word	Least efficient $O(n)$ operation, since words above the one removed must be shifted down	Most efficient, since although accessing the word may be $O(n)$, the actual deletion is $O(1)$, merely requiring the adjustment of one link
Add a word	Least efficient $O(n)$ operation, since words above the one inserted must be shifted up	Most efficient, since although accessing the position of the word may be $O(n)$, the actual insertion is $O(1)$, merely requiring the creation of a node and the adjustment of two links

4.3 (8 μονάδες) Για κάθε μια από τις προηγούμενες υλοποιήσεις, γράψτε τα τμήματα του κώδικα της Java που απαιτούνται για να βρούμε την θέση μιας δεδομένης λέξης W σε μια ακολουθία, ή -1 εάν η λέξη δεν είναι εκεί (υποθέστε ότι η πρώτη λέξη βρίσκεται στην θέση 0). Δηλώστε οποιεσδήποτε επιπλέον υποθέσεις κάνετε κατά την υλοποίησή σας.

Σημείωση: Δεν απαιτείται η πλήρη υλοποίηση των μεθόδων.

Λύση:

For the array representation, assuming name of array is a.

```
int ordPos = -1;
for (int i=0; i<size; i++)
if (w.equals(a[i])) {
ordPos = i; break;
}
```

For the linked list representation, assuming SLLNode has fields Object element and SLLNode succ.

```
int ordPos = -1, count = 0;
for (SLLNode n = first; n!=null; n = n.succ)
if (w.equals(n.element)) {
ordPos = count; break;
} else
count++;
```

Άσκηση 5 (20 μονάδες)

Πολλά παιχνίδια υπολογιστών χρησιμοποιούν ένα ζάρι (die) για να παράγουν έναν τυχαίο ακέραιο μεταξύ του 1 και του 6 συμπεριλαμβανομένου. Ένα ζάρι είναι ουσιαστικά ένας κύβος, όπου στην κάθε επιφάνειά του εμφανίζεται ένας από τους πρώτους έξι ακέραιους αριθμούς. Όταν περιστρέφεται καταλήγει να στηριχτεί σε μία από αυτές τις επιφάνειες: ο αριθμός που εμφανίζεται στην πάνω επιφάνεια από αυτήν που στηρίζεται είναι το αποτέλεσμα μιας ζαριάς.

5.1 (10 μονάδες) Δώστε το συμβόλαιο του ΑΤΔ Ζάρι (Die) με τις υπογραφές, τις εκ των προτέρων και εκ των υστέρων συνθήκες κάθε λειτουργίας καθώς και τις αμετάβλητες συνθήκες (περιορίστε την προδιαγραφή σας μόνο σε εκείνες τις λειτουργίες που είναι απαραίτητες).

Λύση:

For a die ADT we essentially need operations to create a new die, read the current value of the die, and throw the die. Optionally we may include a 'cheat' operation, to set the die to a given value, but this is not necessary (and won't be included here).

Specification

Associated ADTs: Integer

Operations:

Create: → Die

Value: Die → Integer

Throw: Die → Die

Invariants:

For every d , $0 \leq Value(d) \leq 6$

For every d , $Value(d)$ and $Value(Throw(d))$ must be independent

5.2 (10 μονάδες) Γράψτε μια κλάση Java, η οποία υλοποιεί το συμβόλαιο του ΑΤΔ Ζάρι (Die).

Σημείωση: Χρησιμοποιήστε την `java.util.Random`.

Λύση:

```
// Implementation of a die ADT
public class Die {
    private int faceValue;
    private java.util.Random randomSequence;
    public Die() { // constructor
        randomSequence = new java.util.Random();
        faceValue = randomSequence.nextInt(6) + 1;
    }
    public int value() { // returns face value
        return faceValue;
    }
    public void throw() { // throws this die
        faceValue = randomSequence.nextInt(6) + 1;
    }
}
```

Άσκηση 6 (20 μονάδες)

Ο κώδικας Java της άσκησης σκιαγραφεί την υλοποίηση ενός applet, που προορίζεται να εμφανίσει ένα χρονόμετρο (Stopwatch) χρησιμοποιώντας μια κλάση ρολόι (Clock), που χειρίζεται τις ώρες, τα λεπτά και τα δευτερόλεπτα. Το χρονόμετρο παρουσιάζει μια συμβολοσειρά της μορφής ώρες:λεπτά:δευτερόλεπτα στο applet στη θέση (10, 20). Το χρονόμετρο θα πρέπει να υλοποιεί την ακόλουθη συμπεριφορά:

- 1) Ο χρόνος πρέπει να ενημερώνεται κάθε δευτερόλεπτο κατά την διάρκεια ενεργοποίησης του χρονομέτρου.
- 2) Το χρονόμετρο θα πρέπει να ενεργοποιείται όταν το ποντίκι φεύγει από το applet.
- 3) Το χρονόμετρο θα πρέπει να απενεργοποιείται όταν το ποντίκι μπαίνει στο applet.
- 4) Το χρονόμετρο θα πρέπει να επαναρυθμίζεται στον χρόνο 00:00:00 όταν το ποντίκι πατιέται μέσα στο applet.

Η κλάση Ρολόι καθορίζεται ως εξής:

```
public class Clock {
    // Private instance variables not shown
    // Creates a Clock initialized to 00:00:00
    public Clock()
    {
        // implementation not shown
    }
    // Increases the time by a second, wrapping if necessary
    public void tick()
    {
        // implementation not shown
    }
    // Sets the clock to the time specified by hour,
    // minutes, and seconds.
    public void set(int hours, int minutes, int seconds)
    {
        // implementation not shown
    }
}
```

```

//Returns the time as hours:minutes:seconds
    public String toString()
    {
        // implementation not shown
    }
}

```

Γράψτε τον απαραίτητο κώδικα των μεθόδων της κλάσης Stopwatch, ώστε το applet να συμπεριφέρεται όπως περιγράφηκε προηγουμένως (συμπληρώστε μόνο το σώμα των μεθόδων που σας δίνονται στην συνέχεια).

```

import java.awt.*;
import java.applet.Applet;
import javax.swing.Timer;
import java.awt.event.*;

```

```

public class Stopwatch extends Applet implements
    MouseListener, ActionListener {
    Clock sw; // Holds the stopwatch object
    Timer timer; // A timer that should respond every second.
    public void init()
    {

```

Λύση:

```

sw = new Clock();
addMouseListener(this);
timer = new Timer(DELAY, this);

```

```

    }
    public void paint(Graphics page)
    {

```

Λύση:

```

page.drawString(sw.toString(), 10, 20);

```

```

    }
    public void actionPerformed(ActionEvent event)
    {

```

Λύση:

```

sw.tick();
repaint();

```

```

    }
    public void mouseClicked(MouseEvent event)
    {

```

Λύση:

```

sw.set(0, 0, 0);
repaint();

```

```

    }
    public void mouseExited(MouseEvent event)
    {

```

Λύση:

```

timer.start();

```

```

    }

```

```
public void mouseEntered(MouseEvent event)
{
```

```
    Λύση:  
    timer.stop();
```

```
    }  
public void mousePressed(MouseEvent event) {}  
public void mouseReleased(MouseEvent event) {}  
}
```