

Object Oriented Programming Course (CEID_NNY106)

Εργαστηριακή Άσκηση

Gui based RPN Calculator

1. Στόχος

Η Εργαστηριακή Άσκηση (ΕΑ) έχει στόχο την εξοικείωση με:

- Τη δόμηση προγράμματος από ένα σύνολο από συνιστώσες (**system integration**)
- Την αξιοποίηση κλάσεων τρίτου κατασκευαστή οι οποίες δίνονται με την μορφή βιβλιοθηκών τύπου jar
- Την ανάπτυξη κλάσεων με ζητούμενη δομή και συμπεριφορά
- Επέκταση λειτουργικότητας και βελτίωση ποιότητας πηγαίου κώδικα
- Την Incremental development τεχνική στην ανάπτυξη αντικειμενοστρεφούς εφαρμογής
- ένα πιο επαγγελματικό IDE όπως είναι το Eclipse.

RPN Calculator από συνιστώσες

Η ΕΑ βασίζεται στις Δραστηριότητες 6.5 – 6.10 του κεφαλαίου 6 (ενότητες 6.4-6.6) και στις Δραστηριότητες 8.1-8.8 του κεφαλαίου 8 του βιβλίου “Από τις Υπολογιστικές στις Κυβερνο-Φυσικές Διεργασίες και το ΙοΤ: Αντικείμενα και Υπηρεσίες”, Κ. Θραμπουλίδης, ISBN 978-960-418-961-8, 2022, ΕΚΔΟΣΕΙΣ Α. ΤΖΙΟΛΑ & ΥΙΟΙ Α.Ε. <https://sites.google.com/view/fromcomputationalto cyber-physi/home>

2. Αναμενόμενα αποτελέσματα

Με την ολοκλήρωση της ΕΑ θα έχετε μια λειτουργούσα έκδοση της RPN Calculator με γραφική διεπαφή η οποία θα επιτρέπει τον υπολογισμό τιμής εκφράσεων remote polish notation.

Το βασικότερο όμως θα έχετε εξοικειωθεί:

1. με τη διαδικασία του **system integration** με αξιοποίηση έτοιμων συνθετικών που σας δίνονται υπό την μορφή αρχείων τύπου jar σε συνδυασμό με την **incremental development** τεχνική, και,
2. με την **ανάπτυξη συνιστωσών που θα ικανοποιούν δεδομένες προδιαγραφές**

3. Οδηγίες εκτέλεσης ΕΑ

Η ΕΑ αποτελείται από τρεις (3) ασκήσεις:

Άσκηση Νο 1: Σύνθεση της RPN Calculator από έτοιμα συνθετικά (System Integration),

Άσκηση Νο 2: Ανάπτυξη συνθετικών της RPN Calculator engine, και

Άσκηση Νο 3: Ανάπτυξη Γραφικής διεπαφής.

Η Άσκηση Νο 1 έχει ως στόχο να σας εξασκήσει με το integration των συνθετικών ενός συστήματος. Θα σας εξοικειώσει με τη διαδικασία της σύνθεσης (integration) συστήματος και θα σας βοηθήσει στην εκτέλεση των iterations των δύο επόμενων ασκήσεων.

Στην Άσκηση Νο 2 θα έχετε τη δυνατότητα να αναπτύξετε εσείς τα συνθετικά της engine της αριθμομηχανής, τα συνθετικά της δηλαδή εκτός αυτού της γραφικής διεπαφής. Θα δημιουργήσετε έτσι μία έκδοση της engine με τα δικά σας συνθετικά.

Στην Άσκηση Νο 3 θα δημιουργήσετε και την δική σας γραφική διεπαφή.

Οι ασκήσεις 2 και 3 υλοποιούνται με βάση την incremental development τεχνική. Κάθε ένα iteration είτε προσθέτει επιπλέον λειτουργικότητα στο αποτέλεσμα του προηγούμενου iteration είτε βελτιώνει την ποιότητα του κώδικα του προηγούμενου iteration αξιοποιώντας καλύτερα την αντικειμενοστρεφή προσέγγιση. Για τον λόγο αυτό **για κάθε iteration θα χρησιμοποιείτε ως βάση τον κώδικα που αναπτύξατε σε προηγούμενο iteration.**

Μόνο όταν έχετε ολοκληρώσει επιτυχώς ένα iteration προχωράτε στο επόμενο.

4. Άσκηση Νο 1 (System integration)

4.1 Περιγραφή και στόχος της Άσκησης

Στόχος της άσκησης Νο 1 είναι η εξοικείωση με τη διαδικασία του System integration. Αυτό θα γίνει μέσα από την δημιουργία μιας αριθμομηχανής με γραφική διεπαφή που υποστηρίζει περιορισμένη λειτουργικότητα. Μπορεί να δεχθεί μια έκφραση που περιέχει τους τελεστές + και =, υπολογίζει την τιμή της και τη δίνει στον χρήστη.

Την έκδοση αυτή της RPNCalculator **θα αναπτύξετε χρησιμοποιώντας έτοιμα συνθετικά** που σας δίνονται με την μορφή βιβλιοθηκών.

4.1.1 Η βιβλιοθήκη

Σας **δίνονται σε μορφή βιβλιοθηκών όλες οι κλάσεις** που απαιτούνται για την κατασκευή μιας έκδοσης που ικανοποιεί τον στόχο της άσκησης.

Εσείς **θα πρέπει να ασχοληθείτε μόνο με τη σύνθεση τους (integration)** ώστε να δημιουργηθεί η πρώτη δική σας RPN Calculator.

Σας δίδονται οι παρακάτω κλάσεις:

CalculatorGui, **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.

Η κλάση **CalculatorGui** βρίσκεται στο πακέτο **gui**.

Μια **τεκμηρίωση των κλάσεων** αυτών θα βρείτε στην ιστοσελίδα <http://seg.ceid.upatras.gr/Courses/OT/activities/activity3/doc/index.html>

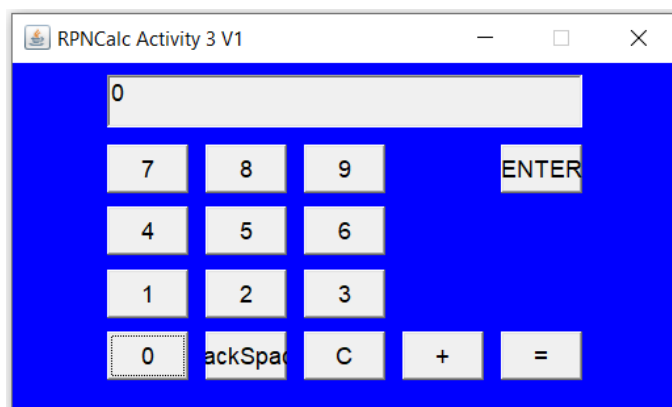
Τις κλάσεις (εκτός από την **CalculatorGui**) θα βρείτε στην **βιβλιοθήκη** η οποία σας δίνεται με την μορφή αρχείου τύπου **jar** (**CalcComponents.jar**). Θα την κατεβάσετε από τον παρακάτω σύνδεσμο

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

Την κλάση **CalculatorGui** θα χρησιμοποιήσετε από την βιβλιοθήκη **CalcGuiActivity3V1.jar**

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

Η **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 σύντομη αναφορά στα αρχεία τύπου **jar** θα βρείτε στην [ενότητα 4.4](#).

Δράση 1.1 – Ορισμός της κλάσης που αναπαριστά το πρόγραμμα σας

Με βάση τα παραπάνω και με τις μέχρι τώρα γνώσεις σας θα πρέπει να είστε σε θέση να ολοκληρώσετε μια πρώτη έκδοση της RPN Calculator.

Στην επόμενη ενότητα σας δίνονται οδηγίες για την ανάπτυξη της άσκησης. Ανατρέξτε σε αυτές μόνο μετά από την δική σας προσπάθεια. Αξιοποιήστε επιλεκτικά όσες λιγότερες από αυτές μπορείτε.

Με την ολοκλήρωση της Δράσης 1 προχωρήστε στην 2 και την 3 για να ολοκληρώσετε την Άσκηση No 1.

Δράση 1.2 – Έλεγχος λειτουργίας

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

Δράση 1.3 – Επέκταση λειτουργικότητας

Τροποποιήστε το πρόγραμμα που αναπτύξατε παραπάνω ώστε να τυπώνει την κατάσταση της στοίβας κάθε φορά που ολοκληρώνεται ένας τελεστής και μπαίνει στην στοίβα. Αυτό χωρίς να επέμβετε στην κλάση Operand που σας δίνεται στο jar.

4.2 Οδηγίες Εκτέλεσης της άσκησης No 1

Σας δίνονται οδηγίες για:

1. [χρήση του Eclipse](#) για την ανάπτυξη του αντίστοιχου project.
2. αξιοποίησης βιβλιοθήκης με την μορφή αρχείου τύπου jar.
3. χρήση των κλάσεων της βιβλιοθήκης.

4.2.1 Δημιουργία project στο Eclipse

Αν επιλέξατε ως IDE το Eclipse εκτελέστε τις παρακάτω ενέργειες για την δημιουργία του Project.

E1.1. Δημιουργήστε Project

Επιλέξτε

File ->New->Java Project.

Δώστε όνομα στο project και επιλέξτε Finish και στην συνέχεια στο παράθυρο Create module-info.java επιλέξτε Don't Create για **να μην δημιουργηθεί module**. Σε νεότερη έκδοση του Eclipse αν η δημιουργία module είναι προεπιλεγμένη απενεργοποιήστε την.

Δημιουργήστε κάτω από το ευρετήριο src του project σας ένα package με όνομα tester ή ότι άλλο όνομα θέλετε.

E1.2. Δημιουργήστε την κλάση που αναπαριστά το πρόγραμμά σας.

Στον package explorer επιλέξτε το project και στη συνέχεια το πακέτο που δημιουργήσατε. Δεξί κλικ και **New-> Class**.

Δώστε όνομα Calc (ή RPNCalculator ή όποιο άλλο θέλετε) και ζητήστε τη δημιουργία της main() (Σχήμα 2). Πατήστε Finish.

E1.3. Ορίστε τη μέθοδο main.

Πρώτα από όλα δοκιμάστε την διαδικασία μεταγλώττισης και εκτέλεσης με μία απλή System.out.println. Προχωρήστε στη συνέχεια στην συγγραφή της main.

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

Για να αξιοποιήσετε κλάσεις της βιβλιοθήκης θα πρέπει να έχετε ενημερώσει κατάλληλα το IDE σας.

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.

The image shows the 'New Class' dialog in Eclipse. The 'Name' field is 'Calc'. The 'Modifiers' section has 'public' selected. The 'Superclass' is 'java.lang.Object'. The 'Interfaces' section is empty. The 'Which method stubs would you like to create?' section has the checkbox for 'public static void main(String[] args)' checked.

Σχήμα 2 Δημιουργία κλάσης Calc στο Eclipse.

4.2.2 Αξιοποίηση βιβλιοθηκών τύπου Jar

Στην ενότητα αυτή δίνονται οδηγίες για αξιοποίηση βιβλιοθηκών τύπου jar. Περισσότερες πληροφορίες για αρχεία τύπου jar θα βρείτε στη ενότητα [Αρχεία τύπου jar \(Java Archive files\)](#)

Για να αξιοποιήσετε βιβλιοθήκες τύπου .jar εκτελέστε τις παρακάτω ενέργειες:

E1.4. – Κατεβάστε τις βιβλιοθήκες

Αντιγράψτε τις βιβλιοθήκες στο ευρετήριο του project που δημιουργήσατε.

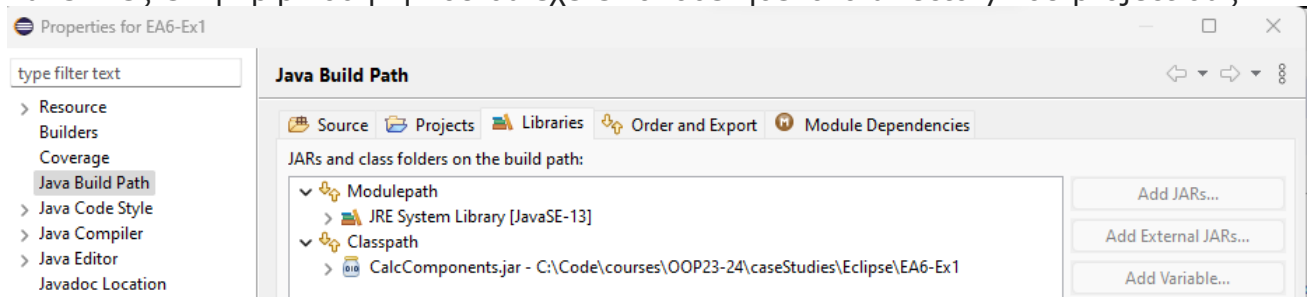
E1.5. - Ενημερώστε το IDE

A) Για Eclipse

Ενημερώστε το Eclipse (ή όποιο IDE χρησιμοποιείτε) για την θέση στην οποία θα βρει τις βιβλιοθήκες που χρησιμοποιεί το project σας. Επιλέξτε στον Package Explorer το project και δεξί κλικ και επιλέξτε

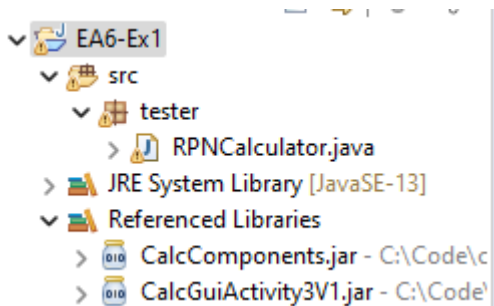
Project-Properties-Java Build Path-Libraries-ClassPath-Add external JARs

και επιλέξτε την βιβλιοθήκη που θα έχετε τοποθετήσει στο directory του project σας.



Επιλέξτε Apply and Close.

Στον project Explorer θα τις δείτε κάτω από το **Referenced Libraries** όπως φαίνεται παρακάτω.



B) Για BlueJ

Για ενημέρωση του BlueJ για την βιβλιοθήκη κλάσεων επιλέξτε

Tools-Preferences-Libraries-Add και επιλέξτε την βιβλιοθήκη
CalcGuiActivity3V1.jar

Αν χρησιμοποιείτε άλλο IDE αναζητήστε στο διαδίκτυο πως το κάνετε ενημέρωση για χρήση αρχείων jar.

4.3 Περιγραφή των συνιστωσών του προγράμματος

Μια προσεκτική μελέτη του προβλήματος με στόχο την ικανοποίηση των επιδιώξεων της ΕΑ οδήγησε στον ορισμό της δομής της RPN Calculator ως συνάθροισης από τα παρακάτω αντικείμενα.

4.3.1 CalculatorGui

Το βασικότερο, για τον χρήστη, αντικείμενο της εφαρμογής είναι αυτό που του επιτρέπει να χρησιμοποιήσει την υπηρεσία που περιγράφει το πρόγραμμα μας και θα προσφέρει η μηχανή που θα εκτελεί το πρόγραμμα μας.

Επιλέξαμε στη φάση αυτή η επικοινωνία ανθρώπου με την μηχανή να υλοποιείται με χρήση γραφικής διεπαφής (**Σχήμα 1**).

Προσέξτε όμως, η γραφική διεπαφή δεν εκτελεί καμία ουσιαστική λειτουργία εκτός από το να επιτρέπει στον χρήστη να αλληλοεπιδρά με τα αντικείμενα που υλοποιούν τη λειτουργικότητα της αριθμομηχανής μας, την RPN Calculator engine. 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 θα σας δώσει και τη δυνατότητα να κατανοήσετε τις αρμοδιότητες που έχουν ανατεθεί στα αντικείμενα που απαρτίζουν την μηχανή (engine) της αριθμομηχανής μας.

4.3.2 Operand

Βασική κλάση που αναπαριστά δομή και συμπεριφορά του τελεστέου ο οποίος 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`), τη 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.

4.3.3 Adder, Divider, Multiplier, και Subtractor

Η κλάση `Adder` αναπαριστά τον αθροιστή που δουλεύει πάνω σε στοίβα. Την κλάση αυτή έχουμε ήδη μελετήσει αναλυτικά σε προηγούμενη ΕΑ.

Αντίστοιχές είναι και οι άλλες κλάσεις που αναπαριστούν τους υπόλοιπους τελεστές που υποστηρίζει η έκδοση αυτή.

4.3.4 ResultPresenter

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

4.3.5 RPNCalculator

Είναι η κλάση που θα αναπαριστά το πρόγραμμα και για τον λόγο αυτό θα την ονομάσουμε `RPNCalculator`. Αυτή θα περιέχει την `main()` η οποία θα πρέπει να εμφανίσει τη γραφική διεπαφή.

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

Έχει δηλαδή έχει την ευθύνη να δημιουργήσει τα αντικείμενα που απαρτίζουν το σύστημα και να αποκαταστήσει τις μεταξύ τους εξαρτήσεις.

4.4 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 ευρετήριο 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 No 2 - Ανάπτυξη συνθετικών της RPN Calculator

Στην άσκηση αυτή θα αναπτύξετε μια RPN Calculator με δικά σας συνθετικά. Δεν θα χρησιμοποιήσετε δηλαδή τα έτοιμα συνθετικά που σας δόθηκαν στην Άσκηση1 αλλά θα αναπτύξετε τα δικά μας. Εξαιρέση αποτελεί η γραφική διεπαφή την οποία θα αναπτύξετε στην ενότητα 6.

5.1 Διαδικασία Ανάπτυξης

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

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.

Ανάλογα με το επίπεδο σας μπορείτε να επιλέξετε να συμπτύξετε 2 ή περισσότερα iterations σε ένα.

Στην επόμενη ενότητα (5.2) σας δίνονται οδηγίες για κάθε iteration. Ανατρέξτε σε αυτές μόνο μετά από την δική σας προσπάθεια. Αξιοποιήστε επιλεκτικά όσες λιγότερες από αυτές μπορείτε.

5.1.1 Iteration No 1 – Εμφάνιση γραφικής διεπαφής

Το iteration αυτό έχει στόχο την ανάπτυξη της πρώτης έκδοσης της RpnCalculator η οποία απλά θα εμφανίζει την γραφική διεπαφή.

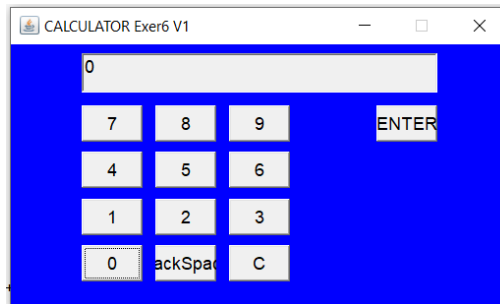
Ως γραφική διεπαφή θα χρησιμοποιήσετε αυτή που σας δίνεται στο αρχείο CalcGuiV1.jar <https://drive.google.com/file/d/1Lkm03uoYTJuew-osf66Mk5q-KN3Ee8hN/view?usp=sharing> (Πηγή)

Δράση 2.1

Δημιουργήστε ένα project με όνομα **LabEx6-It1**. Δουλέψτε στο default package.

Ο δημιουργός της `CalculatorGui` δέχεται ένα όρισμα τύπου Operand.

Δώστε τον πηγαίο κώδικα δουλεύοντας με μία dummy Operand. Στην οθόνη θα πρέπει να δείτε μια γραφική διεπαφή όπως αυτή στο Σχήμα 3.



Σχήμα 3 Η γραφική διεπαφή του 1ου iteration.

Πατήστε τα πλήκτρα των ψηφίων, το Backspace και το ENTER. Εξηγήστε την συμπεριφορά του προγράμματος.

5.1.2 Iteration No 2 – Ορισμός της κλάσης Operand

Στο iteration αυτό θα δημιουργήσουμε μια 2^η έκδοση της RpnCalculator. 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 Άσκηση 1 την είχαμε έτοιμη. Εδώ θα πρέπει να την ορίσετε.

Δράση 2.2

Δημιουργήστε ένα αντίγραφο του project του 1^{ου} iteration και δουλέψτε πάνω σε αυτό. Θα πρέπει να ορίσετε το σώμα της Operand. Εξετάστε δύο τουλάχιστον υλοποιήσεις της Operand.

Δράση 2.3

Τροποποιήστε το πρόγραμμα 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 Ch6_LegoApproachPartC.

5.1.3 Iteration No 3 – Ορισμός λοιπών κλάσεων

Στην έκδοση αυτή (3^η έκδοση) θα προσθέσουμε τη λειτουργικότητα των πράξεων και της εμφάνισης του αποτελέσματος. Θα ορίσετε δηλαδή τις δικές σας κλάσεις *Adder*, *Subtracter*, *Multiplier*, κλπ.

Δράση 2.4

Δημιουργήστε ένα αντίγραφο του project του 2^{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 *CalcGuiV2.jar*

https://drive.google.com/file/d/12T0_g1wHFG_w4S_Cib9ighVhkQmH7O9F/view?usp=sharing

Η κλάση έχει τον παρακάτω δημιουργό

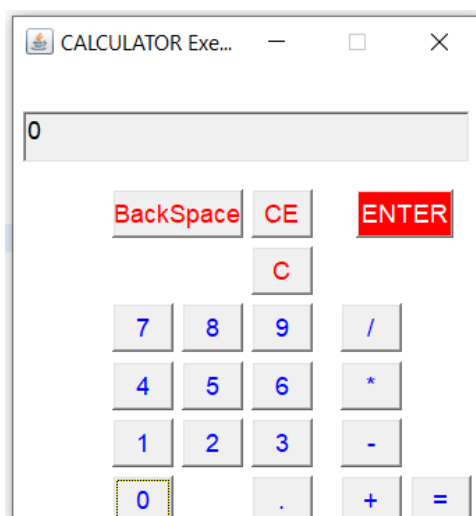
```
CalculatorGui(Operand op, Adder add, Subtracter sub, Multiplier mul, Divider div, ResultPresenter rp)
```

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

Δράση 2.5

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 *Ch6_LegoApproachPartC*.



Σχήμα 4. Η γραφική διεπαφή του 4ου iteration.

5.1.4 Iteration No 4 – Αξιοποίηση της Κληρονομικότητας (inheritance)

Στην έκδοση αυτή δεν θα προσθέσετε επιπλέον λειτουργικότητα. Θα τροποποιήσετε τον κώδικα ώστε να αξιοποιήσετε τον μηχανισμό της κληρονομικότητας.

Δράση 2.6

Αναγνωρίστε σχέση γενίκευσης/εξειδίκευσης στο πρόβλημα σας (ή στον κώδικα σας) και δώστε ένα διάγραμμα κλάσεων που καταγράφει την σχέση αυτή.

Δράση 2.7

Δημιουργήστε ένα αντίγραφο του project του 3^{ου} iteration και δουλέψτε πάνω σε αυτό. Αξιοποιήστε τον μηχανισμό της κληρονομικότητας (inheritance) για να υλοποιήσετε την σχέση γενίκευσης/εξειδίκευσης που καταγράψατε στο διάγραμμα κλάσεων.

Την όλη διαδικασία συζητήσαμε στις διαλέξεις. Περιγράφεται στις διαφάνειες με τίτλο Ch6_LegoApproachPartC.

5.1.5 Iteration No 5 – Αξιοποίηση του Interface

Στην έκδοση αυτή δεν θα προσθέσουμε επιπλέον λειτουργικότητα. Θα τροποποιήσουμε τον κώδικα της Operand και της CalculatorGui ώστε να αξιοποιούν την κατασκευή του interface.

Το iteration αυτό θα το δουλέψετε όταν έχετε ένα πρώτο δικό σας ορισμό της κλάσης CalculatorGui. Αυτό θα το κάνετε στην επόμενη Άσκηση ή μετά το iteration No 6.

Δράση 2.8

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

Την όλη διαδικασία συζητήσαμε στις διαλέξεις.

5.1.6 Ερώτηση

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

5.1.7 Iteration No 6 – Επέκταση λειτουργικότητας και βελτιώσεις πηγαίου κώδικα

Ο στόχος αυτού του iteration είναι να επεκτείνετε λειτουργικότητα και να βελτιώσετε πηγαίο κώδικα του οποίου τη σημασία δεν κατανοείτε στην φάση αυτή πλήρως. Ο πηγαίος κώδικας αφορά τη γραφική διεπαφή της αριθμομηχανής και περιλαμβάνει λάθη και κακές επιλογές που συνήθως υιοθετούν αρχάριοι προγραμματιστές. Προσπαθήστε να εκτελέσετε το iteration αυτό με τα όσα έχετε διδαχθεί μέχρι σήμερα. Τα περί γραφικών διεπαφών θα δούμε αναλυτικά σε επόμενες διαλέξεις.

Δράση 2.9

Δημιουργήστε ένα αντίγραφο του iteration 3 και δουλέψτε πάνω σε αυτό.

Δράση 2.10

Κάντε τις απαραίτητες τροποποιήσεις στο project σας ώστε να χρησιμοποιεί για γραφική διεπαφή αντί για την κλάση του αρχείου CalcGuiV2.jar την κλάση CalculatorGui η οποία σας δίνεται παρακάτω σε μορφή πηγαίου κώδικα.

Την κλάση CalculatorGui συνοδεύει και ένα σύνολο από άλλες βοηθητικές κλάσεις οι οποίες έχουν τοποθετηθεί στο αρχείο της κλάσης CalculatorGui.

Ως πρώτο βήμα διορθώστε ορισμένα λάθη που αναγνωρίζει το IDE.

Εκτελέστε το πρόγραμμα σας για να ελέγξετε την λειτουργικότητα του.

Δράση 2.11 Επέκταση λειτουργικότητας

Τροποποιήστε την γραφική διεπαφή ώστε να εκτελεί τις πράξεις του πολλαπλασιασμού και της διαίρεσης. Ελέγξτε την συμπεριφορά της αριθμομηχανής στα πλήκτρα C και CE.

Δράση 2.11 Κακές πρακτικές

Εντοπίστε κακές πρακτικές και τροποποιήστε τον κώδικα σας με στόχο την καλύτερη εφαρμογή του αντικειμενοστρεφούς παραδείγματος.

```
/**
 * CalculatorGui set of classes developed for the LabEx6-It6*
 * @author: Kleanthis Thramboulidis
 * date: 16/3/23
 */
import java.awt.event.*;

public class CalculatorGui extends Frame {
```

```

public static Operand op;
public static Adder add;
public static Subtractor sub;
public static ResultPresenter rp;
public Button button0, button1, button2, button3, button4;
public Button button5, button6, button7, button8, button9;
public Button buttonPlus, buttonMinus, buttonResultPresenter;
public Button buttonEnter, buttonBackSpace, buttonClear,buttonClearAll;
public static Frame window;
public static TextField display;

    public CalculatorGui(Operand op2, Adder add2, Subtractor sub2, Multiplier mult, Divider div2,
ResultPresenter rp2) {
        CalculatorGui.op = op2;
        CalculatorGui.add = add2;
        CalculatorGui.sub = sub2;
        CalculatorGui.rp = rp2;

        window = new Frame("LabEx5-It6");

        window.setLayout(null);
        window.setFont(new Font("TimesRoman", Font.PLAIN, 14));
        window.setBackground(Color.blue);

        button0 = new Button("0");
        button0.setBounds(64, 265, 35, 28);
        button0.setFont(new Font("TimesRoman", Font.PLAIN, 14));
        button0.setForeground(Color.blue);
        button0.addActionListener(new Button0Handler());
        CalculatorGui.window.add(button0);

        button1 = new Button("1");
        button1.setBounds(64, 232, 35, 28);
        button1.setFont(new Font("TimesRoman", Font.PLAIN, 14));
        button1.setForeground(Color.blue);
        button1.addActionListener(new Button1Handler());
        CalculatorGui.window.add(button1);

        button2 = new Button("2");
        button2.setBounds(104, 232, 35, 28);
        button2.setFont(new Font("TimesRoman", Font.PLAIN, 14));
        button2.setForeground(Color.blue);
        button2.addActionListener(new Button2Handler());
        CalculatorGui.window.add(button2);

        button3 = new Button("3");
        button3.setBounds(144, 232, 35, 28);
        button3.setFont(new Font("TimesRoman", Font.PLAIN, 14));
        button3.setForeground(Color.blue);
        button3.addActionListener(new Button3Handler());
        CalculatorGui.window.add(button3);

```

```
button4 = new Button("4");
button4.setBounds(64, 199, 35, 28);
button4.setFont(new Font("TimesRoman", Font.PLAIN, 14));
button4.setForeground(Color.blue);
button4.addActionListener(new Button4Handler());
CalculatorGui.window.add(button4);
```

```
button5 = new Button("5");
button5.setBounds(104, 199, 35, 28);
button1.setFont(new Font("TimesRoman", Font.PLAIN, 14));
button5.setForeground(Color.blue);
button5.addActionListener(new Button5Handler());
CalculatorGui.window.add(button5);
```

```
button6 = new Button("6");
button6.setBounds(144, 199, 35, 28);
button6.setFont(new Font("TimesRoman", Font.PLAIN, 14));
button6.setForeground(Color.blue);
button6.addActionListener(new Button6Handler());
CalculatorGui.window.add(button6);
```

```
button7 = new Button("7");
button7.setBounds(64, 166, 35, 28);
button7.setFont(new Font("TimesRoman", Font.PLAIN, 14));
button7.setForeground(Color.blue);
button7.addActionListener(new Button7Handler());
CalculatorGui.window.add(button7);
```

```
button8 = new Button("8");
button8.setBounds(104, 166, 35, 28);
button8.setFont(new Font("TimesRoman", Font.PLAIN, 14));
button8.setForeground(Color.blue);
button8.addActionListener(new Button8Handler());
CalculatorGui.window.add(button8);
```

```
button9 = new Button("9");
button9.setBounds(144, 166, 35, 28);
button9.setFont(new Font("TimesRoman", Font.PLAIN, 14));
button9.setForeground(Color.blue);
button9.addActionListener(new Button9Handler());
CalculatorGui.window.add(button9);
```

```
buttonEnter = new Button("ENTER");
buttonEnter.setFont(new Font("TimesRoman", Font.PLAIN, 14));
buttonEnter.setBackground(Color.red);
buttonEnter.setForeground(Color.white);
buttonEnter.setBounds(204, 100, 56, 28);
buttonEnter.addActionListener(new ButtonEnterHandler());
CalculatorGui.window.add(buttonEnter);
```



```

buttonBackSpace = new Button("BackSpace");
buttonBackSpace.setFont(new Font("TimesRoman", Font.PLAIN, 14));
buttonBackSpace.setForeground(Color.red);
buttonBackSpace.setBounds(64, 100, 75, 28);
buttonBackSpace.addActionListener(new ButtonBackSpaceHandler());
CalculatorGui.window.add(buttonBackSpace);

buttonClearAll = new Button("C");
buttonClearAll.setFont(new Font("TimesRoman", Font.PLAIN, 14));
buttonClearAll.setForeground(Color.red);
buttonClearAll.setBounds(144, 133, 35, 28);
buttonClearAll.addActionListener(new ButtonClearAllHandler());
CalculatorGui.window.add(buttonClearAll);

buttonClear = new Button("CE");
buttonClear.setFont(new Font("TimesRoman", Font.PLAIN, 14));
buttonClear.setForeground(Color.red);
buttonClear.setBounds(144, 100, 35, 28);
buttonClear.addActionListener(new ButtonClearHandler());
CalculatorGui.window.add(buttonClear);

buttonPlus = new Button("+");
buttonPlus.setBounds(195, 265, 35, 28);
buttonPlus.setFont(new Font("TimesRoman", Font.PLAIN, 14));
buttonPlus.setForeground(Color.blue);
buttonPlus.addActionListener(new ButtonPlusHandler());
CalculatorGui.window.add(buttonPlus);

buttonMinus = new Button("-");
buttonMinus.setBounds(195, 232, 35, 28);
buttonMinus.setFont(new Font("TimesRoman", Font.PLAIN, 14));
buttonMinus.setForeground(Color.blue);
buttonMinus.addActionListener(new ButtonMinusHandler());
CalculatorGui.window.add(buttonMinus);

buttonResultPresenter = new Button("=");
buttonResultPresenter.setBounds(235, 265, 35, 28);
buttonResultPresenter.setFont(new Font("TimesRoman", Font.PLAIN, 14));
buttonResultPresenter.setForeground(Color.blue);
buttonResultPresenter.addActionListener(new ButtonResultPresenterHandler());
CalculatorGui.window.add(buttonResultPresenter);

//D I S P L A Y   S E T T I N G S

display = new TextField("0");
display.setEditable(false);
display.setBounds(13, 55, 257, 30);

//W I N D O W   S E T T I N G S
window.add(display);

```

```

        window.setSize(283,320);
        window.setLocation(40,80);
        //window.show();
        window.setVisible(true);
        window.setResizable(false);
        window.addWindowListener(new CloseWindowAndExit());
    }
}

class CloseWindowAndExit extends WindowAdapter{
    public void windowClosing(WindowEvent closeWindowAndExit){
        System.exit(0);
    }
}

class Button0Handler implements ActionListener{
    public Button0Handler() { }

    public void actionPerformed(ActionEvent pushingButton0){
        CalculatorGui.op.addDigit('0'); }
}

class Button1Handler implements ActionListener{
    public Button1Handler() { }

    public void actionPerformed(ActionEvent pushingButton1)
    { CalculatorGui.op.addDigit('1'); }
}

class Button2Handler implements ActionListener{
    public Button2Handler() { }

    public void actionPerformed(ActionEvent pushingButton2)
    { CalculatorGui.op.addDigit('2'); }
}

class Button3Handler implements ActionListener{
    public Button3Handler() { }

    public void actionPerformed(ActionEvent pushingButton3)
    { CalculatorGui.op.addDigit('3'); }
}

class Button4Handler implements ActionListener{
    public Button4Handler() { }

    public void actionPerformed(ActionEvent pushingButton4)
    { CalculatorGui.op.addDigit('4'); }
}

class Button5Handler implements ActionListener{

```

```

public Button5Handler() { }

public void actionPerformed(ActionEvent pushingButton5)
{ CalculatorGui.op.addDigit('5'); }
}

class Button6Handler implements ActionListener{
public Button6Handler() { }

public void actionPerformed(ActionEvent pushingButton6)
{ CalculatorGui.op.addDigit('6'); }
}

class Button7Handler implements ActionListener{
public Button7Handler() { }

public void actionPerformed(ActionEvent pushingButton7)
{ CalculatorGui.op.addDigit('7'); }
}

class Button8Handler implements ActionListener{
public Button8Handler() { }

public void actionPerformed(ActionEvent pushingButton8)
{ CalculatorGui.op.addDigit('8'); }
}

class Button9Handler implements ActionListener{
public Button9Handler() { }

public void actionPerformed(ActionEvent pushingButton9)
{ CalculatorGui.op.addDigit('9'); }
}

class ButtonEnterHandler implements ActionListener{
public ButtonEnterHandler() { }

public void actionPerformed(ActionEvent pushingButtonEnter)
{ CalculatorGui.op.complete(); }
}

class ButtonBackSpaceHandler implements ActionListener{
public ButtonBackSpaceHandler() { }

public void actionPerformed(ActionEvent pushingButtonBackSpace)
{ CalculatorGui.op.deleteLastDigit(); }
}

class ButtonClearAllHandler implements ActionListener{
public ButtonClearAllHandler() { }
}

```

```

    public void actionPerformed(ActionEvent pushingButtonClearAll){
        CalculatorGui.op.reset();
    }
}

class ButtonClearHandler implements ActionListener{
    public ButtonClearHandler() { }

    public void actionPerformed(ActionEvent pushingButtonClear)
    { CalculatorGui.op.reset(); }
}

class ButtonPlusHandler implements ActionListener{
    public ButtonPlusHandler() { }

    public void actionPerformed(ActionEvent pushingButtonPlus)
    { CalculatorGui.add.operate(); }
}

class ButtonMinusHandler implements ActionListener{
    public ButtonMinusHandler() { }

    public void actionPerformed(ActionEvent pushingButtonMinus)
    { CalculatorGui.sub.operate(); }
}

class ButtonResultPresenterHandler implements ActionListener{
    public ButtonResultPresenterHandler() { }

    public void actionPerformed(ActionEvent pushingButtonResultPresenter)
    { CalculatorGui.rp.operate(); }
}

```

5.2 Οδηγίες Εκτέλεσης της άσκησης No 2

5.2.1 Iteration No 1 – Εμφάνιση γραφικής διεπαφής

Δημιουργήστε ένα project με όνομα **LabEx6-It1**.

Κάντε δεξί κλικ στον φάκελο src του project σας και επιλέξτε New->Class. Δώστε όνομα στην κλάση σας και επιλέξτε κάτω από το πλαίσιο Interfaces το κουτάκι

```
public static void main(String[] args)
```

Πατήστε Finish.

Κατεβάστε το αρχείο της γραφικής διεπαφής και ενημερώστε για αυτό το Eclipse.

Ενημερώστε την main ώστε να δημιουργεί ένα στιγμιότυπο της γραφικής διεπαφής. Ο δημιουργός της CalculatorGui είναι ο παρακάτω

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

Μπορείτε να τον χρησιμοποιήσετε με μια dummy Operand, δηλαδή μια Operand χωρίς σώμα. Θα ορίσετε το σώμα της Operand στο επόμενο iteration.

Εκτελέστε το πρόγραμμα σας. Παρατηρήστε και εξηγήστε την συμπεριφορά του.

5.2.2 Iteration No 2 – Ορισμός της κλάσης Operand

Για την διαμόρφωση του operand θα πρέπει να ορίσετε την μέθοδο **addDigit()**.

Για την τοποθέτηση του τελεστέου στην στοιβα (πλήκτρο ENTER) πρέπει να ορίσετε τη μέθοδο **complete()**.

Εξετάστε δύο τουλάχιστον υλοποιήσεις της Operand. Μία με μεταβλητή πρωτογενούς τύπου int και μία με αλφαριθμητικό.