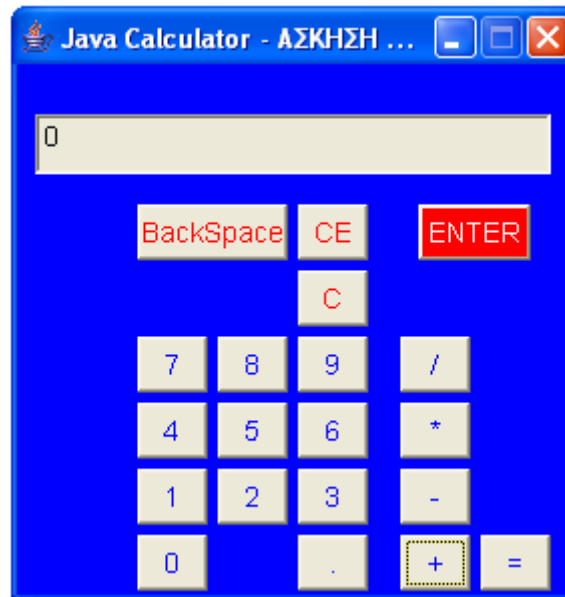


## 12.8. Άσκηση 8 - [Κληρονομικότητα - Αξιολόγηση εναλλακτικών υλοποιήσεων] (έκδοση 2004)

### 12.8.1. Περιγραφή

Σας δίνεται μια διαφορετική έκδοση (έκδοση 2) της κλάσης `CalculatorGui` η οποία εμφανίζει την γραφική διεπαφή που βλέπετε στο σχήμα 8.1. Η διεπαφή αυτή σας δίνεται με τη μορφή Java κλάσεων σε bytecodes στο αρχείο *CalcGuiV2Exer8.zip* ή *CalcGuiV1Exer8.jar* (δες δικτυακό τόπο του βιβλίου).



Σχήμα 8.1. Γραφική διεπαφή της έκδοσης 2 του `CalculatorGui`.

Η 2<sup>η</sup> έκδοση της κλάσης `CalculatorGui` έχει τον παρακάτω δημιουργό

```
public CalculatorGui(Operand op, Adder add,
    Substracter sub, Multiplier mul, Divider div,
    ResultPresenter eq);
```

Η συμπεριφορά του πλήκτρου + όταν αυτό πατηθεί ορίζεται για την 1<sup>η</sup> έκδοση από την παρακάτω γραμμή κώδικα

```
Calc.add.operate();
```

Στα πλαίσια της άσκησης αυτής θα αναπτύξετε μια δεύτερη έκδοση της αριθμομηχανής σας χρησιμοποιώντας την γραφική διεπαφή που σας δίνεται. Θα ορίσετε κατάλληλα την κλάση `Calc` και `Operand` ώστε ο χρήστης να μπορεί να κάνει τις βασικές πράξεις.

### 12.8.2. Στόχος

Εφαρμογή κληρονομικότητας στην αριθμομηχανή. Αξιολόγηση εναλλακτικών υλοποιήσεων.

**Χρόνος εκτέλεσης:** Κατά την διάρκεια της μελέτης του κεφαλαίου 8 ή με την ολοκλήρωση του. Μόνο μετά την εκτέλεση της άσκησης 6.

### 12.8.3. Μεθοδολογία εκτέλεσης

Η άσκηση δίνεται σαν ένα σύνολο από δραστηριότητες που πρέπει να εκτελέσετε μόνοι σας. Σε κάθε δραστηριότητα σας δίνονται οδηγίες και βοήθεια ώστε να ξεπεράσετε τα δύσκολα σημεία.



#### Δραστηριότητα 1

Αναπτύξτε το πιο απλό πρόγραμμα με όνομα **Calc.java** που θα εμφανίζει στην οθόνη τη γραφική διεπαφή CalculatorGui. Εκτελέστε τις παρακάτω ενέργειες:

**Y** Αποσυμπιέστε το αρχείο CalcGuiEx8v2.zip και βάλτε τα αρχεία που περιέχει στο ευρετήριο εργασίας σας.

**Y** Δημιουργήστε ένα πρόγραμμα με όνομα **Calc.java** που θα εμφανίζει την γραφική διεπαφή στην οθόνη.

**Χρησιμοποιήστε την κλάση Operand που αναπτύξατε στα πλαίσια της άσκησης 6.**

Δηλώστε απλά τις κλάσεις των operators όπως παρακάτω

```
class Adder {
    public void operate() {
        System.out.println("Addition");
    }
}
```

Εκτελέστε το πρόγραμμα που μόλις αναπτύξατε. Στην οθόνη θα δείτε την γραφική διεπαφή της αριθμομηχανής. Ελέγξτε την λειτουργικότητα της.



#### Δραστηριότητα 2


Τρέξτε την εφαρμογή σας και πιέστε το πλήκτρο BackSpace. Τι παρατηρείτε; Αιτιολογήστε. Τροποποιήστε τον κώδικα σας για να διορθώσετε την συμπεριφορά της εφαρμογής σας.



#### Δραστηριότητα 3


Ορίστε το σώμα της κλάσης Adder ώστε η αριθμομηχανή σας να εκτελεί την πράξη της πρόσθεσης.

**Y E1.** Εκτελέστε την εφαρμογή σας. Πιέστε το πλήκτρο +. Τι παρατηρείτε; Αιτιολογήστε. Τροποποιήστε τον κώδικα σας για να διορθώσετε την συμπεριφορά της εφαρμογής σας.

 **E2.** Ορίστε τα σώματα των υπόλοιπων κλάσεων που αντιστοιχούν στις βασικές πράξεις.



#### Δραστηριότητα 4

 **E1.** Πως ορίζεται η συμπεριφορά που έχει το πλήκτρο + στο κλικ του mouse για την πρώτη και πως για την δεύτερη έκδοση της `CalculatorGui`;

 **E2.** Ποία από τις δυο εκδόσεις είναι πιο ευέλικτη και γιατί;



#### Δραστηριότητα 5

Ορίστε τις κλάσεις `Adder`, `Substracter`, `Divider`, `Multiplier` και `ResultPresenter` σαν απογόνους της κλάσης `Operator` η οποία δίνεται στη συνέχεια

```
abstract class Operator {
    public abstract void operate();
}
```

Η μέθοδος `operate` προσδιορίζει την διεργασία που αναπαριστά ο εκάστοτε τελεστής.




#### Δραστηριότητα 6

Δημιουργήστε ένα package στο BlueJ με τις παραπάνω κλάσεις και ελέγξτε την ορθότητα τους.



#### Δραστηριότητα 7 - επέκταση λειτουργικότητας

Σας δίνεται ο πηγαίος κώδικας της γραφικής διεπαφής `CalculatorGui` που χρησιμοποιήσατε παραπάνω. (αρχείο [CalcGuiV2Exer8Source.zip](#)) Μελετήστε τον κώδικα.

 Προσθέστε πλήκτρα για τις βασικές λειτουργίες μνήμης. Προσθέστε πλήκτρο για την εισαγωγή του  $\pi = 3,14\dots$



#### Δραστηριότητα 8 - βελτίωση πηγαίου κώδικα

Προσπαθήστε να δώσετε μια πιο συμπαγή έκδοση για τον πηγαίο κώδικα.