



Ελληνική Δημοκρατία  
Τεχνολογικό Εκπαιδευτικό  
Ίδρυμα Ηπείρου

# Προγραμματισμός Διαδικτύου

Ενότητα 9 : Δυναμικοί τύπου δεδομένων

Ιωάννης Τσούλος



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



## Τμήμα Μηχανικών Πληροφορικής Τ.Ε

### Προγραμματισμός Διαδικτύου

### Ενότητα 9 : Δυναμικοί τύπου δεδομένων

Ιωάννης Τσούλος  
Επίκουρος Καθηγητής  
Άρτα, 2015



# Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



# Χρηματοδότηση

- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «**Εκπαίδευση και Δια Βίου Μάθηση**» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο ΤΕΙ Ηπείρου**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



# Object

- Αποτελεί την βάση για όλες τις κατηγορίες της γλώσσας.
  - Μέθοδος `equals()`.
  - Μέθοδος `getClass()`: Μπορεί να χρησιμοποιηθεί σαν ένας έμμεσος δείκτης προς την κατηγορία που βρίσκεται πίσω από ένα αντικείμενο.
  - Μέθοδος `toString()`: Χρησιμοποιείται προκειμένου να επιστρέψει με την μορφή αλφαριθμητικού πληροφορία για ένα αντικείμενο όπως για παράδειγμα να επιστρέψει σαν αλφαριθμητικό τις συντεταγμένες ενός σημείου κτλ.



# Object

```
import javax.swing.*;
public class Object1
{
    public static int readInt()
    {
        int x;
        String s;
        s=JOptionPane.showInputDialog("Enter _integer _");
        x=Integer.parseInt(s);
        return x;
    }
    public static void main(String arg[])
    {
        Integer val1=new Integer(readInt());
        Integer val2=new Integer(readInt());
        if(val1.equals(val2))
            System.out.println("Same _integers _");
        else
            System.out.println("Not _same _integers _");
    }
}
```



# Object

```
class Person
{
    private String name,lastname;
    public Person(String x,String y)
    {
        name=x;
        lastname=y;
    }
    public String getName()
    {
        return name;
    }
    public String getLastName()
    {
        return lastname;
    }
    public boolean equals(Person other)
    {
        if(name.equals(other.getName()) &&
            lastname.equals(other.getLastName()))
            return true;
        return false;
    }
}
public class Object2
{
    public static void main(String arg [])
    {
        Person p1 =new Person("Giannis","Tsoulos");
        Person p2 =new Person("Kostas","Antoniou");
        if(p1.equals(p2))
            System.out.println("Same_persons");
        else
            System.out.println("Not_same_persons");
    }
}
```



## Reflection στην Java.

# Object

```
import java.lang.reflect.*;
class Person
{
    private String name,lastname;
    public Person(String x,String y)
    {
        name=x;
        lastname=y;
    }

    public String getName()
    {
        return name;
    }

    public String getLastName()
    {
        return lastname;
    }

    public boolean equals(Person other)
    {
        if(name.equals(other.getName()) &&
            lastname.equals(other.getLastName()))
            return true;
        return false;
    }
}

public class Object3
{
    public static void main(String arg[])
    {
        int i;
        Person p1 =new Person("Giannis","Tsoulos");
        String pname=p1.getClass().getName();
        System.out.println("Class_name_"+pname);
        try{
            Class c=Class.forName(pname);
            Method m[]=c.getDeclaredMethods();
            for(i=0;i<m.length;i++)
            {
                System.out.println("Method_"+m[i]);
            }
        }catch(Exception e)
        {
            System.out.println("Wrong_class_name_");
        };
    }
}
```





# ArrayList

- Χρησιμοποιείται προκειμένου να αποθηκεύσουμε σε αυτήν δεδομένα διαφόρων ειδών τα οποία δεν είναι γνωστά εκ των προτέρων για το πλήθος τους
  - `contains(Object x)`: Αυτή η μέθοδος επιστρέφει αληθές αν η λίστα περιέχει το αντικείμενο `x` και ψευδές σε διαφορετική περίπτωση.
  - `remove(Object x)`: Αυτή η μέθοδος προσπαθεί να διαγράψει το αντικείμενο `x` από την λίστα Σε περίπτωση επιτυχίας επιστρέφει αληθές διαφορετικά ψευδές.
  - `isEmpty()`: Η μέθοδος αυτή επιστρέφει αληθές αν δεν υπάρχει κανένα
  - `get(position)`: Η μέθοδος επιστρέφει το αντικείμενο `Object` που είναι στην θέση `position`. Οι θέσεις ξεκινούν από το 0 και σταματούν στο `size()-1`, όπως ακριβώς και στους στατικούς πίνακες.
  - `clear()`: Η μέθοδος αυτή διαγράφει όλα τα στοιχεία της λίστας.
  - `indexOf(Object x)`: Η μέθοδος αυτή επιστρέφει την θέση του αντικειμένου `x` μέσα στην λίστα.



# ArrayList

```
import java.util.*;
public class ArrayList2
{
    public static void main(String arg[])
    {
        int i;
        Integer int1=new Integer(10);
        Integer int2=new Integer(20);
        Float fl1=new Float(150.25);
        Float fl2=new Float(-89.22);
        ArrayList list=new ArrayList();
        list.add(int1);
        list.add(fl1);
        list.add(int2);
        list.add(fl2);
        for(i=0;i<list.size();i++)
        {
            Object x=list.get(i);
            if(x.getClass().getName().
                equals("java.lang.Integer"))
            {
                Integer xx=(Integer)x;
                System.out.println("Integer_"+xx);
            }
            else
            {
                Float yy=(Float)x;
                System.out.println("Float_"+yy);
            }
        }
    }
}
```



# Σύνολα

## Ένωση συνόλων

- Τα σύνολα είναι μια μικρή επέκταση των λιστών που είδαμε προηγουμένως με την προσθήκη πως δεν επιτρέπονται διπλές εγγραφές σε αυτά.
- Δηλαδή αν για παράδειγμα ένας ακέραιος έχει ήδη προστεθεί με κάποια μέθοδο `add()` αυτό δεν μπορεί να ξαναγίνει.

```
import javax.swing.*;
import java.util.*;

public class Set1
{
    public static int readInt()
    {
        String s=JOptionPane.showInputDialog("Enter integer");
        int x=Integer.parseInt(s);
        return x;
    }

    public static void printSet(HashSet set)
    {
        System.out.print("Set={");
        System.out.print(set);
        System.out.println("}\n");
    }

    public static void main(String arg[])
    {
        HashSet set1,set2,set3;
        int i;
        final int maxNumbers=10;
        Random rand=new Random();
        set1=new HashSet();
        set2=new HashSet();
        set3=new HashSet();

        for(i=0;i<maxNumbers;i++)
        {
            Integer x=new Integer(readInt());
            if(rand.nextInt()%2==0) set1.add(x);
            else
                set2.add(x);
            set3.add(x);
        }
        System.out.println("Set1");
        printSet(set1);
        System.out.println("Set2");
        printSet(set2);
        System.out.println("Set3");
        printSet(set3);
    }
}
```



# Στοιίβες

- Σε αυτήν την δομή θεωρούμε πως υπάρχει ένα σημείο στο οποίο μπορούμε να κάνουμε εισαγωγές και εξαγωγές.
- Αυτό το σημείο θεωρείται η κορυφή της λίστας και όλες οι πράξεις γίνονται μόνον εκεί.

```
import javax.swing.*;
import java.util.*;

public class Stack1
{
    public static void main(String arg[])
    {
        String s=JOptionPane.showInputDialog("Enter_a_phrase_");
        Stack mystack=new Stack();
        int i;
        for(i=0;i<s.length();i++)
        {
            char ch=s.charAt(i);
            if(ch=='(') mystack.push(new Integer(10));
            else
                if(ch==')')
                {
                    if(mystack.isEmpty())
                    {
                        System.out.println("Stack_is_empty_");
                        System.out.println("The_phrase_is_not_ok");
                    }
                    else
                        mystack.pop();
                }
        }
        if(mystack.isEmpty())
            System.out.println("The_phrase_is_ok");
        else
            System.out.println("The_phrase_is_not_ok");
    }
}
```

# Επαναλήπτες

- Οι επαναλήπτες είναι μια απλή στην χρήση δομή για την ανάκτηση των στοιχείων της λίστας και δεν ενδιαφέρονται για το μέγεθος της λίστας και είναι απλούστεροι στην χρήση

```
import java.util.*;

public class Iterator1
{
    public static void main(String arg[])
    {
        final int N=10;
        int i;
        ArrayList list=new ArrayList();
        Iterator iter;
        Random rnd=new Random();
        for (i=0;i<N;i++)
            list.add(rnd.nextInt());
        iter=list.iterator();
        while(iter.hasNext())
        {
            Integer x=(Integer)iter.next();
            System.out.println("x="+x);
        }
    }
}
```



# Επαναλήπτες

## Χρήση του ListIterator

```
import java.util.*;
public class Iterator2
{
    public static void main(String arg [])
    {
        ArrayList list=new ArrayList();
        final int N=10;
        int i;
        ListIterator iter;
        int count=0;
        Random rnd=new Random();
        for (i=0;i<N;i++)
            list.add(rnd.nextInt() %100);
        iter=list.listIterator();

        while(iter.hasNext())
        {
            Integer x=(Integer)iter.next();
            System.out.println("Count_"+count);
            System.out.println("Element_"+x);
            if(rnd.nextInt() % 2 ==0)
                iter.set(new Integer(0));
            count++;
        }
        while(iter.hasPrevious())
        {
            Integer x=(Integer)iter.previous();
            System.out.println("Count_"+count);
            System.out.println("Element_"+x);
            count--;
        }
    }
}
```



# Βιβλιογραφία

- Πλήρες Εγχειρίδιο της Java 2 Platform, Laura Lemay & Rogers Cadenhead, Ελληνική έκδοση Γκιούρδα
- Java Προγραμματισμός, 6η έκδοση, Harvey M. Deitel, Paul J. Deitel . Εκδόσεις Μ. Γκιούρδας, Αθήνα 2005.
- Εισαγωγή στη γλώσσα προγραμματισμού Java, Εμμ. Σκορδαλάκης, Εργαστήριο Λογισμικού Ομάδα Τεχνολογίας Λογισμικού ΕΜΠ
- Java με UML, Else Lervik και Vegard B. Havdal, Εκδόσεις Κλειδάριθμος, 2005.
- Προγραμματισμός με Java, Γιάννη Κάβουρα, Εκδόσεις Κλειθάρθμος, Αθήνα 2003.
- Java in a nutshell, Deluxe Edition και Java Examples in a Nutshell, David Flanagan, Εκδόσεις OReilly
- Java Collections An Introduction to ADTs, Data Structures and Algorithms, David Watt, Deryck Brown, John Wiley and Sons
- Thinking in Java (3rd edition), B. Eckel, Pearson Prentice Hall, 2003.
- Introduction to Java Programming (5th edition), Y. Daniel Liang, Pearson Prentice Hall, 2005.



# Σημείωμα Αναφοράς

Copyright Τεχνολογικό Ίδρυμα Ηπείρου. Ιωάννης Τσούλος.  
Προγραμματισμός Διαδικτύου.

Έκδοση: 1.0 Άρτα, 2015. Διαθέσιμο από τη δικτυακή  
διεύθυνση:

<http://eclass.teiep.gr/OpenClass/courses/COMP114/>





# Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά Δημιουργού-Μη Εμπορική Χρήση-Όχι Παράγωγα Έργα 4.0 Διεθνές [1] ή μεταγενέστερη. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, Διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

[1] <http://creativecommons.org/licenses/by-nc-nd/4.0/deed.el>



# Τέλος Ενότητας

Επεξεργασία: Ευάγγελος Καρβούνης  
Άρτα, 2015



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



# Τέλος Ενότητας

Δυναμικοί τύπου δεδομένων



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

