Πανεπιστήμιο Κρήτης
Τμήμα Επιστήμης Υπολογιστών

ΗΥ-252 – Αντικειμενοστραφής Προγραμματισμός
Βασίλης Χριστοφίδης

Πρόοδος (3 ώρες)
Ημερομηνία: 20 Νοεμβρίου 2004

Ονοματεπώνυμο:
Αριθμός Μητρώου:

Άσκηση 1 (10 μονάδες)
Σας δίνεται ο παρακάτω κώδικας Java ο οποίος αναλύει ένα αρχείο κειμένου με το όνομα "input.txt" και τυπώνει στην έξοδο κατάλληλα μηνύματα ανάλογα με το περιεχόμενο του αρχείου:

```
import java.io.*;

public class ParseFarce {
 public static void main(String args[]) {
   try {
       double result1 = 0;
       double result2 = 0;
       StreamTokenizer st = new StreamTokenizer(new
                                          FileReader("input.txt"));
       while (st.nextToken != st.TT_EOF) {
         if ((st.ttype==st.TT_WORD)&&(st.sval.compareTo("foo")==0))
            {System.out.println(""+result1);
             result1 = 0;
            }
         else if (st.ttype==st.TT_NUMBER)
            {result1 = result1 + st.nval;
             result2 = result2 + st.nval;
            }
         }
       System.out.println("finished: " + result2);
      } catch (IOException ioe) {
       System.out.println("Error parsing file input.txt" + e);
      }
   }
}
```

Τι θα τυπωθεί στην στάνταρ έξοδο του προγράμματος ParceFarce, όταν το αρχείο "input.txt" περιέχει το παρακάτω κείμενο:

```
CS 150 and CS 252 are courses in which the word foo appears at least 10 times
each semester.  Every year, at least 6 students ask the origin of foo, but
the answer is widely debated.  There are only 1 or 2 people in the world who
know true origin of foo, and I am fortunately not 1 of them.  Of course,
the preceding is utter nonsence and should not be taken seriously by any
self-respecting person.
```

Λύση:
The output would be
402
16
3
finished: 412

Άσκηση 2 (10 μονάδες)

Σας δίνεται ο παρακάτω ατελής κώδικας Java για τον χειρισμό ενός πίνακα (array) με αλφαριθμητικά (`String`) στον οποίο αναφερόμαστε μέσω της μεταβλητής αντικειμένων με όνομα **`animals`**. Συμπληρώστε τον κώδικα έτσι ώστε:

1. Να δημιουργείται ένα αντικείμενο πίνακας (αναφορά **`animals`**) που περιέχει τα ακόλουθα αλφαριθμητικά: **`lion`**, **`tiger`**, **`bear`**, **`goat`**, και **`horse`**.
2. Να γίνεται έλεγχος εάν το στοιχείο του πίνακα **`animals`** που βρίσκεται στην θέση **`0`** ισούται με το στοιχείο που βρίσκεται στην θέση **`3`** και να τυπώνετε το αποτέλεσμα της σύγκρισης.
3. Να ανακτάται το στοιχείο του πίνακα **`animals`** το οποίο έχει λεξικογραφικά το μεγαλύτερο όνομα.

Τι θα τυπωθεί στην στάνταρ έξοδο του προγράμματος μετά την συμπλήρωση των γραμμών κώδικα που λείπουν?

Σημείωση: Για την λεξικογραφική σύγκριση των στοιχείων του πίνακα δεν απαιτείται η χρησιμοποίηση κάποιας μεθόδου ταξινόμησης. Αντιθέτως, πρέπει να χρησιμοποιήσετε τις μεθόδους σύγκρισης αλφαριθμητικών που ορίζονται στην κλάση `String`.

```
// File StringTest.java
public class StringTest {
   public static void main (String[] args) {
      // Exam 2 question 1
```

Λύση:
```
String[] animals= {"lion", "tiger", "bear", "goat", "horse"};
```

```
      printArray("Question 2.1", animals);
      // Exam 2 question 2
```

Λύση:
```
System.out.print("Question 2.2:  Positions 0 and 3 are ");
if (animals[0].equals(animals[3]))
    System.out.println("equal");
else
    System.out.println("not equal");
```

```
      // Exam 2 question 3
```

Λύση:
```
String last = animals[0];
for (int i = 1; i < animals.length; i++)
     if (last.compareTo(animals[i]) < 0) last = animals[i];
System.out.println("Question 2.3:  Last animal: " + last);
Alternative solution
int index = 0;
for (int n = 1; n < animals.length; n++)
     if (animals[n].compareTo(animals[index]) > 0) index = n;
System.out.println("Question 2.3:  Last animal: "+
                                          animals[index]);
```

```
   }
   private static void printArray(String heading, String[] f) {
      System.out.print(heading + ":  Array animals: ");
      for (int i = 0; i < f.length; i++)
         System.out.print(f[i] + " ");
      System.out.println();
   }
}
```

Λύση:

Recall that **str1.compareTo(str2)** returns a number < 0 in case **str1** comes before **str2** in alphabetic order, and returns a number > 0 in case **str1** comes after **str2** in alphabetic order.
Here is sample output:

```
Question 2.1:  Array animals: lion tiger bear goat horse
Question 2.2:  Positions 0 and 3 are not equal
Question 2.3:  Last animal: tiger
```

Άσκηση 3 (10 μονάδες)

Στους παρακάτω ορισμούς κλάσεων Java υπάρχουν τουλάχιστον 10 (συντακτικά ή τύπων) λάθη.

```
public class ExamBuggleWorld extends BuggleWorld {          // line 1
                                                             // line 2
      public void run () {                                   // line 3
            Color c = Color.cyan();                          // line 4
            int n = 4                                        // line 5
            ExamBuggle emma = ExamBuggle();                  // line 6
            emma.mystery1(c,n);                              // line 7
            emma.mystery1(3,Color.red);                      // line 8
            boolean answer = emma.mystery2();                // line 9
            this.mystery3();                                 // line 10
            }                                                // line 11
}                                                            // line 12
                                                             // line 13
class ExamBuggle extends Buggle {                            // line 14
                                                             // line 15
      public void mystery1(Color c, int n1) {                // line 16
            n2 = n1 + 1;                                     // line 17
            this.setColor(Color.c);                          // line 18
            forward(n2);                                     // line 19
            this.dropBagel();                                // line 20
                                                             // line 21
      public boolean mystery2() {                            // line 22
            this.isOverBagel();                              // line 23
            }                                                // line 24
                                                             // line 25
      public mystery3() {                                    // line 26
            this.dropBagel();                                // line 27
            }                                                // line 28
                                                             // line 29
}                                                            // line 30
```

Αρ. Λάθους	Αρ. Γραμμής	Σύντομη περιγραφή του λάθους	Διορθωμένη γραμμή
1	4	Color.cyan() is not a method invocation	`Color c = Color.cyan;`
2	5	The local variable declaration int n = 4 is missing a semi-colon at the end	`int n = 4;`
3	6	There is a missing new in the constructor method invocation that creates an ExamBuggle	`ExamBuggle emma = new ExamBuggle();`
4	8	The two arguments of the instance method invocation emma.mystery1(3,Color.red) are in the wrong order	`emma.mystery1(Color.red,3);`
5	10	In this.mystery3(), this stands for an instance of ExamBuggleWorld, which does not understand the mystery3 message; the recipient should be an instance of ExamBuggle	`emma.mystery3();`
6	17	The local variable declaration n2 = n1 + 1; is missing a type for the contents of the variable	`int n2 = n1 + 1;`
7	18	Color.c attempts to reference a non-existent class constant rather than the parameter c	`this.setColor(c);`
8	21	The instance method declaration for mystery1 is missing a close squiggly brace	`}`
9	23	The non-void method mystery2 is missing a return statement.	`return this.isOverBagel();`
10	26	The method header for mystery3 is missing the return type, void	`public void mystery3() {`

Για κάθε ένα από τα λάθη τα οποία εμφανίζονται σε διαφορετικές γραμμές του παραπάνω κώδικα συμπληρώστε τον πίνακα που σας δίνεται με τα παρακάτω στοιχεία:
1. Αριθμός γραμμής του λάθους,
2. Σύντομη περιγραφή του λάθους
3. Διόρθωση του λάθους.

Σημείωση: Μπορείτε να περιγράψετε τα λάθη με οποιαδήποτε σειρά (δηλ όχι αναγκαστικά με την σειρά εμφάνισής τους στο πρόγραμμα). Θυμηθείτε ότι η κλάση `Color` έχει σαν στατικά μέλη δεδομένων όλα τα βασικά χρώματα (red, gray, etc.) που ορίζονται στον χώρο χρωμάτων sRGB.

Άσκηση 4 (30 μονάδες)

Σας δίνεται το παρακάτω συμβόλαιο του ΑΤΔ Διατεταγμένη Ακολουθία (`Ranked Sequence`) με την μορφή μιας διεπαφής Java:

```
public interface RankedSeq {
public int size();
public boolean isEmpty();
public void insertElemAtRank(int r, Object e);
public void removeElemAtRank(int r);
public void reverse();
}
```

1) **(10 μονάδες)** Δώστε σε μορφή ψευδοκώδικα τον αλγόριθμο υλοποίησης της μεθόδου **`reverse`** χρησιμοποιώντας μόνο μεθόδους που ορίζονται στο συμβόλαιο του ΑΤΔ Διατεταγμένη Ακολουθία.

Λύση:

```
// Reverses the elements of a Ranked Sequence object
public void reverse() {
  int n = size();
  for (int r = 0; r<= n-2/2; r++ ;){
      e = removeElemAtRank(r);
      insertElemAtRank(n-1-r, e);
      e = removeElemAtRank(n-2-r);
      insertElemAtRank(r, e);
     }
}
```

2) **(10 μονάδες)** Ποια είναι η πολυπλοκότητα (complexity) του χρόνου εκτέλεσης (στην χειρότερη περίπτωση) της μεθόδου **`reverse`**, που υλοποιήσατε στο ερώτημα 1 όταν ο ΑΤΔ Διατεταγμένη Ακολουθία υλοποιείται χρησιμοποιώντας έναν πίνακα; Δικαιολογήστε σύντομα την απάντησή σας.

Λύση:

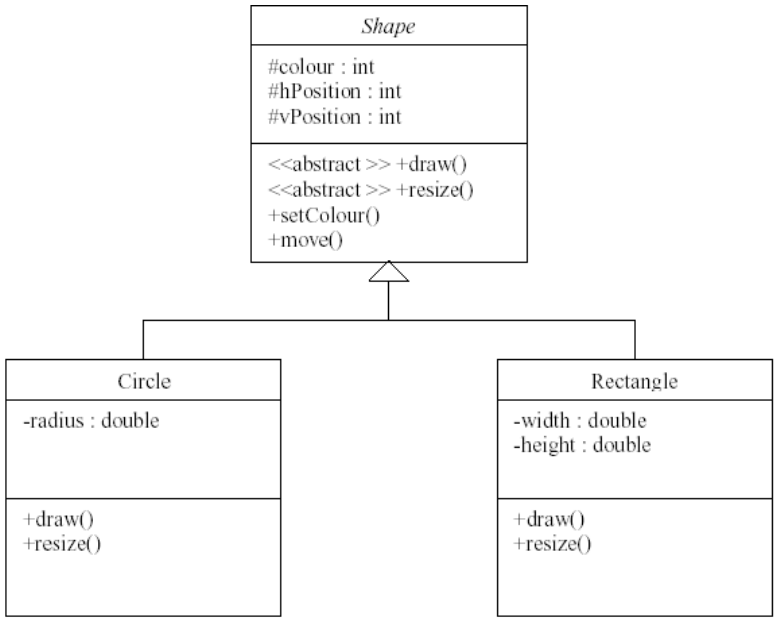
With an array implementation, each method called by `reverse()` takes $O(n)$ time because order-n array elements have to be shifted to make or remove space in the array. The for loop is executed on the order of n times, so the total running time of `reverse()` is $O(n^2)$.

3) **(10 μονάδες)** Ποια είναι η πολυπλοκότητα (complexity) του χρόνου εκτέλεσης (στην χειρότερη περίπτωση) της μεθόδου **`reverse`**, που υλοποιήσατε στο ερώτημα 1 όταν ο ΑΤΔ Διατεταγμένη Ακολουθία υλοποιείται χρησιμοποιώντας μια διπλά συνδεδεμένη λίστα; Δικαιολογήστε σύντομα την απάντησή σας.

Λύση:

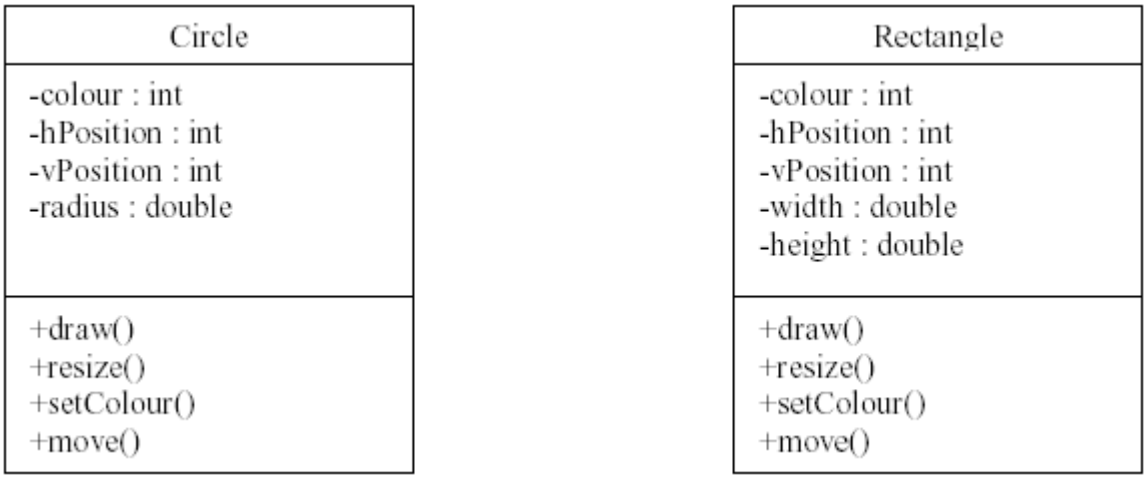
With a linked-list implementation, each method called by `reverse()` takes $O(n)$ time because order-n links have to be followed to identify an element of a given rank. Thus, the total running time of `reverse()` is again $O(n^2)$.

Άσκηση 5 (30 μονάδες)
Απαντήστε στις παρακάτω ερωτήσεις χρησιμοποιώντας την ακόλουθη ιεραρχία κλάσεων, όπου η κλάση στην κορυφή της ιεραρχίας έχει τις αφαιρετικές (abstract) μεθόδους `draw()` και `resize()`.



(α) **(10 μονάδες)** Πώς μπορείτε να υλοποιήσετε την παραπάνω ιεραρχία κλάσεων σε μια γλώσσα προγραμματισμού η οποία δεν υποστηρίζει κληρονομικότητα κλάσεων; Ξανασχεδιάστε το παραπάνω διάγραμμα για την λύση που προτείνετε και εξηγήστε τις διαφορές μεταξύ του αρχικού και του τροποποιημένου διαγράμματος καθώς και τους κυριότερους λόγους για τους οποίους η χρήση της κληρονομικότητας καθιστά ευκολότερη την σχεδίαση και συντήρηση της παραπάνω ιεραρχίας κλάσεων.

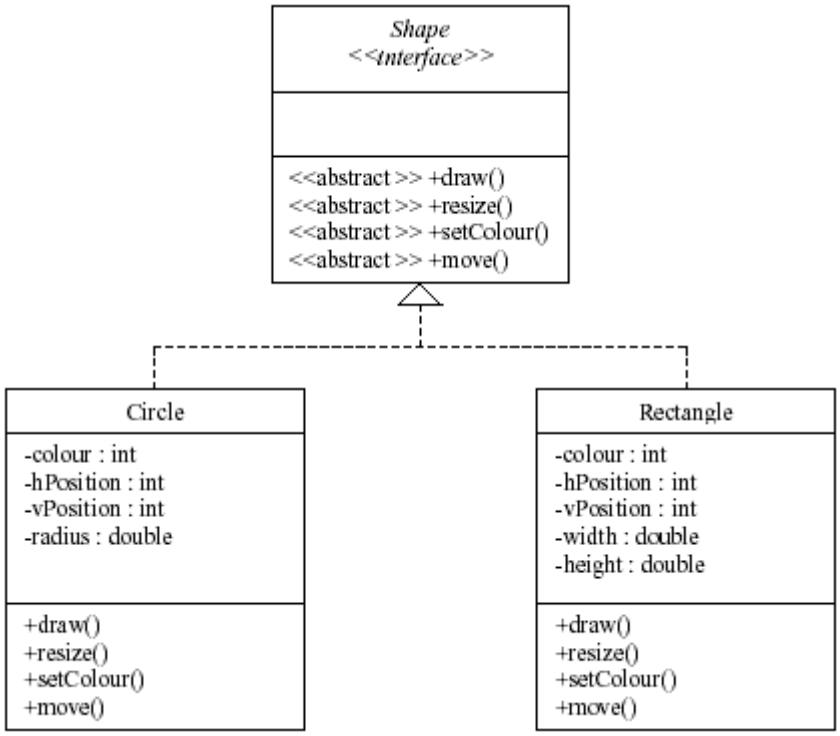
Λύση:
Solution 1



There are two valid solutions to this question: 1) have two classes, `Circle` and `Rectangle`, that implement everything from the initial system, plus all functionality from the `Shape` abstract class; 2) change the `Shape` abstract class to an interface, move all attributes from `Shape` to the subclasses, make methods `setColour()` and move() abstract, and implement `setColour()` and `move()` in both children classes. The downside of either solution, as compared with the initial solution using inheritance, is that code common to both `Circle` and `Rectangle` will have to be duplicated. This is not so much a problem at development-time as it is during maintenance, where changing one of the methods should require changing it in both `Circle` and `Rectangle` classes.

Corollaries to this are that there is an increased amount of code to read and understand, and the chance of making a mistake/typo increases with the size of the code base. A further downside of the first solution is that dynamic binding will not be possible. In order to get a perfect mark on this question, I was also looking for some sense (particularly relevant if you chose answer 1) that taking away the parent class means that the two classes that actually share a lot in common are no longer related. Our mental model of the system is much simpler if we think of having two types of shapes rather than two seemingly distinct classes.

Solution 2



Common Mistakes:
I saw several students choose the option where `Circle`/`Rectangle` were distinct classes, then for some reason kept the `Shape` class –which no longer serves any purpose? Also, several people denoted the methods in the `Circle`/`Rectangle` classes as being protected, which really doesn't make sense if inheritance is not permitted.

(β) **(20 μονάδες)** Υλοποιήστε σε ένα πρόγραμμα Java την σχεδίαση που σας δόθηκε στην αρχή της άσκησης. Στην απάντησή σας να συμπεριλάβετε μόνο τις μεταβλητές στιγμιοτύπων και τις μεθόδους που αναγράφονται στο διάγραμμα. Σημειώστε ότι τα σύμβολα πριν από τις μεταβλητές στιγμιοτύπων και τις μεθόδους της κάθε κλάσης υποδεικνύουν τα αντίστοιχα δικαιώματα εξουσιοδοτημένης πρόσβασης: συν (+) για **`public`**, πλην (-) για **`private`** και δίεση (#) για **`protected`**.

Λύση:

```
public abstract class Shape {
 protected int colour, hPosition, vPosition;
 public abstract void draw();
 public abstract void resize(Resize res);
 public void setColour(int col) {
   colour = col;
   return ;
 }
 public void move(int hMove, int vMove) {
   hPosition += hMove;
   vPosition += vMove;
   return;
 }
}
public class Circle extends Shape {
```

```
 private double radius;
 public void draw() { … }
 public void resize(Resize res) {
   if (res instanceOf ResizeCircle) {
     radius = (ResizeCircle) res.newRadius();
   }
   return;
 }
}
public class Rectangle extends Shape {
 private double width, height;
 public void draw() { … }
 public void resize(Resize res) {
    if (res instanceOf ResizeRectangle) {
      width = (ResizeRectangle) res.newWidth();
      height = (ResizeRectangle) res.newHeight();
    }
    return;
 }
}
public interface Resize() {
}
public class ResizeRectangle implements Resize {
    double width, height;

    public ResizeRectangle(double newHeight, double newWidth){
       width = newWidth;
       height = new Height;
    }
    public double newWidth() {
      return width;
    }
    public double newHeight() {
      return height;
    }
}
public class ResizeCircle implements Resize {
    double radius;
    public ResizeRectangle(double newRadius){
       radius = newRadius;
    }
    public double newRadius() {
      return radius;
    }
}
```

For this question, the main things I was looking for were:

· You recognized that Shape is an abstract class, and correctly defined it as such

· You correctly identified the +/-/# parameters as representing public/private/protected

· Correct definition of the `draw()` and `resize()` abstract methods in the `Shape`, and their concrete implementation in the `Shape` and `Rectangle` classes

· Correctly identified that the `setColour()` and `move()` methods are implemented only in the `Shape` class

· Use of the 'extends Shape' clause when defining the `Circle` and `Rectangle` classes

· Constructor definitions were not required for this question, and were not marked.

Some common mistakes:

· Missing the '`abstract`' clause in the `Shape` class definition.

· Including `super.resize()` or `super.draw()` in the concrete implementation of these methods in the children classes. This doesn't make sense, since there is no method implementation in the parent.

· Adding a `super()` method to the child class. Remember that `super()` is actually a Java defined method, that you should not (and probably cannot) override.

Άσκηση 6 (10 μονάδες)

Θεωρήστε τους παρακάτω ορισμούς κλάσεων και δηλώσεων μεταβλητών Java:

```
class A { int x; String y; }
class B extends A { char z; }
class C { int x; String y; }
A a,a2; B b; C c; Object d;
```

Για κάθε μία από τις ακόλουθες αναθέσεις υποδείξτε αν εμπλέκουν ευρείς (widening) ή στενές (narrowing) μετατροπές ή δεν εμπλέκουν μετατροπές. Επίσης υποδείξτε σε ποιες αναθέσεις θα εμφανιστούν λάθη κατά την μετάφραση του προγράμματος (compiler error).

```
(a) a = a2;
(b) a = b;
(c) a = c;
(d) a = d;
(e) b = (B)a;
(f) b = (B)c;
(g) b = (B)d;
(h) c = (C)d;
```

Λύση:

kind of conversion	compiler error
(a) neither (or, widening)	no
(b) widening	no
(c) neither	yes
(d) narrowing	yes
(e) narrowing	no
(f) neither	yes
(g) narrowing	no
(h) narrowing	no