



Εργαστήριο 02: Προγραμματισμός με Γενικούς Τύπους (JAVA Generics)

Διδάσκων: Παναγιώτης Ανδρέου

Generic Types

- Τα Generics έχουν προστεθεί στη JAVA από το 2004 ως μέρος του J2SE 5.0
- Με τη χρήση των generics μπορούμε να δημιουργήσουμε κλάσεις και μεθόδους που να διαχειρίζονται διαφόρων ειδών αντικείμενα (π.χ., Στοίβα με ακέραιους, Στοίβα με συμβολοσειρές) με την ίδια υλοποίηση.
- Ο προγραμματισμός με generics μπορεί να γίνει αρκετά περίπλοκος
- Εμείς, θα επικεντρωθούμε στα ιδιώματα και την σύνταξη που παρουσιάζονται στο βιβλίο.
- Δεν θα μπούμε σε βάθος που να καλύπτει όλο το εύρος της χρήσης τους

Μία γενική κλάση MemoryCell (πριν την JAVA 5.0)

```
// MemoryCell class
// Object read( ) --> Returns the stored value
// void write( Object x ) --> x is stored
public class MemoryCell {
    // Private internal data representation
    private Object storedValue;

    // Public methods
    public Object read() {
        return storedValue;
    }

    public void write(Object x) {
        storedValue = x;
    }
}
```

Μία γενική κλάση MemoryCell (με Generics)

```
// MemoryCell class
// AnyType read( ) --> Returns the stored value
// void write( AnyType x ) --> x is stored
public class MemoryCell<AnyType> {
    // Private internal data representation
    private AnyType storedValue;

    // Public methods
    public AnyType read() {
        return storedValue;
    }

    public void write(AnyType x) {
        storedValue = x;
    }
}
```

Γιατί να χρησιμοποιηθούν τα generics;

Ο κώδικας που χρησιμοποιεί generics έχει πολλά πλεονεκτήματα:

- **Καλύτερος έλεγχος σε επίπεδο μεταγλώττισης**
Ο μεταγλωττιστής της JAVA εφαρμόζει ισχυρό έλεγχο για τύπους. Έτσι τα λάθη διορθώνονται σε επίπεδο μεταγλώττισης και είναι πιο δύσκολο να δημιουργηθούν runtime exceptions
- **Απάλειψη των μετατροπών τύπων (casting)**
 - Παράδειγμα χωρίς generics: `List list = new ArrayList(); list.add("hello"); String s = (String) list.get(0);`
 - Παράδειγμα με generics: `List<String> list = new ArrayList<String>(); list.add("hello"); String s = list.get(0); // no cast`
- **Επιτρέπει την δημιουργία γενικών αλγορίθμων**
Οι προγραμματιστές μπορούν να δημιουργούν γενικούς αλγόριθμους που δουλεύουν σε συλλογές διαφορετικών τύπων που είναι ασφαλής όσο αφορά τους τύπους (type safe) και είναι πιο κατανοητές

Κοινοί τύπου Generics

- E - Element (used extensively by the Java Collections Framework)
- K - Key
- N - Number
- T - Type
- V - Value
- S,U,V etc. - 2nd, 3rd, 4th types

- Αυτοί οι τύποι χρησιμοποιούνται σε διάφορες βιβλιοθήκες του Java API

Δημιουργία και Αρχικοποίηση ενός τύπου Generic

- Για δημιουργία και αναφορά ενός τύπου Generic στον κώδικα, θα πρέπει να αντικαταστήσετε τον τύπο Generic (π.χ., E, T) με ένα “γνωστό/κανονικό” τύπο (π.χ., Integer).

Παράδειγμα:

```
MemoryCell<Integer> intMemoryCell;
```

- Προσοχή: μπορούν να χρησιμοποιηθούν μόνο αναφορές σε αντικείμενα (δηλ., Integer και όχι ο αρχέγονος τύπος int)
- Παρόμοια με άλλες δηλώσεις μεταβλητών, ο πιο πάνω κώδικας έχει δημιουργήσει μία αναφορά σε ένα αντικείμενο τύπου MemoryCell object. Για να αρχικοποιηθεί το αντικείμενο θα πρέπει να χρησιμοποιηθεί το keyword **new**, αλλά επίσης να δηλωθεί ο τύπος που θα χρησιμοποιηθεί (δηλ., Integer) πριν τις παρενθέσεις ():

```
MemoryCell<Integer> intMemoryCell = new  
MemoryCell<Integer>();
```

Δημιουργία και Αρχικοποίηση ενός τύπου Generic

- Από την Java SE 7, κατά τη διαδικασία αρχικοποίησης, μπορούν να αντικατασταθούν οι παράμετροι τύπου με το άδειο σύνολο (<>) δεδομένου ότι ο μεταγλωττιστής μπορεί να ερμηνεύσει τον τύπο από την δήλωση

Παράδειγμα:

```
MemoryCell<Integer> intMemoryCell = new MemoryCell<>();
```

- Οι παρενθέσεις <>, ονομάζονται και *the diamond*.

Generic Interfaces

- Η διαπροσωπεία Comparable, Java v.x<5

```
package java.lang;  
public interface Comparable {  
    public int compareTo( Object o );  
}
```

- Η διαπροσωπεία Comparable, Java v.x>=5

```
package java.lang;  
public interface Comparable<T> {  
    public int compareTo( T o );  
}
```

Τύποι Μπαλαντέρ (Wildcards <?>)

- Υποθέστε μία κλάση **Shape** και μία μέθοδο `double area()`
- Υποθέστε επίσης ότι υπάρχει μία στατική μέθοδος `totalArea()` που δέχεται μία συλλογή (Collection) από Shapes (δηλ., **Collection<Shape>**) και βρίσκει το συνολικό εμβαδό όλων των σχημάτων της συλλογής

```
public static double totalArea (Collection<Shape> arr) {  
    double total = 0.0;  
    for ( Shape s : arr )  
        if ( s != null ) total += s.area();  
    return total;  
}
```

- Τώρα, υποθέστε δύο κλάσεις **Circle** και **Square** που κληρονομούν την κλάση **Shape**.
- Ερώτηση: Τι θα συμβεί αν περάσουμε σαν παράμετρο μία συλλογή τύπου **Collection<Square>**;

Χρήση Wildcards <?>

- Απάντηση: Λάθος μεταγλώττισης.
- Οι Generic συλλογές δεν είναι γενικού τύπου (covariant).
- Συνεπώς, δεν μπορούμε να περάσουμε σαν παράμετρο αναφορά σε αντικείμενο τύπου Collection<Square> στη μέθοδο totalArea()
- Τα Wildcards <?> χρησιμοποιούνται για να αναφερθούμε σε υποκλάσεις (ή υπερκλάσεις) στις παραμέτρους.
- Παράδειγμα μετασχηματισμού της μεθόδου totalArea

```
public static double totalArea (Collection<? extends Shape> arr) {  
    double total = 0.0;  
    for ( Shape s : arr )  
        if ( s != null ) total += s.area();  
    return total;  
}
```
- Χωρίς τη χρήση WildCard: Collection< T extends Shape> arr>

Έλεγχος Ορίων Τύπων (Type Bounds)

- Υποθέστε την πιο κάτω μέθοδο

```
public static <AnyType> AnyType findMax( AnyType [ ] arr ) {  
    int maxIndex = 0;  
    for( int i = 1; i < arr.length; i++ )  
        if( arr[ i ].compareTo( arr[ maxIndex ] ) > 0 )  
            maxIndex = i;  
    return arr[ maxIndex ];  
}
```

- Η πιο κάτω γραμμή

```
arr[ i ].compareTo( arr[ maxIndex ] ) > 0
```

προϋποθέτει την υλοποίηση της διαπροσωπείας Comparable

- **Ερώτηση:** Έχουμε ορίσει ότι ο τύπος AnyType υλοποιεί την διαπροσωπεία Comparable; → ΌΧΙ (σφάλμα μεταγλώττισης)

Χρήση Type Bounds

- Πρέπει με κάποιο τρόπο να αποδεχόμαστε μόνο αντικείμενα που υποστηρίζουν την μέθοδο `compareTo` (δηλ., να υλοποιούν την διαπρωπεία `Comparable`)

- Υποθέστε την πιο κάτω εναλλακτική υλοποίηση:

```
public static <AnyType extends Comparable<AnyType>>
    AnyType findMax( AnyType[] arr ) {
    int maxIndex = 0;
    for( int i = 1; i < arr.length; i++ )
        if( arr[i].compareTo( arr[ maxIndex ] ) > 0 )
            maxIndex = i;
    return arr[ maxIndex ];
}
```

- Ο πιο πάνω κώδικας μεταγλωττίζεται
- Η γραμμή κώδικα `<AnyType extends Comparable<AnyType>>` είναι ένα type bound

Χρήση Type Bounds (συν.)

- ...συνέχεια από την προηγούμενη διαφάνεια
- Υποθέστε ότι η κλάση Shape υλοποιεί την διαπρωπεία Comparable<Shape>
- Υποθέστε επίσης ότι η κλάση Circle κληρονομεί από την Shape
- Συμπέρασμα:
 - Η κλάση Square ΕΙΝΑΙ ένα Comparable<Shape>
 - Η κλάση Square ΔΕΝ ΕΙΝΑΙ ένα Comparable<Square>
- Συνεπώς πρέπει με κάποιο τρόπο να πούμε ότι AnyType ΕΙΝΑΙ ένα Comparable<T> όπου Tα είναι μία υπερκλάση του AnyType
- Για να το καταφέρουμε, μπορούμε να χρησιμοποιήσουμε ένα wildcard

```

public static <AnyType extends
    Comparable<? super AnyType>>
    AnyType findMax( AnyType [ ] arr ) {
    int maxIndex = 0;
    for( int i = 1; i < arr.length; i++ )
        if( arr[ i ].compareTo( arr[ maxIndex ] ) > 0 )
            maxIndex = i;
    return arr[ maxIndex ];
}

```

- Ο μεταγλωττιστής θα δέχεται πίνακες από αντικείμενα τύπου AnyType, έτσι ώστε το T υλοποιεί την διαπρωπεία Comparable<S>, όπου το T ΕΙΝΑΙ S.
- Ευτυχώς, στο μάθημα δεν θα δούμε τίποτα πιο περίπλοκο από το πιο πάνω!

References

- Java Generic types
<http://docs.oracle.com/javase/tutorial/java/generics/index.html>
- Mark A. Weiss - Data Structures and Algorithm Analysis in Java (3rd Edition) - Publication Date: November 28, 2011
ISBN-10: 0132576279
- ISBN-13: 978-0132576277
- http://en.wikipedia.org/wiki/Generics_in_Java
- <http://docs.oracle>

Άσκηση 1

- Υλοποιήστε μία κλάση `GenericMemoryCell` ως εξής:
 - Να έχει μία `private` μεταβλητή `storedValue` τυχαίου τύπου
 - Ένα άδειο κατασκευαστή
 - Ένα κατασκευαστή που να δέχεται ένα αντικείμενο τυχαίου τύπου
 - Μία μέθοδο `read()` που επιστρέφει την μεταβλητή `storedValue`
 - Μία μέθοδο `write()` που να ενημερώνει την μεταβλητή `storedValue`
- Δημιουργείστε ένα πρόγραμμα που να ελέγχει την πιο πάνω κλάση

Άσκηση 2

- Μελετήστε, Μεταγλωττίστε και Τρέξετε την κλάση TypeB.java