

# Υ325 - Αντικειμενοστρεφής Τεχνολογία

---

2021

Τμ. ΗΜ&ΤΥ - Πανεπιστήμιο Πατρών

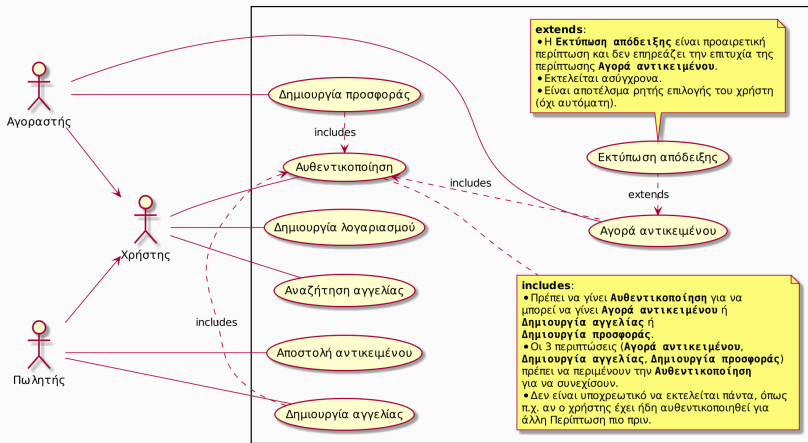
## **Εβδομάδα 9 - 09/12/2021**

---

# Java GUI programming - Calculator

---

# UML Use Case extend Vs include



Εικόνα 1: Παράδειγμα UML Use Case. Ένα απλοποιημένο ebay.

# UML Use Case extend Vs include

**Extend** is a directed relationship that specifies how and when the behavior defined in usually supplementary (optional) **extending use case** can be inserted into the behavior defined in the **extended use case**. (UML Use Case Extend)

Use case **include** is a directed relationship between two use cases which is used to show that behavior of the **included** use case (the addition) is inserted into the behavior of the **including** (the base) use case. (UML Use Case Include)

# StringBuilder

Στη Java τα αντικείμενα `String` είναι immutable, δηλ. είναι αμετάβλητα.

```
String name = "Μήτσος";  
//αν και φαίνεται πως τροποποιούμε τη name, στην  
// πραγματικότητα κατασκευάζεται νέο αντικείμενο String  
name = name + " Μήτσου";
```

Με την `StringBuilder` έχουμε αλφαριθμητικά που είναι mutable.

```
StringBuilder sb=new StringBuilder("Μήτσος");  
sb.append(" Μήτσου"); //τροποποιήθηκε η sb
```

Παράδειγμα στο JavaTutor

Μπορούμε να χρησιμοποιήσουμε την `StringBuilder` για να κατασκευάσουμε τους αριθμούς στην αριθμομηχανή μας.

- Δημιουργία του GUI (της γραφικής διεπαφής)
  - αποτελείται από `JFrame`, `JButtons`, `TextField`
  - για τη διάταξη των στοιχείων χρησιμοποιούμε τις στρατηγικές που ξέρουμε:
    - `BorderLayout` και `GridLayout`
    - `GridBagLayout`
- Μετατροπή έκφρασης από μορφή infix σε μορφή postfix.
  - ο αλγόριθμος δίνεται στις διαφάνειες της διάλεξης.

Το `GridBagLayout` είναι πιο ευέλικτο και επιτρέπει να έχουμε “ελαστικά” γραφικά στοιχεία, που προρμίζονται δηλαδή στο διαθέσιμο χώρο.

# Δημιουργία του GUI - GridBagLayout

```

    JTextField display = new JTextField();
    GridBagLayout gridBagLayout = new GridBagLayout();
    // το gbc καθορίζει τους κανόνες που ισχύουν όταν τοποθετούμε
    // το στοιχείο στο layout
    GridBagConstraints gbc = new GridBagConstraints();
    jPanel.setLayout(gridBagLayout);
    //τί θα γίνει αν ο κοντέινερ είναι μεγαλύτερος από το στοιχείο
    //εδώ, το στοιχείο θα μεγαλώσει ώστε να καλύψει όλο το διαθέσιμο
    //χώρο
    gbc.fill = GridBagConstraints.BOTH;
    // default weight=0. Πόσο από το διαθέσιμο χώρο θα πιάσει
    // το γραφικό στοιχείο
    gbc.weightx = 1.0;
    gbc.weighty = 1.0;
    // πόσα κελιά υπάρχουν στη γραμμή
    gbc.gridwidth = 5;
    jPanel.setLayout(gridBagLayout);
    jPanel.add(display, gbc);

```



# Δημιουργία του GUI - Δομή της εφαρμογής

Προσπαθούμε να εφαρμόσουμε τις αρχές του προγραμματισμού OOP στο πρόβλημα: κληρονομικότητα, πολυμορφισμός, ενθυλάκωση.

- Ομαδοποίηση κοινών χαρακτηριστικών σε υπερκλάσεις, π.χ. τα κουμπιά μοιράζονται κάποια λειτουργικότητα, αλλά έχουν και διαφορές (εξειδίκευση).
- Ελαχιστοποίηση της χρήσης πεδίων και μεθόδων που είναι `static` και η ορατότητα των στοιχείων του προγράμματος να είναι η απαραίτητως αναγκαία.
- Χρήση του μηχανισμού χειρισμού εξαιρέσεων για την αντιμετώπιση προβληματικών καταστάσεων.

Ο αντικειμενοστρεφής προγραμματισμός είναι στενά συνυφασμένος με τη ευρεία χρήση γραφικών διεπαφών, από τη δεκαετία '80 και ιδιαίτερα με την εμφάνιση των Windows. Η συγγραφή GUI προσφέρεται για την εφαρμογή των αρχών του OOP.

# Ο Αντικειμενοστρεφής Προγραμματισμός είναι εύκολος



**Εικόνα 2:** Στην πραγματικότητα δεν το πίστευε κανείς αυτό ούτε το 1991, αλλά η ευρεία χρήση OOP συνέπεσε με την διάδοση των GUI. (Hardcore Software, 2021)

## Μετατροπή infix σε postfix

Ο αλγόριθμος περιγράφεται αναλυτικά στις διαφάνειες.

Το πρότζεκτ αποτελείται από δυο συστατικά, το GUI και τη μηχανή μετατροπής της έκφρασης σε postfix.

Μπορούμε να ξεκινήσουμε με δύο ανεξάρτητα πρότζεκτ, ένα για το GUI και ένα για τη μηχανή μετατροπής σαν μια κλάση που υλοποιεί τον αλγόριθμο και τον εφαρμόζει σε ένα `String` εισόδου (την έκφραση που παίρνουμε από το GUI).

# Infix to Postfix

1. Διάτρεξε την έκφραση `infix` από τα αριστερά προς τα δεξιά.
2. Αν βρεις τελεστέο, σπρώξτον στη στοίβα `postfix`.
3. Αν είναι τελεστής, τότε:
  - Αν η στοίβα `operators` είναι άδεια, σπρώξτον στη στοίβα `postfix`.
  - Αλλιώς, αν η στοίβα `operators` δεν είναι άδεια,
    - Αν έχει  $\geq$  προτεραιότητα από τον τελεστή που βρίσκεται στην `operators`, σπρώξτον στην `operators`
    - Αλλιώς, μέχρι να αδειάσει η `operators` ή να βρεις τελεστή με  $\geq$  προτ.  $\geq$  αυτού που έχεις, βάλε τους τελεστές από την `operators` στην `postfix`. Σπρώξε τον τελεστή στην `operators`.
4. Αν βρεις αριστερή παρένθεση, σπρώξτη στη στοίβα `operators`.
5. Αν βρεις “)”, βάλε τους τελεστές από την `operators` στην `postfix`, μέχρι να βρεις την “(” (απόρριψε τες).
6. Αν στο τέλος της επανάληψης η `operators` έχει τελεστές, βάλτους στην `postfix`.