

Σχεδιασμός, Εμφάνιση και Ύφος

Διομήδης Σπινέλλης¹, Παναγιώτης Λουρίδας²

¹Τμήμα Διοικητικής Επιστήμης και Τεχνολογίας
Οικονομικό Πανεπιστήμιο Αθηνών

²Εθνικό Δίκτυο Έρευνας και Τεχνολογίας Α.Ε.

19 Δεκεμβρίου 2006

Περιεχόμενα

1 Αρχές και Αντιπαραδείγματα

Περιεχόμενα

- 1 Αρχές και Αντιπαραδείγματα
- 2 Οδηγίες και Παραδείγματα Μορφοποίησης

Δομημένος Προγραμματισμός

- Ο δομημένος προγραμματισμός (structured programming) βασίζεται στην οργάνωση του προγράμματος σε μικρές, αυτόνομα κατανοητές, οντότητες.
- Για να επιτευχθεί αυτό κάθε οντότητα έχει ένα σημείο εισόδου και ένα σημείο εξόδου.
- Η δομή του προγράμματος βασίζεται σε τρεις μόνο λογικές δομές:
 - 1 ακολουθία (*sequence*)
 - 2 επιλογή (*condition*)
 - 3 επανάληψη (*repetition*)
- Με τη χρήση των παραπάνω δομών μπορεί να εκφραστεί οποιοσδήποτε αλγόριθμος.
- Σε περιπτώσεις που η σχολαστική τήρηση των παραπάνω κάνει το πρόγραμμα δυσανάγνωστο πρέπει οι εξαιρέσεις να κωδικοποιούνται δομημένα, τεκμηριωμένα, με οικονομία στη χρήση τους, και με τάξη.

Αντιπαράδειγμα Δομημένου Προγραμματισμού

Το παρακάτω κομμάτι κώδικα προκάλεσε την εννιάωρη διακοπή του τηλεφωνικού δικτύου της Νέας Υόρκης στις 15 Ιανουαρίου 1990.

```

1  do {
2      // ...
3      switch
4      // ...
5      if (something) {
6          // ...
7          break;
8          // ...
9      }
10     // ...
11 } while (somethingelse);
    
```

Το σωστό θα ήταν το παρακάτω, που και πάλι δεν είναι δομημένο :

Αντιπαράδειγμα Δομημένου Προγραμματισμού

Το παρακάτω κομμάτι κώδικα προκάλεσε την εννιάωρη διακοπή του τηλεφωνικού δικτύου της Νέας Υόρκης στις 15 Ιανουαρίου 1990.

```

1  do {
2      // ...
3      switch
4      // ...
5      if (something) {
6          // ...
7          break;
8          // ...
9      }
10     // ...
11 } while (somethingelse);
    
```

Το σωστό θα ήταν το παρακάτω, που και πάλι δεν είναι δομημένο :

```

1  do {
2      // ...
3      switch
4      // ...
5      CALL_CENTER_DOWN:
6      if (something) {
7          // ...
8          break CALL_CENTER_DOWN;
9          // ...
10     }
11     // ...
12 } while (somethingelse);
    
```

Ο Νόμος του Demeter

- Στον αντικειμενοστραφή προγραμματισμό ο νόμος του Demeter ορίζει το συνιστώμενο τρόπο συνεργασίας των αντικειμένων.
- Ο νόμος αυτός προάγει την περιορισμένη σύζευξη μεταξύ κλάσεων.
- Έτσι σε κλάσεις μια μέθοδος επιτρέπεται μόνο να καλεί μεθόδους:
 - της δικής της κλάσης
 - αντικειμένων που έλαβε ως παραμέτρους
 - αντικειμένων που δημιούργησε
 - αντικειμένων που περιέχει

Παράδειγμα Νόμου Demeter

```

1  class Demeter {
2      private A a;
3      private int myFunc() { /* ... */ }
4      public void example (B b) {
5          C c = new C();
6          int f = myFunc(); // itself
7
8          // ...
9          b.paramMethod(); // parameter passed
10         a = new A();
11         a.createdMethod(); // created object
12         c.ownMethod(); // contained object
13     }
14 }
```


Αντιπαράδειγμα Νόμου Demeter

```

1  class Demeter {
2      private A a;
3      private int myFunc() { /* ... */ }
4      public void example (B b) {
5          C c = new C();
6          int f = myFunc(); // itself
7
8          // ...
9          b.paramMethod().otherMethod(); // WRONG!!!
10         a = new A();
11         a.createdMethod(); // created object
12         c.ownMethod(); // contained object
13     }
14 }
```

Φρικώδικας 1

```
1  if ( ( country == SING) || (country == BRNI) ||  
2      (country == POL) || (country == ITALY) )  
3  {  
4      /*  
5          * If the country is Singapore , Brunei or Poland  
6          * then the current time is the answer time  
7          * rather than the off hook time .  
8          * Reset answer time and set day of week.  
9          */
```

Φρικώδικας 2

```
1  if (month == FEB) {  
2      if (year % 4 == 0)  
3          if (day > 29)  
4              legal = FALSE;  
5      else  
6          if (day > 28)  
7              legal = FALSE;  
8  }
```

Φρικώδικας 2—Διόρθωση

```
1  if (month == FEB) {  
2      if (year % 4 == 0) {  
3          if (day > 29)  
4              legal = FALSE;  
5      } else {  
6          if (day > 28)  
7              legal = FALSE;  
8      }  
9  }
```

Φρικώδικας 2—Βελτίωση

Ο βελτιωμένος κώδικας είναι πιο καθαρός και επιπλέον υπολογίζει σωστά τα δίσεκτα έκτη στα πολλαπλάσια του 400.

```
1  if (month == FEB) {  
2      int nday;  
3  
4      nday = 28;  
5      if ((year % 400 == 0) || ((year % 100 != 0) && (year % 4 == 0)))  
6          nday = 29;  
7      if (day > nday)  
8          legal = FALSE;  
9  }
```

Φρικώδικας 3

```
1 // Compute roots of a quadratic equation
2 // This code assumes that (b*b - 4 *a*c) is positive
3 temp = Math.sqrt(b*b - 4*a*c);
4 root(0) = (-b + temp) / (2 * a);
5 root(1) = (-b - temp) / (2 * a);
6
7 ...
8
9 //swap the roots
10 temp = root(0);
11 root(0) = root(1);
12 root(1) = temp;
```

Φρικώδικας 3—Διόρθωση

```

1 // Compute roots of a quadratic equation
2 // This code assumes that (b*b - 4 * a*c) is positive
3 discriminant = Math.sqrt(b*b - 4*a*c);
4 root(0) = (-b + discriminant) / (2 * a);
5 root(1) = (-b - discriminant) / (2 * a);
6
7 ...
8
9 //swap the roots
10 temp = root(0);
11 root(0) = root(1);
12 root(1) = temp;
    
```

Φρικώδικας 4

```
1  /*
2   * default
3   */
4   default :
5       break ;

1  /* return SUCCESS */
2  return SUCCESS;

1  zerocount++; /* Increment zero entry counter */

1  /* Initialise "total" to "number_received" */
2  node.total = node.number_received;
```


Φρικώδικας 5

Ερώτηση: Τι κάνει το παρακάτω πρόγραμμα:

```
1 public class CleverSwap {
2     public static void main(String[] args) {
3         int x = 1984;
4         int y = 2001;
5         y = (x ^ (y ^ x)) ^ y;
6         System.out.println("x=␣" + x + "; y=␣" + y);
7     }
8 }
```

Φρικώδικας 5

Ερώτηση: Τι κάνει το παρακάτω πρόγραμμα:

```
1 public class CleverSwap {
2     public static void main(String[] args) {
3         int x = 1984;
4         int y = 2001;
5         y = (x ^ (y ^ x)) ^ y;
6         System.out.println("x_=_ " + x + " ;_y_=_ " + y);
7     }
8 }
```

Απάντηση: Ό,τι και αυτό

```
1 public class CleverSwap {
2     public static void main(String[] args) {
3         int x = 1984;
4         int y = 2001;
5         int tmp;
6
7         tmp = x;
8         x = y;
9         y = tmp;
10
11         System.out.println("x_=_ " + x + " ;_y_=_ " + y);
12     }
13 }
```

Σχόλιο στην Αρχή του Αρχείου

- Τα σχόλια γράφονται στα αγγλικά
- Τα * στοιχίζονται κάθετα μεταξύ τους
- Όνομα κλάσης
- Το όνομα του αρχείου πρέπει να είναι ίδιο με το όνομα της κλάσης
- Προσδιοριστής CVS/RCS.
- Copyright

```
1  /*
2   * Class name
3   *
4   * $Id$
5   *
6   * Copyright year name/organisation.
7   */
```

Προίμιο της Κλάσης

```
7
8
9 package gr . aueb . dmst . Package ;
10
11 import java . bla . blough . bligh ;
12
13 /**
14  * One-line class summary.
15  * Class summary.
16  *
17  * @version 1.82 18 Mar 1999
18  * @author Name Surname
19  */
```

Δήλωση της Κλάσης

```
20  
21 public class SpecialHtmlParser extends FileHtmlParser {  
22     /*  
23     * Class implementation comment.  
24     */
```

Πεδία Κλάσης

```

24  /** One-line comment of classVar1 */
25  public static int classVar1;
26
27  /** One-line comment of classVar2 */
28  protected static int classVar2;
29
30  /** One-line comment of classVar3 */
31  static int classVar3;
32
33  /**
34     * One-line comment of classVar4
35     * Summary of classVar4
36     */
37  private static Object classVar4;
    
```

Πεδία Υπόστασης

```

38  /** One-line comment of classVar1 */
39  public int classVar1;
40
41  /** One-line comment of classVar2 */
42  protected int classVar2;
43
44  /** One-line comment of classVar3 */
45  int classVar3;
46
47  /**
48   * One-line comment of classVar4
49   * Summary of classVar4
50   */
51  private Object classVar4;
    
```

Μέθοδοι Κατασκευής

```
52 /** Construct a parser for the specified HTML dialect. */  
53 SpecialHhtmlParser(String dialect, int maxNesting) {  
54     // Code  
55 }  
56  
57 /** Construct a default HTML parser. */  
58 SpecialHTMLParser(String dialect) { }
```


Λοιπές Μέθοδοι

```
1  
2 /**  
3  * One line method comment.  
4  * Summary of method.  
5  */  
6 public void doSomething() { }
```

Δηλώσεις Μεταβλητών

```
1 int nLines = 0; // HTML lines in the file
2 int nChars = 0; // Number of characters in the file
3 int i;
4 double averageDensity = 1.0; // Average density of comme
5 Car transport; // Transport to use
```

Στοίχιση Δομής for

```
1  for (;;)
2      a++;
3
4  for (int j = 0; j < 10; j++) {
5      a++;
6      b++;
7  }
```

Στοίχιση Δομής while

```
1  while (a < 10) {  
2      a++;  
3      b += 12;  
4  }  
5  
6  while (a < 10) {  
7      a++;  
8      for (int i = 0; i < 10; i++)  
9          b += 12;  
10 }
```

Στοίχιση Δομής do while

```
1 do {  
2     a++;  
3     b += 12;  
4 } while (a < 10);
```

Στοίχιση Δομής switch

```
1  switch (c) {  
2  case 'a':  
3      System.out.println("Alpha");  
4      break;  
5  case 'b':  
6      System.out.println("Bravo");  
7      break;  
8  default :  
9      System.out.println("???");  
10     break;  
11 }
```

Στοίχιση Δομής if, if else

```

1  if (a == 3) {
2      System.out.println("Threee");
3      k = 52;
4  }
```

```

1  if (a == 3) {
2      System.out.println("Threee");
3      k = 52;
4  } else {
5      System.out.println("Not_three");
6      k = 55;
7  }
```

Στοίχιση Δομής if else, if else. . . if else

```

1  if (a == 3)
2      System.out.println("Three");
3  else
4      System.out.println("Not_three");
    
```

```

1  if (s.equals("a"))
2      system.out.println("Alpha");
3  else if (s.equals("b"))
4      system.out.println("Bravo");
5  else if (s.equals("c"))
6      system.out.println("Charlie");
7  else
8      system.out.println("???");
    
```


Στοίχιση Δομής try/catch/finally

```
1  try {  
2      i = 12;  
3      o.myMethod();  
4  } catch (ExceptionClass e) {  
5      statements;  
6  } finally {  
7      statements;  
8  }
```

Javadoc Μεθόδου

```

1  /**
2   * Returns an Image object that can then be painted on the screen.
3   * The url argument must specify an absolute {@link URL}. The name
4   * argument is a specifier that is relative to the url argument.
5   * <p>
6   * This method always returns immediately, whether or not the
7   * image exists. When this applet attempts to draw the image on
8   * the screen, the data will be loaded. The graphics primitives
9   * that draw the image will incrementally paint on the screen.
10  *
11  * @param url an absolute URL giving the base location of the image
12  *           name the location of the image, relative to the url argument
13  * @return    the image at the specified URL
14  * @see      Image
15  */
16  public Image getImage(URL url , String name) {
17      try {
18          return getImage(new URL(url , name));
19      } catch (MalformedURLException e) {
20          return null;
21      }
22  }
    
```

Ειδικές Λέξεις στα Σχόλια

```
1  /*  
2   * A non-javadoc comment with a special keyword  
3   * (recommended keywords in parenthesis)  
4   *  
5   * (XXX, FIXME, TODO): The comment  
6   */
```