

ΑΛΛΑΓΕΣ ΜΕΤΑΞΥ ΠΡΩΤΟΓΕΝΩΝ ΤΥΠΩΝ (1)

- Η Java επιτρέπει αλλαγές κυρίως μεταξύ αριθμητικών τύπων.
- Οι τιμές `true` και `false` του τύπου `Boolean` δεν μπορούν να χρησιμοποιηθούν για αλλαγή τύπου.
- Αυτόματη αλλαγή τύπου γίνεται από
 - ✓ `byte` ή `char` → `int`
 - ✓ `int` → `long` ή `float`
 - ✓ οποιοσδήποτε τύπος → `double`
- Όταν πάμε από «μικρότερο» τύπο σε «μεγαλύτερο» δεν έχουμε απώλειες ακρίβειας. Στο αντίστροφο μπορεί να έχουμε.

ΑΛΛΑΓΕΣ ΜΕΤΑΞΥ ΠΡΩΤΟΓΕΝΩΝ ΤΥΠΩΝ (2)

- Σε άλλες περιπτώσεις απαιτείται ρητή δήλωση αλλαγής:

(<όνομα-τύπου><τιμή>

Π.χ. `int result = (int) (x/y)`

και **ΟΧΙ** `int result = (int) x/y`

ΑΛΛΑΓΗ ΤΥΠΟΥ ΑΝΤΙΚΕΙΜΕΝΩΝ (1)

- Τα αντικείμενα (στιγμιότυπα) μπορούν να αλλάζουν τύπο με τον περιορισμό: οι κλάσεις προέλευσης και προορισμού να συνδέονται με τη σχέση «υποκλάση».

```
public class Parent  
{ . . . }
```

```
public class Child extends Parent  
{ . . . }
```

```
Parent p1 = new Child();
```

```
c1 = (Child) p1;
```

ΑΛΛΑΓΗ ΤΥΠΟΥ ΑΝΤΙΚΕΙΜΕΝΩΝ (1)

- Δεν απαιτείται ρητή δήλωση αλλαγής τύπου όταν η κλάση του αντικειμένου (κλάση προέλευσης) είναι υποκλάση της κλάσης προορισμού (π.χ. του τύπου της μεταβλητής).

```
Child c2 = new Child();
```

```
Parent p2 = (Parent) c2;
```

```
Parent p2 = c2;
```

ΑΛΛΑΓΗ ΤΥΠΟΥ ΑΝΤΙΚΕΙΜΕΝΩΝ (2)

Π.χ. έστω η υποκλάση `GraphicCircle` της κλάσης `Circle` και η μέθοδος:

```
Circle biggerCircle (Circle c) {...}
```

και `c1` στιγμιότυπο της `Circle` και `gc1` στιγμιότυπο της `GraphicCircle`.

- Τότε μπορώ να γράψω την κλήση:

```
c1.biggerCircle (gc1)
```

Η αλλαγή τύπου του αντικειμένου γίνεται αυτόματα. Το ίδιο ισχύει και στην περίπτωση μεταβλητών.

ΑΛΛΑΓΗ ΤΥΠΟΥ ΑΝΤΙΚΕΙΜΕΝΩΝ (3)

• Σε αντίθετη περίπτωση απαιτείται ρητή δήλωση:

(<όνομα-κλάσης>) <αντικείμενο>

Π.χ.

```
Circle c = new Circle();
```

```
GraphicCircle gc = new GraphicCircle();
```

```
c = gc; //Δεν απαιτείται ρητή δήλωση  
        (πιθανή απώλεια πληροφορίας)
```

```
gc = (GraphicCircle)c;
```

```
//Απαιτείται ρητή δήλωση  
(πιθανό κέρδος πληροφορίας)
```

ΜΕΤΑΤΡΟΠΗ ΠΡΩΤΟΓΕΝΩΝ ΤΥΠΩΝ ΣΕ ΑΝΤΙΚΕΙΜΕΝΑ ΚΑΙ ΑΝΤΙΣΤΡΟΦΑ

- Τέτοιες μετατροπές δεν είναι δυνατές.
- Εναλλακτική λύση: δημιουργία στιγμιοτύπου αντίστοιχης κλάσης.
- Κλάσεις: `Boolean`, `Byte`, `Float`, `Double`, `Character`, `Integer`, `Long`, `Short`
- Π.χ.

```
Integer count = new Integer(51);
```

Για να χρησιμοποιηθεί ως πρωτογενής τιμή:

```
int newCount = count.intValue();
```

- Μια συνήθης μετατροπή:

```
String age = "150";
```

```
int newage = Integer.parseInt(age);
```

ΕΥΡΕΣΗ ΚΛΑΣΗΣ ΑΝΤΙΚΕΙΜΕΝΟΥ

- Με τη μέθοδο `getClass()` της `Object`
`x.getClass()` ;
- Η μέθοδος `getName()`
`String name = x.getClass().getName()` ;
(ένθεση κλήσεων μεθόδων)