

Object Oriented Programming Course

Activity No 3

1. Στόχος

Η δραστηριότητα έχει στόχο την εξοικείωση με:

- Τις έννοιες της κλάσης και του στιγμιότυπου
- Την δόμηση προγράμματος ως μια συνάθροιση αντικειμένων
- Αξιοποίηση έτοιμων κλάσεων στην διαδικασία ανάπτυξης εφαρμογής
- Αξιοποίηση κλάσεων της βασικής βιβλιοθήκης της java
- Την αξιοποίηση κλάσεων τρίτου κατασκευαστή οι οποίες δίνονται με την μορφή βιβλιοθηκών τύπου jar.
- Incremental development τεχνική στην ανάπτυξη αντικειμενοστρεφούς εφαρμογής

2. Αναμενόμενα αποτελέσματα

Με την ολοκλήρωση της δραστηριότητας αυτής θα έχετε μια λειτουργούσα έκδοση της RPN Calculator που θα επιτρέπει τον υπολογισμό εκφράσεων μέσα από γραφική διεπαφή.

Το βασικότερο όμως θα έχετε εξοικειωθεί με βασικές έννοιες του αντικειμενοστρεφούς παραδείγματος προγραμματισμού και την incremental development τεχνική. Άκρως απαραίτητο εφόδιο για την αποτελεσματική συνέχεια της εκπαιδευτικής διαδικασίας του μαθήματος.

3. Οδηγίες εκτέλεσης Δραστηριότητας

Η δραστηριότητα αποτελείται από μια Δράση (Integrate existing components) και μια άσκηση (Towards a gui-based RPN Calculator).

Κάθε ένα iteration προσθέτει επιπλέον λειτουργικότητα στο αποτέλεσμα του προηγούμενου Iteration. Αρα **για κάθε iteration χρησιμοποιείτε ως βάση τον κώδικα που αναπτύξατε στο προηγούμενο iteration.**

Η Δράση δεν έχει σχέση με τα iterations. Έχει ως στόχο να σας εξασκήσει με το integration των συνθετικών ενός συστήματος.

Σας συνιστώ να εκτελέσετε πρώτα τα δύο πρώτα iterations (1 και 2) της Άσκησης και μετά την Δράση. Η Δράση εκτός του ότι θα σας εξοικειώσει με την διαδικασία της σύνθεσης (integration) θα σας βοηθήσει και για την εκτέλεση των υπόλοιπων iterations.

Προχωράτε στο επόμενο iteration μόνο όταν έχετε ολοκληρώσει επιτυχώς το προηγούμενο.

4. Δράση - Integrate existing components

4.1 Περιγραφή της Άσκησης

Στόχος της άσκησης είναι να δημιουργήσετε μια αριθμομηχανή με γραφική διεπαφή που υποστηρίζει περιορισμένη λειτουργικότητα. Μπορεί να δεχθεί μια έκφραση που περιέχει τους τελεστές + και =, υπολογίζει την τιμή της και την δίνει στον χρήστη.

Για το βήμα αυτό σας δίνονται ως βιβλιοθήκες όλες οι κλάσεις που απαιτούνται και εσείς θα πρέπει να ασχοληθείτε μόνο με τη σύνθεση τους (integration) ώστε να δημιουργηθεί η πρώτη δική σας RPN Calculator.

Μια προσεκτική μελέτη του προβλήματος μας οδήγησε στον ορισμό της δομής του προγράμματος ως συνάθροισης από τα παρακάτω αντικείμενα.

RPNCalculator

Πρώτο αντικείμενο η κλάση που θα περιέχει την `main()`. Αυτή αναπαριστά το πρόγραμμα και για τον λόγο αυτό θα την ονομάσουμε `RPNCalculator`. Έχει την ευθύνη να δημιουργήσει τα αντικείμενα που απαρτίζουν το σύστημα και να αποκαταστήσει τις μεταξύ τους εξαρτήσεις.

CalculatorGui

Δεύτερο βασικό αντικείμενο είναι αυτό που επιτρέπει στον άνθρωπο να χρησιμοποιήσει την υπηρεσία που περιγράφει το πρόγραμμά μας και θα προσφέρει το σύστημα που θα εκτελεί το πρόγραμμά μας. Επιλέξαμε στη φάση αυτή η επικοινωνία ανθρώπου με την μηχανή να είναι μέσο γραφικής διεπαφής (**Σχήμα 1**). Προσέξτε όμως ότι η γραφική διεπαφή δεν εκτελεί καμία ουσιαστική λειτουργία εκτός από την αλληλεπίδραση με τα αντικείμενα που υλοποιούν την λειτουργικότητα της αριθμομηχανής μας. Απλά θα στέλνει μηνύματα στα αντικείμενα που έχουν επιφορτιστεί με τις κατάλληλες αρμοδιότητες ώστε η συνεργασία τους να μας δώσει την υπηρεσία που θέλουμε σε επίπεδο συστήματος.

Η κλάση `CalculatorGui` μας δίνει και τη δυνατότητα να κατανοήσουμε τις αρμοδιότητες που έχουν ανατεθεί στα αντικείμενα που απαρτίζουν την μηχανή (`engine`) της αριθμομηχανής μας για να τα διακρίνουμε από αυτά που απαρτίζουν την γραφική διεπαφή της.

Operand

Βασική κλάση που αναπαριστά δομή και συμπεριφορά του τελεστέου που είναι βασικό συνθετικό της εισόδου που δέχεται το σύστημα. Δίνει τη δυνατότητα διαμόρφωσης του τελεστέου από τη γραφική διεπαφή οπότε πρέπει να υποστηρίζει την προσθήκη ψηφίου στην τιμή του τελεστέου (θα ονομάσουμε την αντίστοιχη μέθοδο **`addDigit`**), την διαγραφή του τελευταίου εισαχθέντος ψηφίου (θα ονομάσουμε την αντίστοιχη μέθοδο **`deleteLastDigit`**), την ολοκλήρωση του τελεστέου και την αποθήκευση του στην στοίβα (θα ονομάσουμε την αντίστοιχη μέθοδο **`complete`**) κλπ.

Adder, Divider, Multiplier, και Subtractor

Η κλάση `Adder` αναπαριστά τον αθροιστή που δουλεύει πάνω σε στοίβα. Την κλάση αυτή έχουμε ήδη μελετήσει αναλυτικά. Αντίστοιχες είναι και οι άλλες κλάσεις που αναπαριστούν τους υπόλοιπους τελεστές που υποστηρίζει η έκδοση αυτή.

ResultPresenter

Η κλάση `ResultPresenter` αναπαριστά τον τελεστή = και η λειτουργικότητα της είναι η εμφάνιση στη στοίβα του τελευταίου στοιχείου της στοίβας.

Όλες οι παραπάνω κλάσεις σας δίνονται με την μορφή δύο (2) αρχείων `jar`.

A. [CalcComponents.jar](#)

<http://seg.ece.upatras.gr/Courses/OT/activities/activity3/CalcComponents.zip>

Περιέχει τις κλάσεις `Operand`, `Adder`, `Divider`, `Multiplier`, `Subtractor` and `ResultPresenter`.

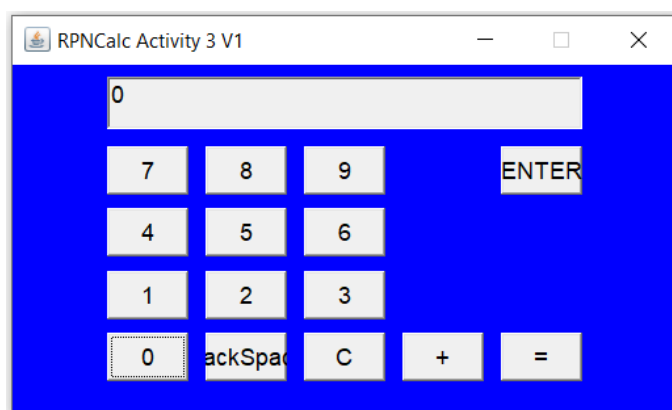
Η κλάση Operand βρίσκεται στο πακέτο engine. Οι κλάσεις Adder, Divider, Multiplier, Subtractor and ResultPresenter βρίσκονται στο πακέτο operators.

B. CalcGuiActivity3V1.jar

<http://seg.ece.upatras.gr/Courses/OT/activities/activity3/CalcGuiActivity3V1.zip>

Περιέχει την κλάση CalculatorGui.

Βρίσκεται στο πακέτο gui και ο δημιουργός της εμφανίζει στην βασική έξοδο μια γραφική διεπαφή όπως αυτή στο **Σχήμα 1**.



Σχήμα 1. Η γραφική διεπαφή που δημιουργεί ο constructor της CalculatorGui.

Μια τεκμηρίωση των παραπάνω κλάσεων μπορείτε να βρείτε στο <http://seg.ece.upatras.gr/Courses/OT/activities/activity3/doc/index.html>

Μια σύντομη αναφορά στα αρχεία τύπου jar θα βρείτε στην ενότητα 4.3.

4.2 Οδηγίες Εκτέλεσης Δράσης

Εκτελέστε τις παρακάτω ενέργειες:

Δ.1) Δημιουργήστε ένα project στο Eclipse. Δημιουργήστε κάτω από το ευρετήριο src του project σας ένα package με όνομα tester ή ότι άλλο όνομα θέλετε.

Δ.2) Αντιγράψτε τις δυο παραπάνω βιβλιοθήκες στο ευρετήριο του project που δημιουργήσατε.

Δ.3) Ενημερώστε το Eclipse (ή όποιο IDE χρησιμοποιείτε) για την θέση στην οποία θα βρει τις βιβλιοθήκες που χρησιμοποιεί το project σας.

Για ενημέρωση του BlueJ για την βιβλιοθήκη κλάσεων επιλέξτε Tools-Preferences-Libraries-Add και επιλέξτε την βιβλιοθήκη CalcGuiV1Exer6.jar

Για την αντίστοιχη ενημέρωση στο Eclipse επιλέξτε Project-Properties-Java Build Path-Libraries-Add external JARs και επιλέξτε την βιβλιοθήκη που θα έχετε τοποθετήσει στο directory του project σας.

Δ.4) Δημιουργήστε την κλάση που αναπαριστά το πρόγραμμα σας. Ονομάστε την RPNCalculator ή όπως αλλιώς θέλετε. Ορίστε την main().

Δ.5) Ορίστε την μέθοδο main.

Η main θα πρέπει να δημιουργήσει ένα στιγμιότυπο της γραφικής διεπαφής. Αυτό θα το κάνει καλώντας τον δημιουργό της CalculatorGui.

Προσέξτε πως για την κλήση του δημιουργού θα πρέπει να έχετε πρώτα δημιουργήσει στιγμιότυπα των οποίων αναφορές πρέπει να περάσετε στον δημιουργό κατά την κλήση του.

Για την αξιοποίηση κλάσης των βιβλιοθηκών θα πρέπει να περιλάβετε στον πηγαίο κώδικα την αντίστοιχη πρόταση **import**. Το Eclipse σας δίνει βοήθεια προς την κατεύθυνση αυτή.

Δ.6) Ελέγξτε το πρόγραμμα σας για την ορθή λειτουργία του. Θα πρέπει να εμφανίζει την γραφική διεπαφή η οποία θα σας επιτρέψει να εισάγεται τελεστέους και να κάνετε την πράξη της πρόσθεσης (+) και της παρουσίασης του αποτελέσματος (=). Πατήστε το ENTER για να σηματοδοτήσετε τον τέλος του τελεστέου.

Παρατηρήστε ότι το πρόγραμμα τυπώνει την τρέχουσα τιμή του τελεστέου στην βασική έξοδο και όχι στο display της γραφικής διεπαφής. Στη φάση αυτή δεν μπορείτε να επέμβετε σε αυτό. Θα το αντιμετωπίσετε στα πλαίσια της άσκησης.

4.3 Τα αρχεία τύπου jar (Java Archive files)

Τα αρχεία τύπου jar (Java Archive files) αποτελούν την καθιερωμένη μέθοδο ενσωμάτωσης πολλών class αρχείων σε ένα. Το αρχείο αυτό μπορεί να χρησιμοποιηθεί ως βιβλιοθήκη κλάσεων για την ανάπτυξη Java εφαρμογών.

Για περισσότερες πληροφορίες σχετικά με τα αρχεία τύπου jar και τον τρόπο δημιουργίας και χρήσης τους μπορείτε να ανατρέξετε στη διεύθυνση:

<https://docs.oracle.com/javase/tutorial/deployment/jar/basicsindex.html>

4.3.1 Χρησιμοποιώντας αρχεία jar με το SDK της Sun.

Για να αξιοποιήσουμε σε ένα πρόγραμμα κλάσεις που περιέχονται σε μια βιβλιοθήκη με όνομα "myLibrary.jar" ενημερώνουμε τον Java compiler για το όνομα και το ερευνηριο στο οποίο βρίσκεται η βιβλιοθήκη.

Για παράδειγμα, αν υποθέσουμε ότι η κλάση που αναπαριστά το πρόγραμμα είναι η MyProgram και ότι η βιβλιοθήκη myLibrary.jar βρίσκεται στον κατάλογο c:\libraries καλούμε τον java compiler όπως παρακάτω:

```
javac -classpath ".;c:\libraries\myLibrary.jar" *.java
```

και εκτελούμε το πρόγραμμα μας καλώντας τον java interpreter όπως παρακάτω:

```
java -classpath ".;c:\libraries\myLibrary.jar" MyProgram
```

Για πληροφορίες σχετικά με τη μεταβλητή περιβάλλοντος "classpath" μπορείτε να ανατρέξετε στην ιστοσελίδα :

<http://java.sun.com/j2se/1.3/docs/tooldocs/win32/classpath.html>

5 Άσκηση - Towards a gui-based RPN Calculator

Εφαρμόστε την incremental development τεχνική για την ανάπτυξη μιας λειτουργούσας έκδοσης της RPN Calculator.

Σας συνιστώ να ακολουθήσετε τα παρακάτω βήματα για την καλύτερη κατανόηση των εννοιών. Σε κάθε περίπτωση εσείς αποφασίζετε σε ποια από αυτά θα επιμείνετε με κριτήριο την εξοικείωση σας με τον προγραμματισμό.

Περιγράφονται στη συνέχεια ένα σύνολο από επαναλήψεις (iterations) που οδηγούν σε μια λειτουργούσα αριθμομηχανή η οποία θα χρησιμοποιεί μόνο την γραφική διεπαφή ως έτοιμη κλάση. Την δική σας γραφική διεπαφή θα αναπτύξετε σε επόμενη δραστηριότητα.

5.1 Iteration No 1 – Πρόσθεση δύο αριθμών με χρήση στοίβας

Η πρώτη έκδοση που θα είναι αποτέλεσμα του πρώτου iteration θα έχει πολύ απλή λειτουργικότητα. Θα δημιουργεί 2 στιγμιότυπα της Double, θα τα βάζει στην στοίβα, στην συνέχεια θα τα βγάζει, θα τα προσθέτει και το αποτέλεσμα θα το βάζει πίσω στην στοίβα. Από εκεί θα το παίρνει και θα το εμφανίζει στην οθόνη.

Αν δουλεύετε στο BlueJ

Δημιουργήστε ένα project με όνομα Calc. Στη συνέχεια επιλέξτε από το μενού Edit->New Class και δώστε όνομα Calc στην κλάση σας. Επιλέξτε το ευρετήριο στο οποίο θέλετε να βάλει το BlueJ τις κατασκευές του προγράμματος σας. Το BlueJ εμφανίζει στο κυρίως πλαίσιο του ένα ορθογώνιο με όνομα Calc. Αυτό είναι η γραφική αναπαράσταση της κλάσης σας μέσα από την οποία θα επικοινωνείτε με αυτήν. Κάντε δεξί κλικ πάνω στην κλάση Calc και επιλέξτε Open Editor. Στον editor θα δείτε έτοιμο κώδικα που σας δίνει το BlueJ για να τον χρησιμοποιήσετε ως βάση για τη συγγραφή της κλάσης σας. Διαγράψτε το περιεχόμενο της κλάσης και αντικαταστήστε τον με τον κώδικα

Αν δουλεύετε στο Eclipse

1. Δημιουργήστε ένα Project.
Επιλέξτε File ->New->Java Project. Δώστε όνομα στο project και επιλέξτε Finish και στην συνέχεια Don't Create.
2. Ορίστε την κλάση Calc.
Στον package explorer επιλέξτε το project και στη συνέχεια src. Δεξί κλικ και New-> Class. Δώστε όνομα Calc και ζητήστε την δημιουργία της main() (**Σχήμα 2**). Πατήστε Finish. Δώστε το σώμα της κλάσης σας. Θα πρέπει να δουλέψετε στο default package.
3. Εκτελέστε το πρόγραμμα σας.
Επιλέξτε την κλάση Calc δεξί κλικ

Αν έχετε δυσκολία στον ορισμό της κλάσης Calc θα τον βρείτε στις διαφάνειες με τίτλο LegoApproach6_* partOf (ενότητα 1st iteration).

Name:

Modifiers: public package private protected
 abstract final static

Superclass:

Interfaces:

Which method stubs would you like to create?
 public static void main(String[] args)

Σχήμα 2 Δημιουργία κλάσης Calc στο Eclipse.

Εκτελέστε το πρόγραμμα και ελέγξτε ότι λειτουργεί κανονικά. Λογικά θα σας δίνει το αποτέλεσμα της πρόσθεσης στην κύρια είσοδο.

5.2 Iteration No 2 - Increase reusability

Η δεύτερη αυτή έκδοση δεν θα έχει πρόσθετη λειτουργικότητα αλλά στόχο να βελτιώσει τον κώδικα ώστε να αξιοποιεί τα πλεονεκτήματα της αντικειμενοστρεφούς προσέγγισης για αύξηση της επαναχρησιμοποίησης (reusability) και της τμηματοποίησης με την έννοια του αρθρωτού (modularity).

Δημιουργήστε ένα αντίγραφο του project του 1^{ου} iteration και δουλέψτε πάνω σε αυτό με στόχο να αυξήσετε το reusability στο πρόγραμμα σας.

Ο παρακάτω κώδικας της main() είναι πιθανό να ξαναχρησιμοποιηθεί αν θέλουμε να κάνουμε πάλι πρόσθεση δύο αριθμών. Την διαπίστωση αυτή θα χρησιμοποιήσουμε ως παράδειγμα για να δούμε την εφαρμογή βασικών εννοιών του αντικειμενοστρεφούς προγραμματισμού και πως αυτές επηρεάζουν την ποιότητα του κώδικα.

```
Double d3 = st.pop();
Double d4 = st.pop();
st.push(d3+d4);
```

Έχετε τις παρακάτω επιλογές:

1. Χρήση μεθόδου

Μια επιλογή είναι να ορίσετε μια μέθοδο στην Calc που θα της αναθέσετε αυτό το έργο. Στην επιλογή αυτή η γνώση παραμένει στην κλάση Calc.

2. Χρήση Αντικειμένου

Η άλλη είναι να έχετε ένα νέο αντικείμενο το οποίο θα προσφέρει αυτή την υπηρεσία την οποία θα αξιοποιεί η main() της Calc. Το αντικείμενο αυτό μπορεί να είναι κλάση ή στιγμιότυπο.

Δοκιμάστε και τις δύο επιλογές. Και στις δύο περιπτώσεις η κλάση που θα δημιουργήσετε είναι λογικό να ονομασθεί **Adder**. Επιπλέον θα σας πρότεινα να ονομάσετε **operate** τη μέθοδο που υλοποιεί την υπηρεσία.

Η Calc καθώς τώρα δεν έχει τη γνώση να κάνει πρόσθεση θα πρέπει να συνεργαστεί με το νέο αντικείμενο που θα έχει τη γνώση αυτή. Η συνεργασία

αυτή είναι διαφορετική για την περίπτωση που το αντικείμενο είναι κλάση από την περίπτωση που αυτό είναι στιγμιότυπο.

E1. Θα πρέπει να υλοποιήσετε και τις δύο επιλογές.

Στην τελική σας έκδοση, αυτή που θα χρησιμοποιήσετε ως βάση για την επόμενη έκδοση, υλοποιήστε τη μέθοδο `operate()` ως μέθοδο στιγμιότυπου.

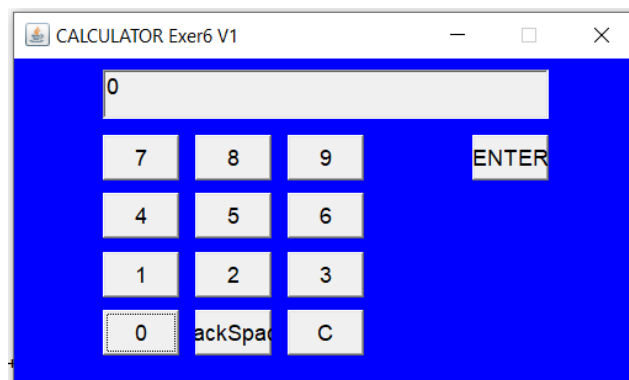
Την όλη διαδικασία συζητήσαμε στις διαλέξεις. Περιγράφεται στις διαφάνειες με τίτλο `LegoApproach6_* partOf` (ενότητα 2nd iteration - Increase modularity).

5.3 Iteration No 3 – Εισαγωγή γραφικής διεπαφής και κλάσης `Operand`.

Δημιουργήστε ένα αντίγραφο του project του 2^{ου} iteration και δουλέψτε πάνω σε αυτό.

Στην 3^η αυτή έκδοση θα προσθέσουμε στο πρόγραμμα γραφική διεπαφή η οποία όμως μας δίνεται έτοιμη. Η έκδοση αυτή θα μας επιτρέψει να διαμορφώνουμε από την γραφική διεπαφή έναν τελεστέο (`Operand`) και να τον βάζουμε στη στοίβα. Η απαίτηση του να διαμορφώνουμε τον τελεστέο από τη γραφική διεπαφή μας οδηγεί στον ορισμό της κλάσης `Operand`. Στην Δράση την είχαμε έτοιμη. Εδώ πρέπει να την ορίσουμε.

Χρησιμοποιήστε την κλάση `CalculatorGui` που σας δίνεται με το αρχείο [CalcGuiV1Exer6.jar](#). Η γραφική διεπαφή είναι όπως αυτή στο **Σχήμα 3**.



Σχήμα 3 Η γραφική διεπαφή του 3ου iteration.

Ο δημιουργός της `CalculatorGui` δέχεται ως όρισμα αναφορά τύπου `Operand`.

```
CalculatorGui(Operand op);
```

Την αναφορά αυτή χρησιμοποιεί η γραφική διεπαφή για να διαμορφώνει τον τελεστέο καθώς ο χρήστης θα πατάει τα πλήκτρα των ψηφίων. Αυτό σημαίνει πως η γραφική διεπαφή συνεργάζεται στενά με την `Operand`.

Ορίστε την δική σας κλάση `Operand` και υλοποιήστε μια έκδοση της RPN Calculator η οποία θα επιτρέψει στον χρήστη να εισάγει ένα `operand` και να τον βάζει στην στοίβα.

Για την διαμόρφωση του operand θα πρέπει να ορίσετε την μέθοδο `addDigit()`. Για την τοποθέτηση του τελεστέου στην στοίβα (πλήκτρο ENTER) πρέπει να ορίσετε τη μέθοδο `complete()`.

E2. Τροποποιήστε το πρόγραμμά σας ώστε να ελέγξετε αν ο operand που ορίσατε από την γραφική διεπαφή μπήκε στη στοίβα.

Στην επόμενη έκδοση θα προσθέσουμε στο πρόγραμμα και τους τελεστές για να κάνουμε πράξεις.

Την όλη διαδικασία συζητήσαμε στις διαλέξεις. Περιγράφεται στις διαφάνειες με τίτλο `LegoApproach6_* partOf` (ενότητα 3rd Iteration – Adding a Gui).

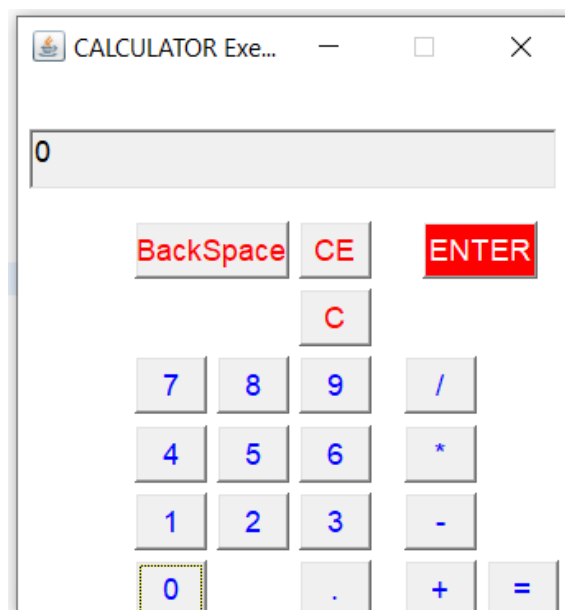
5.4 Iteration No 4 - Adding Operators

Στην έκδοση αυτή θα προσθέσουμε τη λειτουργικότητα των πράξεων και της εμφάνισης του αποτελέσματος.

Δημιουργήστε ένα αντίγραφο του project του 3^{ου} iteration και δουλέψτε πάνω σε αυτό. Καθώς όμως η γραφική διεπαφή της έκδοσης 3 δεν έχει τελεστές θα αντικαταστήσετε την γραφική διεπαφή με αυτή που περιέχεται στο αρχείο [CalcGuiV2Exer8.jar](#). Η κλάση έχει τον παρακάτω δημιουργό

```
CalculatorGui(Operand op, Adder add, Subtractor sub, Multiplier mul, Divider div, ResultPresenter rp)
```

Ο δημιουργός αυτός εμφανίζει στην οθόνη μία γραφική διεπαφή όπως αυτή στο **Σχήμα 4**.



Σχήμα 4. Η γραφική διεπαφή του 4ου iteration.

Ορίστε κατάλληλα τις δικές σας κλάσεις για τους τελεστές και προχωρήστε στην σύνθεση της εφαρμογής σας. Την Adder την έχετε έτοιμη. Ανάλογα θα ορίσετε και τις υπόλοιπες.

Την όλη διαδικασία συζητήσαμε στις διαλέξεις. Περιγράφεται στις διαφάνειες με τίτλο `LegoApproach6_* partOf` (ενότητα 4th Iteration – Adding Operators).

5.5 Iteration No 5 - using inheritance

Στην έκδοση αυτή δεν θα προσθέσουμε επιπλέον λειτουργικότητα. Θα τροποποιήσουμε τον κώδικα ώστε να αξιοποιήσουμε τον μηχανισμό της κληρονομικότητας.

Δημιουργήστε ένα αντίγραφο του project του 4^{ου} iteration και δουλέψτε πάνω σε αυτό.

E3. Αναγνωρίστε σχέση γενίκευσης/εξειδίκευσης στον κώδικα σας και αξιοποιήστε τον μηχανισμό της κληρονομικότητας (inheritance) για να την υλοποιήσετε και να βελτιώσετε την ποιότητα του.

Την όλη διαδικασία συζητήσαμε στις διαλέξεις. Περιγράφεται στις διαφάνειες με τίτλο LegoApproach6_* partOf (ενότητα 5th Iteration – using inheritance).

5.6 Iteration No 6 – using Interface

Στην έκδοση αυτή δεν θα προσθέσουμε επιπλέον λειτουργικότητα. Θα τροποποιήσουμε τον κώδικα ώστε να αξιοποιήσουμε την κατασκευή του interface.

Δημιουργήστε ένα αντίγραφο του project του 5^{ου} iteration και δουλέψτε πάνω σε αυτό.

E4. Ορίστε το interface `OperandIf` έτσι ώστε η κλάση `Operand` να ορίζεται να κάνει implements το interface αυτό.

Την όλη διαδικασία συζητήσαμε στις διαλέξεις. Περιγράφεται στις διαφάνειες με τίτλο LegoApproach6_* partOf (ενότητα 6th Iteration – using interfaces).

5.7 Ερώτηση

Ποια από τις 3 εκδόσεις (iteration 4, 5, 6) θα προτιμήσετε και γιατί; Αιτιολογήστε.

Τοποθετήστε την απάντησή σας σε αρχείο με όνομα `Answer` και τοποθετήστε το στο ευρετήριο του project που θα παραδώσετε.